

Sistem Informasi Akuntansi merupakan alat bagi perusahaan untuk memenangkan persaingan, hal ini terjadi karena sistem informasi akuntansi dapat menghasilkan informasi berkualitas bagi perusahaan tentang segala sesuatu yang terjadi diperusahaannya. Sehingga perusahaan dapat mengetahui apa yang harus dan tidak boleh dilakukan dalam menjalankan operasinya. Keadaan ini akan memberikan keuntungan bagi perusahaan dalam menghadapi persaingan.

Karena begitu tingginya manfaat sistem informasi akuntansi bagi perusahaan. Maka hampir semua perusahaan menengah dan besar didunia saat ini berlomba menggunakan berbagai macam teknologi mutakhir untuk membangun sistem informasi akuntansi yang berkualitas.

Tingginya kebutuhan perusahaan terhadap sistem informasi akuntansi berkualitas sudah barang tentu meningkatkan kebutuhan akan tenaga profesional yang terampil dalam membangun sistem informasi akuntansi.

Melihat keadaan tersebut maka buku ini sangat diperlukan sebagai pegangan baik bagi yang baru mulai memasuki dunia pengembangan sistem informasi atau bagi mereka yang selama ini telah bergelut dalam dunia pengembangan sistem informasi akuntansi sebagai tambahan referensi yang akan sangat bermanfaat.

Buku ini menjelaskan konsep dari sistem/siklus pengolahan transaksi serta kaitannya dengan sistem informasi akuntansi. Disamping itu buku ini juga menjelaskan berbagai siklus, metode dan teknik yang biasa digunakan dalam membangun sistem informasi akuntansi. Untuk mereka yang baru buku ini juga dilengkapi dengan contoh penyusunan program aplikasi sistem informasi akuntansi sederhana dengan menggunakan software DBMS visual foxpro.

Buku ini sangat diperlukan bukan hanya bagi mahasiswa S1, S2, dan S3 Jurusan akuntansi, sistem informasi tapi juga bagi para profesional dan pemula dalam membangun sistem informasi akuntansi. Kesempurnaan dalam buku ini masih jauh, karena ini segala bentuk saran yang sifatnya membangun untuk perbaikan buku ini sangat diharapkan.



Azhar susanto adalah Guru Besar dibidang sistem informasi akuntansi pada Departemen Akuntansi Fakultas Ekonomi dan Bisnis UNPAD. Menyelesaikan Studi S1 di Jurusan Akuntansi UNPAD, Post Graduate Diploma dan Master dalam Business Information System major Sistem Development pada Curtin University of Technology Perth, Western Australia.

Selain sebagai dosen S1, S2, S3 di UNPAD dan PTS, Partner pada KAP. Dr. La Midjan, Konsultan Sistem informasi, penulis juga aktif sebagai peneliti dan penulis di jurnal internasional terindeks Scopus dimana buku ini sering sebagai salah satu referensi.

Analisis dan Perancangan Sistem Informasi Akuntansi: Siklus - Metode - Teknik Prof. Dr. Azhar Susanto, MBus, CPA, AK, CA



Analisis dan Perancangan Sistem Informasi Akuntansi

Siklus - Metode - Teknik

Prof. Dr. Azhar Susanto, MBus, CPA, AK, CA
Universitas Padjadjaran



Analisis dan Perancangan Sistem Informasi Akuntansi **Siklus - Metode - Teknik**

Prof.Dr.Azhar Susanto,MBus,CPA,Ak,CA
Universitas Padjadjaran



Susanto.Azhar

Analisis dan Perancangan Sistem Informasi Akuntansi: Siklus-Metode-Teknik

Bandung: Lingga jaya, 2017, V +173 hlm; 18,3X25,2 Cm

ISBN : 978-602-60978-5-6

Edisi Perdana, Cetakan Pertama, 2017

Analisis dan Perancangan Sistem Informasi Akuntansi:

Siklus-Metode dan Teknik

Oleh: Prof.Dr. Azhar Susanto,MBus,CPA,Ak,CA

Hak Cipta dilindungi undang-undang

Dilarang memperbanyak seluruhnya atau sebagian dari buku ini dalam bentuk apapun tanpa seijin tertulis dari penulis dan penerbit

Pasal 44

- *Barang siapa dengan sengaja dan tanpa hak mengumumkan atau memperbanyak suatu ciptaan atau memberi izin untuk itu, dipidana dengan pidana penjara paling lama 7 (tujuh) tahun dan/ atau denda paling banyak Rp. 100.000.000,- (seratus juta rupiah).*
- *Barang siapa dengan sengaja menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu ciptaan atau barang hasil pelanggaran hak cipta sebagaimana dimaksud dalam ayat (1), dipidana dengan pidana penjara paling lama 5 (lima) tahun dan /atau denda paling banyak Rp. 50.000.000,- (lima puluh juta rupiah)*

Penerbit:



Lingga Jaya

Jl. Cigadung Raya Tengah,Komp.Cigadung Greenland K-2

Telp.(022) 93261163,08132172181,Bandung 40191

Pengantar Penulis

Menganalisis dan merancang sistem informasi akuntansi adalah topik penting yang sangat diperlukan oleh mahasiswa baik S1,S2 maupun S3 yang mengambil penelitian di bidang Akuntansi atau Sistem Informasi akuntansi.

Masyarakat memiliki pemahaman dan pandangan yang beragam tentang apa itu Sistem Informasi Akuntansi dan bagaimana membangunnya. Pandangan yang berbeda muncul antara praktisi analisis sistem, akademisi dan pengguna sistem informasi akuntansi.

Dosen yang memahami sistem informasi akuntansi sebatas teori-teori yang ada di buku akan memiliki pemahaman yang berbeda dengan para pengembang atau dosen lain yang memiliki pemahaman teoritis dan praktis serta akan sangat berbeda sekali pemahamannya dengan sebagian orang yang mengaku sebagai pengembang sistem yang memiliki keahlian dalam pemrograman yang dipelajarinya secara autodidak.

Terlepas dari banyaknya perbedaan pandangan diantara akademisi dan praktisi baik yang belajar secara formal di perguruan tinggi atau belajar sendiri yang pasti untuk dapat membangun sistem informasi akuntansi selain harus memahami dulu apa itu sistem informasi akuntansi juga harus dipahami bagaimana membangunnya dengan menggunakan siklus metode dan teknik yang tersedia.

Buku ini menjelaskan secara ringkas apa itu sistem informasi akuntansi dan siklus/sistem pengolahan transaksi yang dikenal sebagai bisnis proses atau fungsi bisnis serta siklus metode dan teknik yang digunakan dalam membangun sistem informasi akuntansi.

Buku ini dibuat selain untuk mahasiswa S1,S2 dan S3 juga sangat cocok digunakan oleh para praktisi agar pengembangan sistem informasi akuntansi yang mereka lakukan bersumber dari literatur yang umum digunakan diseluruh dunia.

Hormat kami,
Prof.Dr.Azhar Susanto

Daftar Isi

Pengantar Penulis	iii
Daftar Isi	iv
BAB 1 Sistem Informasi Akuntansi dan Sistem Pengolahan Transaksi	1
1.1 Peristiwa ekonomi.....	2
1.2 Akuntansi ..	2
1.3 Sistem pengolahan transaksi	3
1.4 Aktivitas dalam sistem pengolahan transaksi.....	4
1.5 Siklus pengolahan transaksi	7
1.6 Sistem informasi akuntansi	9
BAB 2 Analisis dan Perancangan Sistem Informasi Akuntansi.....	23
2.1 Analisis dan analisis sistem	24
2.2 Tugas dan tanggung jawab analisis sistem informasi akuntansi	25
2.3 Perbedaan tugas dan tanggungjawab analisis sistem dan programmer	27
2.4 Analisis sistem sebagai pemecah masalah	29
2.5 Kedudukan analisis sistem dalam organisasi.....	37
2.6 Persiapan karier sebagai analisis sistem	39
BAB 3 Siklus, Metode dan Teknik dalam Membangun Sistem Informasi Akuntansi	45
3.1 Siklus, metode dan teknik	46
3.2 Siklus	46
3.3 Metodologi	47
3.4 Teknik	64
3.5 Keterlibatan user dalam membangun sistem informasi akuntansi	75
BAB 4 Praktika Pembuatan Model Aplikasi sistem informasi akuntansi sederhana	79
4.1 Mengenal Microsoft Visual FoxPro	80
4.2 Membuat Program dengan Microsoft Visual FoxPro	83

1

Sistem Informasi Akuntansi dan Sistem Pengolahan Transaksi

Pendahuluan

Sistem informasi akuntansi pada dasarnya merupakan integrasi dari berbagai sistem pengolahan transaksi yang ada di perusahaan yang dulu dikenal sebagai Pengolahan Data Secara Elektronik (PDE) atau *Electronic Data Processing (EDP)*. Sistem-sistem pengolahan data transaksi di perusahaan yang bekerja di berbagai fungsi organisasi utamanya pada tingkat operasional,

Sistem pengolahan transaksi (SPT) ada di berbagai fungsi bisnis organisasi dengan tujuan untuk membantu fungsi bisnis tersebut dalam mengolah data operasinya menjadi informasi yang diperlukan oleh pihak. SPT dikenal juga sebagai subsistem dari sistem informasi akuntansi seperti sistem informasi akuntansi keuangan dan sistem informasi akuntansi manajemen. Sistem informasi akuntansi keuangan dibuat utamanya untuk kepentingan pihak luar organisasi yang ingin mengetahui kondisi keuangan perusahaan. Karena kepentingan perusahaan bervariasi maka sistem informasi akuntansi ini harus dibuat berdasarkan standar akuntansi keuangan yang berlaku di Indonesia seperti PSAK ETAP dan PSAK IFRS. Sistem informasi akuntansi keuangan juga dikenal sebagai sistem pelaporan keuangan yang dikenal juga sebagai program aplikasi GL atau SPT keuangan.

Bagian sistem informasi akuntansi lainnya adalah sistem informasi akuntansi manajemen. Sistem Informasi akuntansi manajemen berisi beberapa sistem pengolahan transaksi atau subsistem dari sistem informasi akuntansi manajemen sistem informasi akuntansi manajemen produksi, sumber daya manusia, keuangan dan lain lain tergantung kepada fungsi bisnis apa yang ada dalam organisasi perusahaan. Semua subsistem informasi akuntansi (sistem pengolahan transaksi) yang ada di dalam sistem informasi akuntansi manajemen harus terintegrasi baik secara hardware, software, brainware, prosedur database dan jaringan komunikasinya. Integrasi dari berbagai sub sistem informasi akuntansi manajemen akan membantu sistem informasi akuntansi manajemen tersebut dalam proses agregasi untuk mendapatkan total total dari setiap aktivitas berdasarkan fungsi bisnis. Semua agregasi dan dokumen yang dihasilkan oleh sistem informasi akuntansi manajemen menjadi dasar dalam sistem informasi akuntansi keuangan.

Bagian buku ini akan membahas tentang sistem informasi akuntansi manajemen, bagian selanjutnya buku ini akan membahas tentang bagaimana menganalisis dan merancang sistem informasi akuntansi dengan menggunakan siklus, metode dan teknik yang berlaku.

1.1 Peristiwa Ekonomi

Saat ini kita hidup dalam suatu masyarakat yang semakin kompleks, lebih kompleks dari pada sepuluh tahun yang lalu dan akan semakin kompleks dalam sepuluh tahun kedepan. Seperti kita ketahui di masyarakat terdapat berbagai jenis organisasi yang saling terkait dalam aktivitas yang berbeda dengan tujuan yang berbeda pula. Ada yang ingin mendapatkan laba seperti perusahaan dagang, industri, jasa, lembaga keuangan dll dan ada pula organisasi yang tidak berorientasi untuk mendapatkan laba seperti jawatan sosial, organisasi politik, departemen perijinan dan lain-lain. Semua jenis organisasi tersebut harus mencatat dan melaporkan semua aktivitasnya yang diperlukan untuk pengambilan keputusan..

Sistem informasi akuntansi dirancang tidak hanya untuk mendokumentasikan (mencatat, mengklasifikasikan, meringkas dan melaporkan) hasil aktivitas atau peristiwa ekonomi dengan berbagai organisasi yang terkait untuk kepentingan internal dan eksternal organisasi akan tetapi juga berfungsi sebagai pedoman serta pengendali terhadap bagaimana pendokumentasian tersebut harus dilakukan oleh suatu organisasi, baik itu organisasi berorientasi laba atau bukan.

Bagaimana anda dapat mulai memahami struktur dari berbagai aktivitas ekonomi dan sistem informasi akuntansinya? Bagaimana perancang sistem informasi akuntansi yang mampu mengantisipasi berbagai transaksi/kejadian ekonomi yang mungkin terjadi?.

Memahami aktivitas dasar atau peristiwa ekonomi dan aktivitas lainnya yang mungkin timbul sebagai hasil interaksi perusahaan dengan pihak lain yang ada di lingkungan perusahaan tersebut serta memahami konsep sistem informasi akuntansi secara terstruktur sangat membantu dalam membangun sistem informasi akuntansi yang akan dibahas pada bagian akhir buku ini.

Memahami berbagai peristiwa ekonomi yang terjadi sebagai akibat interaksi perusahaan dengan lingkungannya dapat dilakukan dengan mempelajari berbagai aktivitas yang dilakukan oleh organisasi perusahaan tersebut sehari-hari. Sedangkan untuk memahami konsep sistem informasi akuntansi secara terstruktur terlebih dahulu harus dipahami konsep tentang sistem, informasi, sistem informasi dan akuntansi. Konsep sistem, informasi dan sistem informasi telah dibahas pada bab-bab sebelumnya, Berikut ini akan dibahas mengenai akuntansi sebagai konsep untuk memahami sistem informasi akuntansi.

1.2 Akuntansi

American Accounting Association (AAA)(1966), *Wilkinson* (2000), *Warren dan Fess*(1996), mendefinisikan akuntansi sebagai sistem informasi yang menghasilkan informasi atau laporan untuk berbagai kepentingan baik individu atau kelompok tentang aktivitas/operasi/peristiwa ekonomi atau keuangan suatu organisasi.

Semua kejadian atau peristiwa ekonomi yang diakibatkan oleh aktivitas/operasi suatu organisasi perusahaan yang berpengaruh terhadap posisi keuangan organisasi perusahaan tersebut disebut sebagai transaksi (transaksi bisnis) bagi organisasi perusahaan tersebut. Atau lebih dikenal sebagai transaksi akuntansi.

Semua kejadian ekonomi yang tidak berpengaruh terhadap posisi keuangan organisasi perusahaan bukan merupakan transaksi akuntansi bagi organisasi perusahaan tersebut.

Jadi transaksi bisnis atau akuntansi pada dasarnya merupakan penyeleksian semua aktivitas ekonomi kedalam aktivitas yang hanya berkaitan dengan suatu organisasi perusahaan, aktivitas ekonomi lainnya yang tidak berkaitan dikesampingkan. Transaksi bisnis tersebut kemudian dimasukkan kedalam suatu formulir (kertas atau layar komputer) sehingga menjadi dokumen atau *data* untuk selanjutnya diolah menjadi informasi atau informasi akuntansi.

1.3 Sistem Pengolahan Transaksi

Sistem pengolahan transaksi memproses *data* yang berasal dari kejadian atau peristiwa ekonomi sebagai akibat dari operasi atau aktivitas internal perusahaan dan sebagai akibat dari aktivitas perusahaan dengan pihak luar menjadi informasi keuangan/informasi akuntansi keuangan yang berguna bagi manajemen ditingkat operasional dan pihak lain.

Dengan menggunakan komputer, aktivitas pengolahan transaksi meliputi mengumpulkan, mengirimkan, memasukan, mengolah dan menyimpan *data-data* tentang kejadian atau peristiwa ekonomi yang berasal atau disebabkan oleh aktivitas/operasi organisasi sehari-hari.

Beberapa *data* setelah diolah menjadi informasi diperlukan oleh pemerintah dan lembaga keuangan seperti informasi akuntansi keuangan (laporan keuangan). Informasi yang lainnya mungkin diperlukan untuk mengendalikan operasi organisasi perusahaan, pengambilan keputusan taktis, dan perencanaan strategis.

Pengolah transaksi (program) dalam mengolah data transaksi dapat melakukannya secara *Batch*, secara *On-line* atau gabungan kedua cara tersebut (*Hybrid*). Dalam sistem pengolahan *data* transaksi secara *online*, pengolahan *data* transaksi dilakukan saat transaksi terjadi. Sebagai contoh *data* transaksi penjualan diolah secara *on-line* agar diketahui total penjualan pada saat itu. Pengolahan *data* seketika (*Real time*) adalah jenis lain dari pengolahan secara *on-line*, dalam pengolahan *data* ini bukan hanya *data* transaksi yang masuk diolah ketika transaksi terjadi, akan tetapi juga *data* lainnya seperti persediaan dan piutang dilakukan *updating* sehingga bisa diketahui status persediaan dan piutang setiap saat.

Dalam pengolahan *data* secara *Batch data* terlebih dahulu dikumpulkan sampai pada saat tertentu data tersebut kemudian diolah untuk menghasilkan informasi.

Sistem pengolahan transaksi memproses *data* yang berasal dari kejadian atau peristiwa ekonomi sebagai akibat dari operasi atau aktivitas internal perusahaan dan sebagai akibat dari aktivitas perusahaan dengan pihak luar menjadi informasi yang berguna bagi manajemen ditingkat operasional.

Mengolah data transaksi bisa secara *batch* atau secara *on-line*

Dalam praktek, perusahaan biasanya menggunakan dua macam sistem pengolahan *data* tersebut (*hybrid*). Perusahaan tidak dapat hanya menggunakan satu sistem pengolahan saja dalam mengolah *datanya*.

1.4 Aktivitas dalam Sistem Pengolahan Transaksi

Seperti telah dijelaskan sebelumnya bahwa pengolahan transaksi dapat berupa rangkaian proses yang dimulai dari pengumpulan, pengiriman dan pemasukan *data* untuk disimpan atau diproses sehingga menghasilkan *output* berupa informasi yang berguna bagi pemakai.

- **Pengumpulan, pengiriman, dan pemasukan *data***- Semua *data* bernilai ekonomi yang berasal dari aktivitas operasi organisasi dan berpengaruh terhadap posisi keuangan organisasi perusahaan (transaksi) harus dikumpulkan, termasuk didalamnya *data* yang berasal dari aktivitas yang berkaitan dengan bahan baku, proses produksi, penjualan, distribusi barang dan jasa.

Aktivitas atau transaksi tersebut meliputi sejumlah aktivitas organisasi perusahaan yang berkaitan dengan pihak eksternal ataupun aktivitas yang terjadi antar bagian yang ada di dalam organisasi. Aktivitas atau transaksi yang terjadi diantaranya meliputi penjualan, pengurangan persediaan, pengiriman barang, penerimaan kas, pembelian, penerimaan persediaan, pembayaran, konversi dari tenaga kerja dan bahan baku ke barang jadi atau jasa.

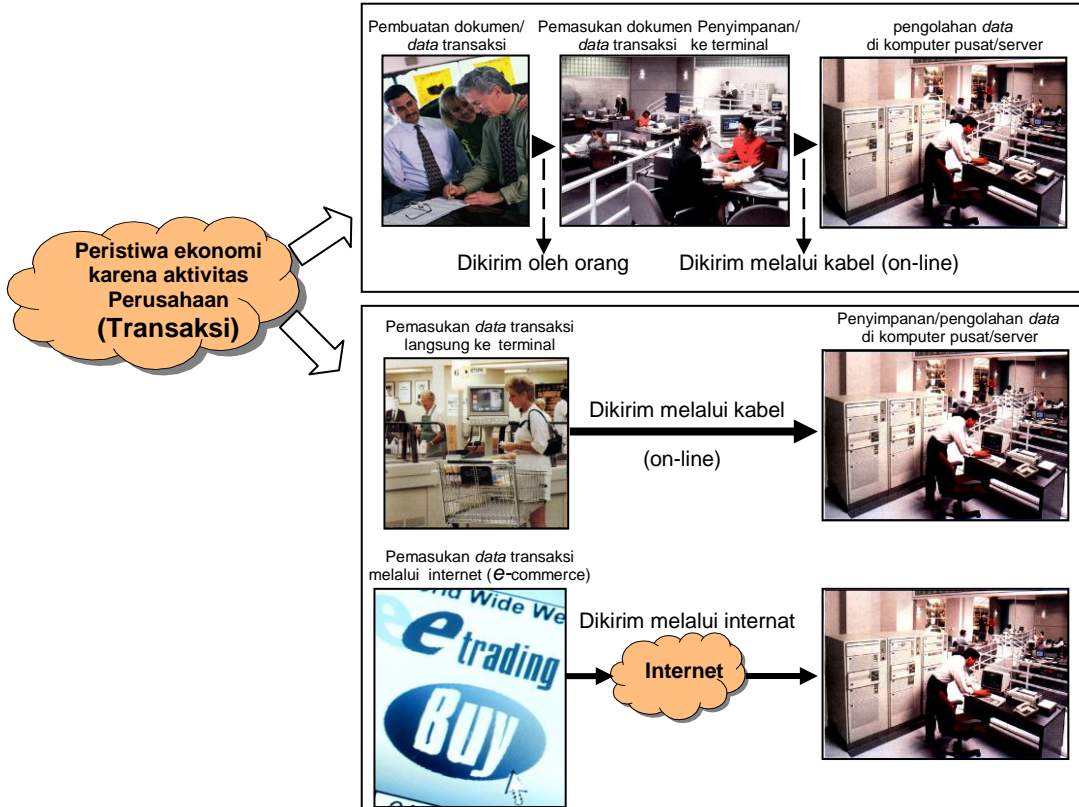
Data transaksi yang diperoleh karena transaksi yang terjadi dikumpulkan atau ditampung ke dalam suatu formulir (formulir bila telah di isi berubah menjadi dokumen), *data* tersebut selanjutnya dikirim ke bagian pemasukan *data* untuk dimasukkan melalui terminal komputer ke pusat komputer (*server*) yang selanjutnya disimpan atau diolah lebih lanjut.

Data transaksi dapat juga langsung dikumpulkan melalui terminal komputer biasa atau terminal *internet* (*e-commerce*) untuk selanjutnya dari terminal tersebut dikirim ke komputer pusat untuk disimpan atau diolah lebih lanjut.

Bagi suatu organisasi perusahaan cara mana yang akan digunakan sangat tergantung kepada kepentingan perusahaan tersebut, situasi kondisi dan sejauh mana organisasi perusahaan mampu menggunakan teknologi informasi untuk SPT-nya. Pemilihan salah satu cara tidak menjamin baiknya tidaknya SPT bekerja. Baik tidaknya kinerja SPT sangat tergantung kepada sejauh mana SPT dapat memenuhi kebutuhan penggunaannya yaitu organisasi perusahaan dan pihak luar yang berkepentingan.

Keharmonisan antara informasi yang dihasilkan oleh SPT terhadap kepentingan pihak intern atau ekstern organisasi akan menentukan kinerja dari SPT tersebut apakah itu menggunakan teknologi informasi yang canggih atau tidak, bahkan mungkin secara manual pun keharmonisan itu dapat diperoleh.

Gambar 1.1 Berbagai cara mengolah transaksi



Semua *data* yang dimasukkan ke komputer pusat mungkin diolah dikemudian hari (*batch*) atau langsung diolah di komputer pusat tersebut ketika *data* transaksi dimasukkan (*on-line*).

Data transaksi dapat dimasukkan ke dalam komputer dengan menggunakan berbagai macam media *input* seperti *keyboard*, *mouse*, *optical character recognition (OCR)*, *magnetic ink reader* dan lain-lain. Bahkan saat ini dapat melalui *personal digital assistant (PDA)* atau komputer tablet

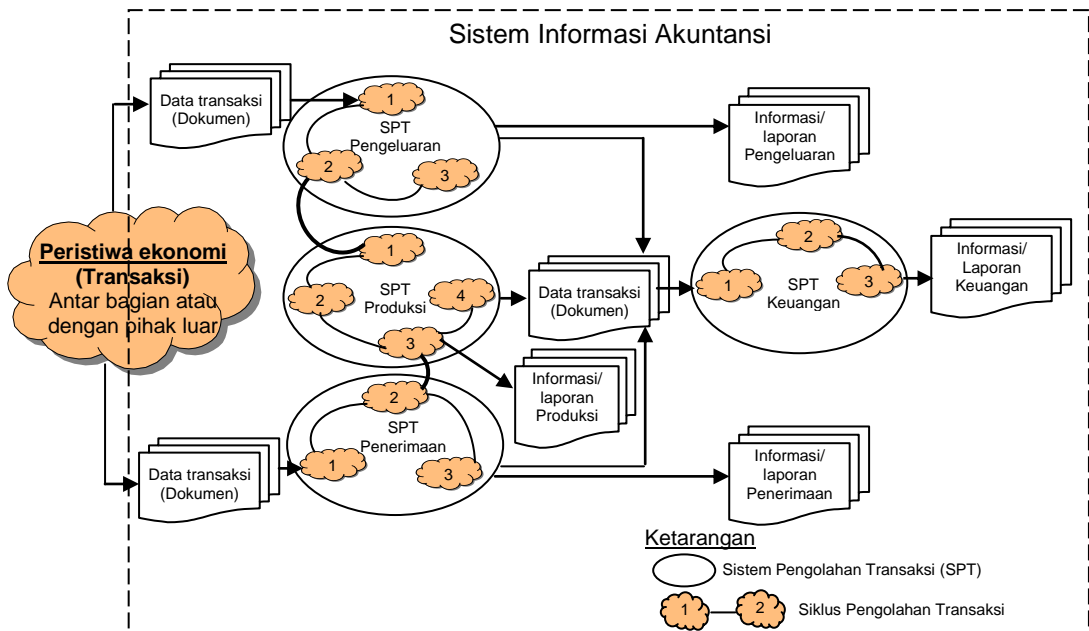
Semua *data* yang masuk memerlukan editing, verifikasi dan validasi untuk menjamin bahwa *data* yang dimasukkan lengkap, akurat dan sah.

- **Pengolahan dan manipulasi *data***- Pengolahan *data* menjadi format yang berarti dan berguna melibatkan beberapa aktivitas termasuk: pengkodean dan pengklasifikasian, penyusunan, perhitungan, dan peringkasan. Sistem klasifikasi harus dibuat untuk menyimpan *data* agar dapat digunakan secara efektif bagi manajemen. *Data* harus diberi kode yang benar agar secara logis dapat disusun dan mudah untuk dibaca.
- **Menyimpan *data*** - Ada beberapa cara penyimpanan *data* yang dapat digunakan seperti penyimpanan *data* secara berurutan

(*sequensial*), acak (*random*), dengan rumus (*hasing*), dan urutan yang diindek (*indexed sequensial*). Perlu dipelajari berbagai cara penyimpanan *data* tersebut lebih detail. Untuk itu berbagai elemen *data* yang harus disimpan perlu diketahui, beberapa diantara elemen *data* tersebut kemungkinan sangat besar untuk disimpan. Yang penting secara umum transaksi yang terjadi hari ini harus memiliki identitas seperti nomor, nama orang yang melakukan transaksi, apa yang ditransaksikan, tanggal transaksi, bagian yang melakukan transaksi, dan otorisasi. Secara singkat siapa, apa, kapan, dimana, dan otorisasi setiap transaksi harus disimpan.

- **Melaporkan-** Produk dari SPT dapat berupa informasi keuangan/akuntansi keuangan bagi manajemen tingkat bawah, dokumen intern organisasi atau bisa juga dalam bentuk laporan keuangan untuk pihak eksternal. Semua dokumen atau laporan tersebut dapat dicetak kapan saja kedalam kertas atau layar monitor,

Gambar 1.2 Aktivitas sistem pengolahan transaksi dan sistem Informasi akuntansi



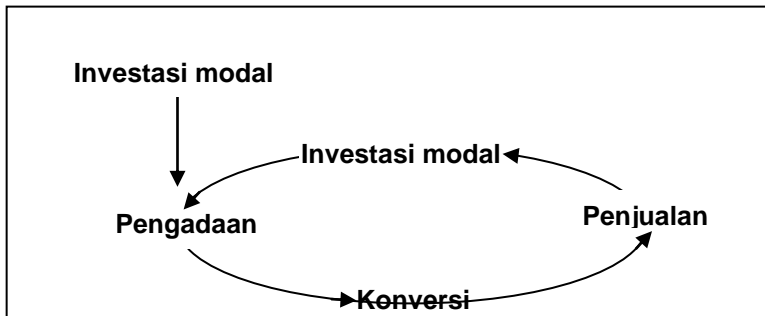
1.5 Siklus Pengolahan Transaksi

Siklus pengolahan transaksi merupakan prosedur atau urutan-urutan subsistem pengolahan transaksi untuk menunjang siklus transaksi dalam aktivitas bisnis perusahaan atau siklus aktivitas bisnis perusahaan, sehingga anda dapat mengatakan bahwa siklus aktivitas bisnis perusahaan direpresentasikan oleh siklus/sistem pengolahan transaksi.

Di dalam siklus aktivitas atau transaksi bisnis perusahaan terdapat aktivitas atau transaksi investasi modal, pengadaan, konversi dan penjualan.

Siklus pengolahan transaksi merupakan prosedur atau urutan-urutan subsistem pengolahan transaksi untuk menunjang siklus transaksi tertentu dalam suatu aktivitas bisnis perusahaan

Gambar 1.3 Siklus aktivitas bisnis perusahaan



- **Investasi modal**, siklus aktivitas bisnis dimulai saat modal diinvestasikan ke dalam perusahaan. Standar akuntansi keuangan (SAK) mengakui perusahaan sebagai suatu entitas yang terpisah dengan pemberi modal. Pemberi modal mungkin pemilik perusahaan atau kreditur. Bila pemberi modal merupakan pemilik perusahaan maka investasi yang ditanam tersebut merupakan kekayaan yang menjadi hak pemilik. Bila modal berasal dari kreditur maka investasi yang ditanamkan merupakan hutang (kewajiban) jangka pendek atau jangka panjang.

Pada kebanyakan perusahaan, sebagian besar modal digunakan untuk membeli aktiva produktif yang memberikan masa manfaat jangka panjang atau aktiva tetap. Perusahaan membeli aktiva produktif untuk meningkatkan modalnya. Secara periodik perusahaan melaporkan hasil dari operasinya ke sumber atau pemberi modal

Aktivitas investasi untuk memberi modal perusahaan melibatkan dua aktivitas ekonomi (transaksi akuntansi) utama yaitu menambah jumlah modal dan menggunakan modal tersebut untuk membeli aktiva produktif.

- **Pengadaan**, komponen ke dua dalam siklus aktivitas bisnis adalah pengadaan bahan baku dan perlengkapan pendukung. Pengadaan ini dimaksudkan untuk meningkatkan modal usaha. Cara pasti bagaimana mereka menggunakan bukan merupakan hal penting disini. Mereka digunakan dengan cara yang berbeda pada perusahaan yang berbeda.

- **Konversi**, Tahap selanjutnya dalam aktivitas bisnis adalah konversi (produksi) dari bahan baku menjadi barang atau jasa. Perusahaan menjual barang dan jasa ini untuk meningkatkan modal. Perusahaan industri membeli persediaan bahan baku, kemudian menggunakan bahan baku tersebut untuk membuat sesuatu yang berbeda dari aslinya dengan menggunakan tenaga kerja dan biaya tidak langsung. Perusahaan jasa merubah permintaan pelayanan menjadi pelayanan dengan bantuan sumber daya manusia. Sedangkan perusahaan dagang (eceran atau grosir) membeli persediaan barang, mengembalinya kembali dan memasarkannya. Semua jenis perusahaan yang disebutkan diatas menggunakan persediaan dan perlengkapan dalam proses konversi mereka.
- **Penjualan**, Komponen akhir dalam siklus bisnis adalah penjualan barang dan jasa yang merupakan output dari proses konversi. Ketika penjualan ini menghasilkan laba maka modal yang diinvestasikan kedalam perusahaan meningkat. Adanya kelebihan kas dapat digunakan untuk investasi kembali atau untuk membayar sumber modal dalam bentuk dividen atau bunga. Dengan memberi pemberi modal tambahan modal, komponen penjualan melengkapi (mengakhiri) siklus aktivitas bisnis perusahaan.

Siklus aktivitas bisnis diatas merupakan siklus transaksi. Siklus transaksi tersebut menghasilkan *data* yang akan diolah oleh siklus pengolahan transaksi. Jadi siklus aktivitas bisnis atau siklus transaksi ditunjang oleh siklus-siklus pengolahan transaksi seperti siklus pengolahan transaksi penerimaan untuk menunjang aktivitas atau transaksi penjualan, siklus pengolahan transaksi pengeluaran untuk menunjang aktivitas atau transaksi pengadaan, dan lain-lain. Sebagai contoh siklus-siklus transaksi dan pengolahan transaksi tersebut diatas akan terdiri dari beberapa transaksi seperti ditunjukkan pada tabel 4.1 dibawah ini. Siklus-siklus transaksi yang didukung oleh beberapa siklus/sistem pengolahan transaksi ini antara satu perusahaan dengan perusahaan lainnya baik itu sejenis atau tidak secara detail belum tentu sama tergantung kepada kebijaksanaan manajemen pada perusahaan tersebut.

Seperti dijelaskan diatas, siklus transaksi menunjukkan bagaimana suatu peristiwa yang terjadi pada awal siklus transaksi mempengaruhi peristiwa dan catatan tentang peristiwa tersebut selanjutnya dalam suatu siklus. Lemahnya pengendalian dalam suatu transaksi dan pencatatan atau pengolahan data transaksi atas data yang dihasilkan oleh transaksi tersebut akan menghasilkan kekeliruan atau kesalahan pengolahan terhadap data transaksi yang terjadi berikutnya.

Pada organisasi perusahaan yang telah modern, dimana data transaksinya telah terintegrasi serta pengolahan data transaksi tersebut telah menggunakan komputer akan sangat sulit membedakan proses data transaksi apa yang sedang dilakukan terhadap transaksi yang terjadi. Untuk menganalisis proses-proses atau transaksi yang terjadi dan bagaimana pengaruhnya terhadap sistem akan mudah dilakukan bila berdasarkan kepada siklus transaksi.

Tabel 1.1 Hubungan siklus bisnis, siklus transaksi, transaksi akuntansi serta dokumen pendukungnya.

Siklus Bisnis Siklus transaksi	Aktivitas ekonomi/ Transaksi akuntansi	Dokumen
Investasi modal Siklus Keuangan Ditunjang oleh siklus/ sistem pengolahan transaksi keuangan	<ol style="list-style-type: none"> 1) Pengadaan atau peningkatan modal dari pemilik atau kreditur 2) Menggunakan modal untuk aset produktif (properti, tanah dan peralatan) 3) Membuat laporan secara periodik ke pemberi modal (pemilik atau kreditur) 	<ol style="list-style-type: none"> 1.1) Bukti penyerahan modal investasi 2.1) Bukti pembelian aset produktif 3.1) Neraca dan rugi laba 3.2) Laporan perubahan modal 3.3) Arus sumber dan penggunaan dana 3.4) Buku besar 3.5) Jurnal 3.6) Daftar akun
Pengadaan Siklus Pengeluaran Ditunjang oleh siklus/sistem pengolahan transaksi pengeluaran	<ol style="list-style-type: none"> 1) Permintaan bahan untuk proses konversi 2) Pemesanan bahan 3) Penerimaan bahan 4) Pembayaran kewajiban 	<ol style="list-style-type: none"> 1.1) Surat permintaan pembelian barang 2.1) Order pembelian 3.1) Tanda terima barang 4.1) Faktur dengan tanda lunas
Konversi Siklus Konversi Ditunjang oleh siklus/sistem pengolahan transaksi konversi	<ol style="list-style-type: none"> 1) Persiapan membuat pesanan 2) Permintaan material 3) Penggunaan bahan baku dan overhead 4) Penggunaan tenaga kerja 5) Penyerahan barang 	<ol style="list-style-type: none"> 1.1) Perencanaan dan pengendalian produksi (Produk planning and control/PPC) 2.1) Surat Permintaan pembelian 3.1) Kartu Biaya produksi/ Job order Cost Sheet 3.2) Bukti penerimaan barang 4.1) Slip Pembayaran gaji 5.1) Bukti pengeluaran barang
Penjualan Siklus Penerimaan Ditunjang oleh siklus/sistem pengolahan transaksi penerimaan	<ol style="list-style-type: none"> 1) Menerima permintaan atau pesanan barang 2) Pengecekan barang ke gudang 3) Mengirim barang ke konsumen 4) Meminta pembayaran atau penagihan 5) Menerima pembayaran atau kas 	<ol style="list-style-type: none"> 1.1) Order penjualan 2.1) Status persediaan 3.1) Surat perintah pengiriman barang 3.2) Slip pengiriman barang dengan tanda terima 4.1) Faktur jual 5.1) Faktur jual dengan tanda terima

Transaksi untuk setiap siklus pengolahan memiliki beberapa pemasukan (*input*), pengolahan (*processing*), penyimpanan (*storage*) dan informasi (*output*) yang mungkin sama akan tetapi, tingkat kepentingan untuk setiap siklus bervariasi tergantung kepada jenis organisasi. Beberapa organisasi memiliki terminal penjualan (*point-of-sale*) yang sangat canggih sedangkan yang lainnya menekankan kepada proses produksi atau sistem perpindahan *data* secara elektronik (*electronic data Interchane/EDI*)

1.6 Sistem Informasi Akuntansi

Sistem informasi akuntansi (SIA) pada dasarnya merupakan integrasi dari berbagai sistem pengolahan transaksi (SPT) atau sub SIA. Karena setiap SPT memiliki siklus pengolahan transaksi maka SIA juga dapat dikatakan sebagai integrasi dari berbagai siklus pengolahan transaksi. Dalam setiap pengolahan transaksi yang dilakukannya, SPT atau sub SIA menggunakan berbagai komponen yang dimilikinya seperti *hardware*, *software*, *brainware*, prosedur, *database* dan jaringan komunikasi.

Sistem informasi akuntansi pada dasarnya merupakan integrasi dari berbagai sistem pengolahan transaksi

Sistem informasi

akuntansi dapat didefinisikan sebagai kumpulan (integrasi) dari sub-sub sistem/ komponen baik fisik maupun non-fisik yang saling berhubungan dan bekerja sama satu sama lain secara harmonis untuk mengolah data transaksi yang berkaitan dengan masalah keuangan menjadi informasi keuangan.

Sesuai dengan penjelasan sebelumnya bahwa akuntansi merupakan sistem informasi dan *data* transaksi yang diolahnya merupakan *data-data* yang berkaitan dengan masalah ekonomi atau keuangan baik yang berasal dari dalam atau luar organisasi perusahaan, maka sistem informasi akuntansi dapat pula didefinisikan sebagai kumpulan (integrasi) dari sub-sub sistem/komponen baik fisik maupun nonfisik yang saling berhubungan dan bekerja sama secara harmonis untuk mengolah *data* transaksi yang berkaitan dengan masalah keuangan menjadi informasi keuangan.

Adapun komponen sistem informasi akuntansi (sistem pengolahan transaksi) secara terperinci adalah sebagai berikut:

Tabel 1.2 Komponen SPT/SIA

Sistem Pengolahan Transaksi/Sistem Informasi Akuntansi	No	Komponen	Sub komponen	Jenis
	1	Hardware	-Bagian input -Bagian pengolah/prosesor dan memori -Bagian output -Bagian komunikasi (di lihat dari fisiknya)	
2	Software	-Sistem operasi -Software aplikasi siklus penerimaan (penjualan) -Software aplikasi siklus pengeluaran (pembelian) -Software aplikasi siklus produksi -Penerimaan dan pengeluaran kas -Software aplikasi siklus GL dan laporan keuangan		Non Fisik
3	Brainware	-Manajer sistem informasi -Analisis sistem informasi -Ahli komunikasi -Administrator database -Programer -Operator		Fisik
4	Prosedur	-Rangkaian aktivitas/ transaksi dalam: • Siklus penerimaan (penjualan) • Siklus produksi dan Penggajian • Siklus pengeluaran (pembelian) • Penerimaan dan pengeluaran kas • Siklus GL dan pembuatan laporan keuangan		Non Fisik
5	Database	-Eksternal data keuangan -Konseptual data keuangan -Internal data keuangan		Non Fisik
6	Jaringan komunikasi	-Server (dilihat dari fungsinya) -Terminal “ -Network card “ -Switching Hub “ -Saluran komunikasi “		Fisik

Seperti dijelaskan diatas bahwa sistem informasi akuntansi merupakan integrasi dari sistem atau siklus pengolahan transaksi dan sistem pengolahan transaksi memiliki komponen *hardware*, *software*, *brainware*, prosedur, *database* serta teknologi jaringan komunikasi. Dari uraian diatas dapat berarti juga bahwa sistem informasi akuntansi merupakan integrasi dari komponen-komponen yang membentuk sistem/siklus pengolahan transaksi. Yang menjadi masalah, bagaimana komponen-komponen tersebut berhubungan secara harmonis membentuk sistem informasi akuntansi seperti dalam konsep sistem? Berikut ini penjelasannya.

Integrasi komponen *Hardware*

Hardware terbagi kedalam beberapa bagian seperti bagian input, bagian pengolah atau prosesor dan memori, bagian output dan bagian komunikasi. Bagian-bagian dari *hardware* tersebut satu sama lain harus berhubungan dan bekerja sama secara harmonis/berintegrasi secara harmonis/bersinergi membentuk *hardware* sistem informasi akuntansi. *Hardware* yang digunakan untuk sistem informasi akuntansi harus juga sesuai atau harmonis dengan kebutuhan sistem informasi akuntansi yang diterapkan dan kemampuan keuangan perusahaan.

Ilustrasi yang menggambarkan bagaimana integrasi yang relatif harmonis atau bersinergi terwujud antara bagian-bagian *hardware* membentuk *hardware* sistem informasi akuntansi dapat dicontohkan sebagai berikut.

Bila anda memiliki komputer dengan prosesor Intel i3,i5,i7 apakah kinerja prosesor tersebut akan optimal bila bersamanya dipasang memori dengan kapasitas 64 MB. Kerjasama antara prosesor Intel i3,i5,i7 dengan memori berkapasitas 64 MB tidak akan bisa berjalan karena kemampuan prosesor jenis Intel i3,i5,i7 sangat tinggi dan memori yang diperlukan untuk prosesor jenis Intel i3,i5,i7 minimal 2GB.Hal ini masuk akal karena menjalankan sistem operating yang beredar saat ini seperti windows 7, 8/8.1 atau windows 10 dengan memori kapasitas 64 MB tidak bisa untuk menjalankan sistem operasi tersebut. Jadi kombinasi prosesor jenis Intel i3,i5,i7 dengan Memori 64MB tidak harmonis. Prosesor tersebut akan bekerja dengan baik menjalankan aplikasi bila katakanlah ditunjang dengan memori paling tidak 2GB dengan.

Hal serupa dapat anda hubungkan kemampuan prosesor seperti contoh diatas i3,i5,i7 dengan layar monitor XVGA yang memiliki katakanlah VGA Intel HD 4400 dengan memori 1 GB anda hubungkan (padukan) prosesor yang berkinerja tinggi, layar dan VGA yang kualitas baik tapi menggunakan printer dot matrik atau laser dengan 300 DPI, kombinasi yang dibangun bukan tidak jalan tapi tidak akan memberikan hasil yang tidak optimal karena kurang sesuai (harmonis) dengan printer dan memori vga yang digunakan.

Anda dapat memiliki sistem komputer dengan spesifikasi yang tinggi untuk saat ini bila memiliki prosesor 3.0 Ghz,Layar XGA 17" digital dengan memori 64MB, Memori minimal 512 MB, Hardisk minimal 80 GB, dan printer laser 1200 DPI atau deskjet 4800 DPI. CDRW,DVD/CDROM, Network Card dan Hub 1000 base T.

Kombinasi lainnya dengan spesifikasi rendah sebagai contoh adalah sistem komputer dengan prosesor 233 MMX, memori 128 Kb, Hardisk 20 GB,Layar SVGA dengan memori 8MB, CD ROM, Laser printer 600 DPI. Network Card dan Hub 100 base T.

Spesifikasi *hardware* mana yang dipilih untuk sistem informasi akuntansi yang diterapkan harus disesuaikan dengan kemampuan keuangan, situasi,kondisi dan kebutuhan manajemen perusahaan, serta sumber daya manusia yang tersedia di perusahaan tersebut yang akan menjalankan sistem informasi akuntansi ter-

sebut. Sehingga nantinya penggunaan komputer dalam sistem informasi akuntansi benar-benar efektif. Analisis lebih jauh dapat juga melihat dari sudut kualitas komponen *hardware* dan merek komputer yang digunakan.

Integrasi komponen software

Software terbagi dua kelompok besar yaitu *software* sistem, dan *software* aplikasi. *Software* sistem terbagi lagi kedalam beberapa kelompok yaitu sistem operasi, interpreter dan kompilator. Sedangkan *software* aplikasi terbagi kedalam beberapa jenis *software* tergantung kepada aplikasi yang akan digunakan. Semua *software* di atas harus dapat terintegrasi secara harmonis (bekerja dengan baik) walaupun masalah ini untuk saat sekarang sudah jarang ditemukan karena digunakannya konsep open sistem untuk setiap sistem operasi yang beredar saat ini.

Dalam memilih *software* apa yang akan dipakai akan lebih baik anda memilih dulu sistem operasi apa yang akan digunakan sesuai dengan aplikasi yang akan dioperasikan. Karena kemampuan sistem operasi sangat banyak sedangkan pengetahuan anda sebagai pemula masih sangat terbatas maka pemikiran yang wajar bila anda menggunakan saja *software* sistem operasi yang banyak digunakan oleh orang atau perusahaan umumnya.

Anda jangan mengambil resiko dengan mencoba-coba *software* sistem operasi baru yang katanya bagus menurut sebagian orang, karena mungkin *software* tersebut kurang bagus untuk memenuhi kepentingan anda sesungguhnya. Anda boleh mencoba kalau sebagian besar orang sudah menggunakannya. Langkah ini adalah langkah yang sederhana dan murah dalam mengurangi resiko pemilihan sistem operasi bagi organisasi perusahaan.

Karena hampir semua aplikasi sistem informasi akuntansi saat ini berbasis jaringan maka sudah barang tentu anda harus memilih sistem operasi yang dapat digunakan untuk menjalankan jaringan (*network*) komputer. Dalam memilih versi berapa yang harus anda pilih, kembali anda jangan tergoda pilihlah yang kira-kira sesuai dengan kebutuhan dan kemampuan keuangan. Jangan terpengaruh dengan adanya versi baru dari merk *software* yang digunakan karena selain harganya masih mahal seringkali versi baru belum stabil atau masih ada *error*.

Integrasi Software dengan Hardware

Selain melihat orang menggunakan sistem operasi apa pada umumnya, sistem operasi yang dipakai harus disesuaikan dengan komputer yang akan menjalankannya. Jangan anda memakai sistem operasi yang paling baru katakanlah Window 10 pada komputer dengan prosesor pentium 233 MMX dengan memori 16 MB. Lihatlah di buku manual *software* sistem operasi yang biasanya mencantumkan kebutuhan minimum prosesor, memori dan hardisk yang diperlukan untuk menjalankan sistem operasi yang dipilih. Uraian diatas menunjukkan perlunya ada keharmonisan antara sistem operasi yang digunakan dengan hardwarenya.

Integrasi *software* aplikasi dengan, sistem operasi, *hardware*

Setelah anda memilih sistem operasi yang harmonis dengan situasi, kondisi, keadaan keuangan, kebutuhan manajemen dan *hardware* yang digunakan maka anda selanjutnya harus menentukan *software* aplikasi sistem informasi akuntansi semacam apa yang diperlukan dan dapat diterapkan pada sistem operasi yang digunakan. Walaupun sekarang relatif tidak banyak masalah karena digunakannya konsep *open systems* oleh hampir semua sistem operasi yang beredar saat ini akan tetapi anda harus ingat *software* baru biasanya dibangun dengan kemampuan sistem operasi baru sehingga ada resiko *software* aplikasi yang dipakai tidak sesuai dengan *softwarena*.

Selain harus sesuai dengan sistem operasi yang digunakan, *software* aplikasi juga harus sesuai dengan beban kerja yang akan dipikul oleh aplikasi tersebut. Hal ini biasanya erat sekali kaitannya dengan kemampuan *hardware* yang digunakan mulai dari *hardisknya*, prosesor,memorinya dan lain-lain.

Software aplikasi sistem informasi akuntansi yang digunakan oleh perusahaan di Indonesia biasanya tidak menggunakan *software* jadi (paket). Karena sesuai konsep sistem keharmonisan juga harus terjadi antara *brainware* dan *software* aplikasi yang digunakan.

Integrasi komponen *brainware*

Brainware adalah 'orang' yang memiliki, membangun dan menjalankan sistem informasi akuntansi. Istilah orang disini bukan sembarang orang dalam arti wujudnya orang akan tetapi orang yang memiliki kompetensi (ilmu pengetahuan dan keterampilan atau keahlian) sesuai dengan jenjangnya dalam sistem informasi. Jenjang *brainware* yang banyak digunakan antara lain:

- Manajer sistem informasi
- Analisis sistem
- Database administrator
- Ahli jaringan (*network*)
- Programer
- Operator
- Pustakawan

Integrasi komponen *brainware* mengandung arti semua jenjang *brainware* tersebut harus dapat bekerja sama secara harmonis dalam mendukung berjalannya sistem informasi akuntansi.

Integrasi antara *brainware*,*software*, dan *hardware*

Keharmonisan atau integrasi antara *hardware* dan *software* untuk saat ini tidak lagi menjadi masalah. Semua komputer dan *software* yang beredar di dunia umumnya sudah diuji terlebih dahulu kemampuan integrasinya (kompatibilitasnya) sebelum mereka dipasarkan. Sedikit masalah yang akan terjadi adalah mengintegrasikan atau menginstal *software* tertentu pada *hardware* yang tidak mendukungnya untuk bekerja secara maksimal.

Masalah utama dalam mengintegrasikan komponen SIA adalah bagaimana mengintegrasikan antara *brainware* dengan *software* dan *hardware* yang akan digunakan. Integrasi terjadi bila *hardware* dan *software* yang digunakan tersebut sesuai dengan kebutuhan *brainware* atau pengguna SIA.

Integrasi *brainware* dengan komputer relatif lebih mudah. *Brainware* tinggal memilih spesifikasi *hardware* yang dibutuhkan berdasarkan dana yang tersedia, dan pasti jalan dalam arti komputernya. Karena, bila komputer tersebut tidak jalan berarti komputer tersebut batal untuk dibeli, paling tidak ada garansi selama beberapa tahun yang menjamin akan jalan.

Lain halnya dengan *software* SIA. Pengguna tidak secara langsung akan tahu apakah *software* yang diadakan akan mampu beroperasi sesuai dengan yang diharapkan oleh semua pengguna paling tidak sampai akhir tahun?.

Karena *software* yang digunakan harus mampu menghasilkan informasi yang dibutuhkan oleh para pengguna dalam melaksanakan pekerjaan atau tugasnya, sedangkan tugas apa yang harus dilaksanakan sangat tergantung kepada perubahan yang terjadi di lingkungan perusahaan (karena perusahaan merupakan sistem terbuka) maka kebutuhan informasi para pengguna atau informasi yang harus dihasilkan oleh *software* sistem informasi akuntansi sangat dipengaruhi oleh perubahan yang terjadi di lingkungan.

Karena itu, suatu *software* sistem informasi akuntansi yang jalan sempurna saat ini mungkin beberapa bulan lagi tidak dapat beroperasi sempurna karena ada perubahan kebijakan pemerintah yang mempengaruhi bagaimana perusahaan harus bekerja.

Karena itu bagaimana cara mendapatkan *software* sistem informasi akuntansi agar dapat beroperasi selamanya?, Berikut ini beberapa alternatif pengadaan *software* sistem informasi akuntansi bagi suatu perusahaan.

□ **Membeli *software* jadi**

Software aplikasi sistem informasi akuntansi yang beredar di pasaran dibuat tidak tahu untuk kebutuhan *brainware* yang mana, jenis perusahaan apa serta budaya organisasi dan gaya manajemen atau pemakai yang bagaimana. Umumnya, *software* aplikasi yang sudah jadi tersebut pada saat diterapkan tidak sesuai dengan kebutuhan dan gaya pengguna serta situasi kondisi yang dihadapi oleh manajemen perusahaan.

Sesuai dengan konsep sistem, *software* yang diterapkan dalam sistem informasi akuntansi keberadaannya harus sesuai atau harmonis dengan keinginan manajemen atau pengguna dilihat dari segi kemampuannya, cara kerjanya, pempeleohnya, harganya dan cara menggunakannya.

Ketidak harmonisan dalam hal ini akan berakibat tidak dapat diterapkannya *software* tersebut dalam membentuk sistem informasi akuntansi atau dengan kata lain sistem informasi akuntansi yang diterapkan akan gagal. Inilah salah satu penyebab utama kegagalan utama yang umum terjadi dalam

membangun sistem informasi akuntansi di Indonesia dan di dunia.

Kebanyakan para pengembang di tanah air berfikir bahwa sistem informasi akuntansi itu adalah *software*, padahal *software* hanyalah salah satu komponen dari sistem informasi akuntansi (SIA). SIA sendiri merupakan keharmonisan dalam berintegrasinya komponen-komponen pembentuknya termasuk salah satu didalamnya adalah *software* aplikasi.

Bagaimana membangun *software* sistem informasi akuntansi yang harmonis dalam mendukung terbentuknya sistem informasi akuntansi suatu perusahaan akan dibahas pada bagian akhir buku ini.

Berdasarkan uraian diatas maka keuntungan dan kerugian membeli *software* jadi secara ringkas dapat dijelaskan diantaranya adalah sebagai berikut :

- *Keuntungannya*
 - Harganya murah
- *Kerugiannya*
 - Kebutuhan perusahaan akan informasi akuntansi dan sistem informasi akuntansi walau perusahaan sejenis secara spesifik tidak sama. *Software* yang sudah jadi tidak jelas dibuat untuk kebutuhan sistem informasi akuntansi perusahaan seperti apa. Karena *software* dibuat tidak berdasarkan kepada kebutuhan pengguna di perusahaan sehingga kemungkinan *software* itu dapat digunakan kecil.
 - Sebagian kecil dari sistem informasi akuntansi yaitu sistem informasi akuntansi keuangan dibuat berdasarkan kepada standar akuntansi keuangan/ SAK tapi sebagian besar yaitu sistem informasi akuntansi manajemen dibuat berdasarkan kebutuhan manajemen dalam melaksanakan pekerjaannya

□ **Membuat *software* sendiri**

Karyawan atau SDM suatu perusahaan adalah orang yang bekerja di perusahaan tersebut dengan pengalaman dan pengetahuan yang luas mengenai perusahaan dan bekerja dibayar sesuai dengan standar gaji perusahaan. Dalam mengadakan SIA, pengembangan SIA oleh para karyawan perusahaan akan menurunkan biaya pengembangan karena karyawan dibayar dengan tarip gaji yang biasa berlaku di perusahaan.

Hal-hal yang dapat mempersulit keberhasilan pengembangan atau pengadaan SIA suatu perusahaan adalah karena karyawan tersebut merupakan bagian dari perusahaan. Karyawan tersebut sulit untuk dapat bertindak dan bersikap sebagaimana seharusnya dilakukan dalam membangun SIA. Sebagai bagian dari sistem, apa yang dilakukan karyawan merupakan hasil interaksi dari berbagai kepentingan. Berbagai kepentingan tersebut mungkin sejalan dengan tujuan organisasi dan konsep sistem mungkin pula tidak.

Pada saat membangun SIA, bila kepentingan pihak-pihak yang berkaitan tidak sejalan dengan kepentingan organisasi perusahaan dan hal ini tidak disadari oleh karyawan pengembangnya karena berbagai alasan maka SIA yang dibangun tidak dapat mendukung aktivitas perusahaan. Dalam kondisi seperti ini SIA yang dibangun hanya mencerminkan kekuasaan dari pihak-pihak tertentu, tidak mencerminkan kebutuhan objektif perusahaan.

Secara ringkas keuntungan dan kerugian tersebut diantaranya dapat diringkas sebagai berikut:

- *Keuntungannya*
 - Biaya murah
 - Karyawan akan lebih menguasai dan memahami masalah yang dihadapi perusahaannya
 - Tidak tergantung kepada pihak lain
 - Bila terjadi masalah atau perlu revisi mudah dilakukan
- *Kerugiannya*
 - Karyawan pembuat *software* merupakan bagian dari sistem dan budaya yang berlaku di perusahaan. Sedangkan sistem merupakan perpaduan dari berbagai kepentingan yang ada di perusahaan berdasarkan budaya yang berlaku. Sebab itu, sulit bagi pembuat *software* SIA untuk secara objektif atau seharusnya melakukan tugas karena setiap tindakan yang dilakukan akan berdampak kepada kepentingan pihak lain di mana pada akhirnya akan berdampak pula kepada karyawan yang bersangkutan.
 - Karyawan internal sering kurang memiliki wawasan yang luas serta sesuai dengan perkembangan dunia saat ini dalam memecahkan permasalahan ke dalam suatu *software* SIA.
 - *Software* SIA yang dibuat oleh karyawan hanya dapat diperbaiki oleh karyawan itu sendiri sehingga akan tampak *software* itu hanya akan bekerja bila ada karyawan pembuat *software*. Seringkali bila karyawan pembuat *software* tidak ada, perbaikan tersebut tidak dapat dilakukan dan SIA tidak dapat beroperasi.
- **Bantuan pihak eksternal (*outsourcing*)**
 - *Keuntungannya*

Bila tidak salah pilih konsultan, karena banyak konsultan di Indonesia dalam bidang TI lebih berperan sebagai makelar yang tidak memiliki pengetahuan apa-apa tentang sistem informasi, maka bila menggunakan pihak eksternal SIA yang dibuat akan:

 - Independen
 - Objektif
 - Lebih berkualitas
 - *Kerugiannya*

Bila tidak menggunakan prinsip, metode dan teknik yang tepat maka :

- Akan banyak waktu terbuang untuk memahami sistem berjalan, masalah yang dihadapi dan spesifikasi sistem informasi serta informasi yang dihasilkannya.
- Akan muncul gap komunikasi antara konsultan sebagai pengembang sistem dengan pengguna sistem sehingga sistem informasi akuntansi yang dibuat dan informasi akuntansi yang dihasilkannya tidak sesuai dengan yang diharapkan sehingga SIA yang dibuat gagal untuk dioperasikan.
- Membangun SIA akan memakan biaya yang mahal sekali

Dari ketiga cara pengadanan *software* tadi maka seperti kebanyakan dilakukan oleh perusahaan di negara-negara yang sudah maju pengembangan *software* sistem informasi akuntansi banyak dilakukan oleh pihak eksternal. Walaupun tidak menjamin sukses tapi lebih memberikan harapan dibandingkan dengan apabila *software* tersebut dibuat sendiri apalagi kalau beli sudah jadi.

Sulitnya mendapatkan *software* SIA yang operasional disebabkan karena *software* bukan satu-satunya komponen SIA. Walaupun *software* SIA secara programming jalan akan tetapi secara sistem belum tentu. Karena komponen SIA selain *software* adalah *hardware*, *brainware*, prosedur, *database* dan jaringan komunikasi.

Hardware, *software* dan jaringan komunikasi sebagai teknologi informasi memberikan kepastian untuk jalan, kalau dipabriknya sudah jalan di Indonesiapun pasti jalan. Lain halnya dengan *brainware*, prosedur dan *database*. Sangat tergantung kepada situasi, kondisi dan budaya dimana mereka digunakan.

Agar *software* aplikasi SIA dapat digunakan maka selain *software* tersebut harus harmonis dengan *hardware* dan teknologi jaringan komunikasi yang digunakan maka *software* tersebut juga harus sesuai dengan kebutuhan atau keinginan pengguna. Agar sesuai dengan konsep sistem maka *software* aplikasi yang digunakan harus merupakan integrasi dari *software-software* aplikasi yang digunakan di berbagai bagian organisasi. Konsep *software* terintegrasi ini oleh Gelinas dan Sutton (2002) disebut sebagai *Enterprise Resource Planning* (ERP).

Integrasi komponen Prosedur

Prosedur adalah rangkaian aktivitas yang menghubungkan aktivitas satu dengan aktivitas yang lainnya. Prosedur bila dapat diprogram berubah menjadi *software* misalnya prosedur posting dan jurnal kalau sudah pasti dapat diprogram secara otomatis sehingga pemakai komputer tidak perlu lagi melakukan penjurnalan dan posting. Prosedur seringkali digunakan sebagai alat kontrol satu transaksi oleh transaksi lainnya yang terjadi di bagian yang sama atau dengan bagian lain dan disebut sebagai *internal ceck*.

Dalam sistem informasi akuntansi atau pengolahan transaksi prosedur mengatur langkah-langkah yang harus dilakukan dalam siklus pengolahan transaksi. Antara prosedur satu dengan prosedur lainnya harus terjalin kerjasama yang harmonis atau bersiner-

gi. Jangan sampai satu prosedur berjalan sendiri dan tidak terkait dengan prosedur yang lain.

Prosedur merupakan pedoman yang harus di ikuti dan dibentuk atas dasar kesepakatan dengan penetapan dan pengesahan dilakukan oleh manajemen yang berwenang. Kesalahan prosedur berarti melanggar aturan perusahaan. Di dalam pemerintahan kesalahan prosedur sering dijadikan dalih agar terhindar dari jeratan hukum. Dalam konsep sistem kesalahan prosedur berarti melanggar sistem atau hukum yang berlaku di perusahaan tersebut.

Integrasi prosedur dengan *brainware*, *software* dan *hardware*

Prosedur merupakan rangkaian aktivitas atau transaksi. Aktivitas pada dasarnya segala sesuatu yang dilakukan oleh *brainware* berdasarkan informasi yang masuk dan persepsi tentang informasi tersebut. Jadi agar suatu prosedur dijalankan oleh karyawan (*Brainware*) secara harmonis atau bersinergi satu sama lain maka persepsi yang dimiliki oleh karyawan yang menjalankan prosedur tersebut satu sama lain harus harmonis. Mengharmoniskan pelaksanaan prosedur salah satunya melalui sosialisasi atau pelatihan tentang prosedur tersebut.

Prosedur merupakan komponen dalam SIA, prosedur yang diterapkan harus sesuai juga dengan *hardware*, *Software*, *Database* dan Teknologi jaringan komunikasi yang digunakan.

Prosedur yang berkaitan dengan *hardware*, *software* dan *teknologi jaringan komunikasi* pada dasarnya hampir sama yaitu mulai dari prosedur pengadaan *hardware* dan *software* tersebut, instalasi dan pengoperasiannya. Sedangkan prosedur penyusunan *database* perusahaan dimulai dengan menentukan kebutuhan informasi para pengguna sistem informasi akuntansi, prosesnya serta data-data yang diperlukan. Karena informasi yang diperlukan bagi organisasi perusahaan selalu berubah sesuai dengan perubahan yang terjadi di lingkungan maka *data* yang diperlukanpun selalu tidak sama antara waktu yang satu dengan waktu lainnya .

Integrasi komponen *database*

Database dalam arti luas merupakan *data-data* yang ada di perusahaan sedangkan dalam arti sempit *database* merupakan *data-data* yang ada di dalam komputer.

Data-data dalam *database* (arti sempit) satu sama lain harus berhubungan secara harmonis (berintegrasi). Integrasi yang terjadi tergantung kepada informasi apa yang diperlukan.

Untuk melihat bagaimana *data* dalam *database* berintegrasi sering ditunjukkan dalam bentuk entity relationship diagram (ERD), Hierarki diagram atau *network* diagram.

Integrasi *database* dengan *hardware*, *software*, *brainware* dan prosedur

Database yang dibangun harus dihitung perkiraan volumenya. Dari perkiraan volume tersebut anda paling tidak dapat menentukan kapasitas *hardisk* yang dibutuhkan dan tipe prosesor yang cocok

untuk menangani *data* sejumlah yang diperkirakan. Dari perkiraan volume tersebut anda juga dapat menentukan *database management systems* (DBMS) mana yang cocok. Para pakar teknologi informasi banyak yang mengelompokkan DBMS ini berdasarkan kapasitasnya kedalam dua kelompok yaitu DBMS untuk kelas kecil sampai menengah dan untuk kelas besar.

DBMS untuk kelas kecil contohnya Clipper, Visual Basic, Visual FoxPro, Delphi, Power Builder dan lainnya. Sedangkan DBMS untuk kelas besar (berat) contohnya Oracle, Sybase IQ, Teradata, Daytona, DB2, SQL Server 2000, Informik, Ingres, SQL for window dan lainnya.

DBMS kelas mana yang anda perlukan atau yang diperlukan oleh suatu perusahaan? DBMS kelas menengah umumnya mampu menangani *data* sampai dengan 2 Giga Byte (GB) atau satu milyar record. Kalau misalnya perhitungan secara kasar satu record *data* master seperti persediaan, karyawan, pelanggan dan lain lain besarnya 200 karakter dan *data* transaksi misalnya tanggal, No, User Id, Lokasi, Kode terminal, Kode barang (rata-rata 20 item barang), unit yang dibeli dan lain-lain besarnya 100 karakter maka kira-kira DBMS tersebut dapat menangani *data* Master sebanyak 10 juta (2GB:200) atau *data* transaksi sebanyak 20 Juta.

Bila anda ambil contoh perusahaan yang dianalisis adalah perusahaan dagang eceran skala besar katakanlah *hyper mall* dan perusahaan tersebut menggunakan 40 point of sales (POS) untuk menangani transaksi dengan periode akuntansi satu tahun dan penjualan semuanya dilakukan secara kas maka anda dapat menghitung berapa jumlah transaksi yang dapat ditangani oleh POS per menitnya yaitu dengan cara membagi 20 juta dengan 365 (hari), dengan 14 (jam kerja), dengan 60 (jam/menit). Hasilnya adalah 1,65. Artinya DBMS skala kecil menengah masih dapat menangani lebih dari satu transaksi permenit untuk perusahaan yang memiliki ratusan cabang, 40 POS setiap cabang dengan jumlah macam item rata-rata yang dibeli 20 macam item. anda tinggal menentukan apakah *data* master perusahaan tersebut lebih dari 20 juta atau kurang, bila lebih anda harus menggunakan DBMS kelas besar. Apakah setiap terminal di ratusan cabang dan 40 POS rata-rata melakukan transaksi lebih dari satu setiap menit atau 40.000 secara bersamaan diseluruh cabang. Bila lebih maka aplikasi anda (*Front End*) dapat menggunakan DBMS skala besar sebagai back End dengan konsep *client-server*.

DBMS skala besar 32/64 bit generasi baru dapat menhandel *data* sampai dengan ratusan tera byte (TB) seperti Oracle 10g, SQL Server Yukon dan lain-lain. Rekord tahun 2003 menurut survey yang dilakukan oleh Winter Corporation di seluruh dunia menunjukkan penggunaan database terbesar dengan berbagai sistem operasi dipegang oleh Oracle (29,232 TB) di perusahaan France Telecom, sedangkan ukuran data hasil pengolahan data secara on-line terbesar dipegang oleh DB2 (18,301 TB) untuk perusahaan Land Registry dan pengolahan data paling banyak perdetiknya

(paling sibuk) untuk aplikasi berbasis window dipegang oleh Sql Server (4010 transaksi per detik) di perusahaan Anonymous.

Dengan ukuran kemampuan sampai ratusan terabyte maka anda sudah tidak perlu susah-susah lagi menghitung ukuran, selanjutnya yang penting diperhatikan adalah kemudahan, ketahanan, kecepatan, integritas dan keamanan dari DBMS yang akan digunakan.

Database yang digunakan (isi dan hubungannya) juga harus sesuai dengan kebutuhan pemakai. Menentukan kebutuhan *data* dalam *database* bagi pemakai dilakukan dengan menentukan terlebih dahulu informasi yang diperlukan. Berdasarkan kebutuhan informasi tersebut maka harus diketahui bagaimana proses untuk menghasilkan informasi tersebut. Berdasarkan dua hal ini maka dapat diketahui *data* apa yang harus dimasukkan dan bagaimana hubungan antar *data-data* yang dimasukkan agar informasi mudah untuk di akses.

Database yang digunakan juga harus sesuai atau ditunjang oleh prosedur yang cocok. Seperti anda ketahui bahwa kualitas informasi yang dihasilkan sangat tergantung kepada kualitas data yang dimasukkan. Bila *data* dimasukkan akurat, relevan, tepat waktu dan lengkap maka informasi pun akan akurat, tepat waktu, relevan dan lengkap. Agar *data* yang tersimpan dalam *database* akurat, relevan, tepat waktu dan lengkap maka harus ditunjang oleh prosedur untuk menjamin bahwa *data* yang dimasukkan tersebut telah memenuhi kriteria yang diharapkan. Prosedur biasanya diterapkan dalam bentuk prosedur pengendalian yang akan dipelajari pada bab 5.

Integrasi teknologi Jaringan telekomunikasi

Jaringan telekomunikasi (*network*) atau disebut juga sebagai jaringan komunikasi *data*. Pada dasarnya merupakan penggunaan media elektronik atau sinar untuk memindahkan *data* dari satu lokasi ke satu atau beberapa lokasi yang lain. Komponen-komponen yang digunakan dalam jaringan komunikasi *data* satu sama lain harus berintegrasi secara harmonis atau bersinergi membentuk jaringan komunikasi *data* dalam SIA. Komponen jaringan komunikasi *data* yang harus bersinergi misalnya antara Hub yang digunakan, saluran komunikasi dan *network card* (*LAN card*). Bila LAN akan menggunakan jaringan komunikasi *data* dengan kecepatan 10/100/1000 MB/detik maka semua komponen tersebut harus mendukung kecepatan tersebut.

Integrasi teknologi jaringan komunikasi dengan *hardware*, *software*, *brainware*, prosedur, dan *database*,

Teknologi jaringan komunikasi yang digunakan harus juga harmonis dengan komponen lainnya. Keharmonisan teknologi yang digunakan dalam jaringan komunikasi (*network*) harus sesuai dengan *hardware* yang digunakan. Misalnya bila *server* suatu perusahaan menggunakan prosesor Intel Xeon atau iP4 3,06 atau AMD Opteron dengan iP4.1.8 atau AMD Athlon 1800 dengan spesifikasi

kasi lainnya yang sesuai maka teknologi jaringan komunikasi yang digunakan juga harus menunjang kepada kecepatan *server* dan terminal yang digunakan. Jangan anda gunakan LAN card Arch net (Kecepatan 2,5 MB/Detik dengan topologi BUS untuk mengintegrasikan *server* dan terminal yang memiliki kinerja tinggi. Gunakanlah komponen jaringan yang mendukung kecepatan transfer *data* 100 atau 1000 MB/detik.

Teknologi jaringan komunikasi yang digunakan juga harus sesuai dengan *software* sistem operasi yang digunakan. Karena itu, *hardware* komponen yang digunakan biasanya selalu dilengkapi dengan *softwrenya* agar harmonis dengan sistem operasi yang digunakan.

Keharmonisan antara teknologi jaringan komunikasi yang digunakan harus juga sesuai dengan kebutuhan atau kemampuan *brain ware* yang menjalankannya. Bagi sebagian user (*brainware*) kecepatan jaringan mungkin tidak perlu terlalu cepat, asal bisa mentranfer *data* selama beberapa jam sudah cukup karena user tersebut biasanya memindahkan *data* dari beberapa cabang dilakukannya dalam beberapa hari. Tapi bagi user yang lain mungkin memerlukan kecepatan transfer *data* yang tinggi sehingga bisa dibanngun sistem yang *real time*.

Keharmonisan antara teknologi jaringan komunikasi dengan prosedurnya harus harmonis. Salah satu tujuan penggunaan Jaringan teknologi komunikasi yaitu untuk mempercepat transfer *data* dari satu lokasi ke beberapa lokasi lain. Jangan sampai prosedur yang digunakan untuk mengoperasikannya justru memperlambat proses yang terjadi. Misalnya dalam menggunakan sistem keamanan jaringan, sistem kewenangan menghkses *data* dan lainnya.

Jaringan teknologi komunikasi *data* yang digunakan juga harus sesuai dengan *data* yang akan didistribusikan. Gunakanlah jaringan komunikasi *data* yang berkecepatan tinggi bila *data* yang akan didistribusikan sangat banyak.

Rangkuman

Sistem pengolahan transaksi memproses *data* yang berasal dari kejadian atau peristiwa ekonomi sebagai akibat dari operasi atau aktivitas internal perusahaan dan sebagai akibat dari aktivitas perusahaan dengan pihak luar menjadi informasi keuangan/informasi akuntansi keuangan yang berguna bagi manajemen ditingkat operasional dan pihak lain

Siklus pengolahan transaksi merupakan prosedur atau urutan-an subsistem pengolahan transaksi untuk menunjang siklus transaksi dalam aktivitas bisnis perusahaan atau siklus aktivitas bisnis perusahaan, sehingga anda dapat mengatakan bahwa sik-

lus aktivitas bisnis perusahaan direpresentasikan oleh siklus/ sistem pengolahan transaksi

Sistem informasi akuntansi pada dasarnya merupakan integrasi dari berbagai sistem atau siklus pengolahan transaksi. Sistem pengolahan transaksi yang merupakan subsistem informasi akuntansi ada di berbagai fungsi operasional organisasi karena itu sistem informasi akuntansi merupakan bagian terbesar dari sistem informasi manajemen. walaupun sistem informasi akuntansi mengadopsi konsep informasi yang berkualitas akan tetapi bobot aktivitasnya lebih banyak berorientasi kepada pengolahan *data*.

Sistem informasi akuntansi tidak hanya mengolah *data* dalam jumlah besar yang meliputi aktivitas pengumpulan *data*, pengolahan, penyimpanan dan dokumentasi serta pelaporan untuk kepentingan internal dan eksternal perusahaan akan tetapi sistem informasi akuntansi juga berfungsi sebagai pedoman serta pengendali terhadap bagaimana pendokumentasian tersebut harus dilakukan oleh suatu organisasi, baik itu organisasi berorientasi laba atau bukan.

Soal

1. Coba jelaskan apa itu sistem informasi akuntansi
2. Coba jelaskan apa itu sistem pengolahan transaksi
3. Coba jelaskan apa itu siklus pengolahan transaksi
4. Coba sebutkan berbagai siklus pengolahan transaksi
5. Coba sebutkan aktivitas pengolahan transaksi

Tugas

1. Coba jelaskan mengapa dikatakan sistem informasi akuntansi sebagai bagian terbesar dari sistem informasi manajemen.
2. Coba jelaskan bagaimana hubungan sistem informasi akuntansi dengan sistem informasi manajemen.
3. Coba jelaskan dan berikan gambaran bagaimana komponen-komponen sistem informasi akuntansi atau sistem/siklus pengolahan transaksi berintegrasi membentuk sistem informasi akuntansi.
4. Coba jelaskan apakah masalah yang dihadapi oleh sistem informasi akuntansi masalah terstruktur atau bukan? Mengapa?
5. Apakah sistem informasi akuntansi bisa dibangun tanpa memahami konsep dan praktik akuntansi yang berlaku dan bagaimana faktor budaya dapat mempengaruhi penerapan sistem informasi akuntansi di Indonesia.

2

Analisis dan Perancangan Sistem Informasi Akuntansi

Pendahuluan

Seperti kita ketahui penggunaan komputer dalam aktivitas bisnis yang ditunjang dengan tersedianya berbagai *software* telah menjanjikan bahwa berbagai transaksi akuntansi dalam bisnis dan penyediaan informasi akuntansi dapat dilakukan lebih mudah, efektif dan efisien. Pada awalnya para pelaku bisnis hanya mengandalkan pemecahan masalah dengan bantuan teknologi komputer semata. Saat itu teknisi dan programmer menjadi andalan bagi para pelaku bisnis untuk membangun suatu sistem informasi akuntansi yang mampu memecahkan masalah-masalah keuangan mereka. Tetapi disisi lain, para pelaku bisnis ini seringkali kurang memahami kelebihan dan keterbatasan teknologi komputer, sehingga pada saat penggunaannya mereka sering mengalami kesulitan. Sementara itu teknisi dan programmer sebagai pembuat program sering kurang memahami masalah akuntansi dalam bisnis, yang akhirnya menimbulkan kesenjangan komunikasi antara para bagian akuntansi dan para pelaku bisnis lainnya sebagai pengguna komputer dan para teknisi/programmer sebagai pembuat *software* aplikasi sistem informasi akuntansi.

Kesenjangan komunikasi antara bagian akuntansi dan para pelaku bisnis sebagai pengguna komputer dengan pembuat program aplikasi sistem informasi akuntansi dapat dijumpai dengan adanya seorang analis sistem informasi/sistem informasi akuntansi. Seorang analis sistem informasi akuntansi dapat membantu meningkatkan kemampuan pemrosesan suatu sistem informasi akuntansi berbasis komputer. Analis sistem informasi akuntansi juga dapat membantu mengidentifikasi masalah-masalah yang dihadapi dalam penggunaan komputer untuk aplikasi sistem informasi akuntansi dalam menunjang kegiatan bisnis. Saat ini para analis sistem informasi akuntansi tidak hanya sebagai perantara yang menjembatani masalah yang dihadapi oleh bagian akuntansi dan para pengguna komputer dengan teknisi/programernya. Dengan metode yang akan diuraikan pada bab ini, seorang analis sistem informasi akuntansi akan bekerja dengan lebih produktif lagi.

2.1 Analis dan Analisis Sistem

Analisis sistem bertugas untuk menganalisis sistem informasi yang telah ada, mengembangkannya dan menyusun sistem baru pada sub sistem yang bermasalah dengan bantuan komputer.

Tugas utama analisis sistem :

- Menganalisis
- Merancang
- Menerapkan dan
- Memelihara sistem informasi yang disusunnya.

Pada saat kita sakit dokter akan memberikan obat sebagai alat untuk menyembuhkan. Ketika kendaraan kita mengalami masalah, bengkel akan mengirim teknisi untuk memperbaikinya. Jika perusahaan menginginkan aktivitas usahanya bekerja dengan lebih produktif dengan tingkat efisiensi dan efektivitas yang lebih baik, maka komputer dijadikan andalan untuk memecahkannya. Jadi obat, teknisi dan komputer adalah alat yang dapat memecahkan masalah-masalah yang kita hadapi.

Dalam suatu sistem yang lebih kompleks, seperti sistem informasi akuntansi dalam suatu organisasi atau perusahaan, maka masalah yang timbulpun akan lebih kompleks lagi, karena satu saja sub sistem mengalami masalah akan berdampak pada aktivitas sub sistem yang lainnya. Pada perusahaan-perusahaan yang mengandalkan otomatisasi pekerjaan pada komputer, diperlukan seorang tenaga ahli yang harus mampu memahami akar permasalahan yang dihadapi perusahaan dan sekaligus memberikan jalan keluar atas permasalahan tersebut.

Analisis sistem adalah orang yang bertanggung jawab untuk mempelajari informasi yang berhubungan dengan masalah-masalah yang timbul dan mampu memberikan jalan keluar sesuai dengan masalah yang dihadapi.

Tugas utama dari seorang analisis sistem ini adalah menganalisis sistem yang telah ada, mengembangkannya dan menyusun sistem baru terutama pada sub sistem yang bermasalah dengan bantuan teknologi komputer. Kunci utama yang perlu diperhatikan adalah mengkombinasikan antara hasil analisisnya dengan teknologi komputer sehingga dapat menjelaskan bagaimana sumber daya manusia yang dimiliki perusahaan dengan metodologi yang tersedia dan teknologi komputer yang dimiliki dapat memberikan hasil yang terbaik dalam meningkatkan aktivitas perusahaan. Rancangan pengumpulan *data*, pemasukan, pemrosesan dan penyimpanan *data* di komputer yang dilakukan analisis sistem haruslah disajikan secara efektif sehingga dapat memberikan informasi yang bermanfaat dan tepat waktu bagi pengguna komputer.

Tugas seorang analisis sistem bukan saja menganalisis dan mendesain sistem informasi akuntansi, tetapi lebih dari itu ia haruslah mampu menyajikan satu sistem informasi akuntansi yang terpadu. Analisis sistem juga menawarkan suatu perubahan dengan mengembangkan teknologi terbaru yang dapat dimanfaatkan oleh suatu perusahaan.

Dengan uraian tugas dan tanggung jawab seperti di atas, maka seorang analisis sistem haruslah orang yang memiliki pengetahuan yang terpadu antara aktivitas bisnis, sistem informasi dan teknologi. Analisis sistem bukanlah seorang programmer yang ditugaskan/merasa mampu membuat program mutakhir dengan komputer untuk menyelesaikan masalah. Seorang programmer komputer belum tentu dapat melakukan analisis masalah yang dihadapi oleh perusahaan, seperti yang harus dilakukan dalam penyusunan sistem informasi akuntansi, suatu sistem informasi yang mem-

berikan informasi tentang aktivitas keuangan perusahaan. Dalam menyusun sistem informasi akuntansi suatu perusahaan diperlukan orang yang mampu memahami apa itu sistem informasi akuntansi, masalah-masalah yang dihadapi dalam sistem informasi akuntansi perusahaan tersebut dan mampu memberikan solusi serta menggabungkan solusi tersebut dengan bantuan teknologi komputer.

Ada banyak istilah bagi analis sistem saat ini, seperti desainer sistem, pengembang sistem, konsultan sistem, konsultan manajemen, analis operasi, analis informasi, analis bisnis, dan *knowledge engineer* untuk sistem pakar tetapi yang paling sering digunakan di Indonesia adalah analis sistem. Sebagai gambaran lebih lengkap selanjutnya akan diuraikan tugas-tugas dari seorang analis sistem.

2.2 Tugas dan Tanggungjawab Analis Sistem Informasi Akuntansi

Untuk menjaga agar setiap bagian dalam perusahaan tidak tumpang tindih dalam melaksanakan tugas dan tanggungjawabnya, maka manajemen perusahaan perlu menguraikan tugas dan tanggungjawab dari masing-masing bagian. Uraian tugas yang jelas dan terinci juga dapat membedakan fungsi dari masing-masing bagian.

Di perusahaan-perusahaan besar keberadaan analis sistem pada umumnya berada pada divisi pengembangan dibawah tanggungjawab manajer pusat pengembangan Tugas dari analis sistem ini berbeda-beda pada setiap jenis dan besarnya perusahaan, tetapi secara umum tugas dan tanggung jawab analis sistem adalah seperti yang diuraikan pada gambar 2.1.

Selain tugas dan tanggung jawab seperti yang diuraikan pada gambar 2.1, seorang analis sistem juga harus memiliki kemampuan berkomunikasi dengan pihak-pihak lain seperti pemakai komputer, manajemen, teknisi, bagian administrasi, programmer, penyedia *hardware* dan *software* dan *database administrator*.

Untuk mendukung pekerjaannya, seorang analis sistem juga harus memiliki kualifikasi khusus dalam bidang pendidikan. Ia harus seorang sarjana atau master dalam bidang komputer, bisnis, ataupun teknik industri. Pengalaman dalam penyusunan program sangat diperlukan, yang ditunjang dengan pelatihan dan pengalaman dalam aktivitas dan sistem bisnis. Pelatihan terhadap seorang analis sistem tentunya akan mendukung pekerjaannya, terutama pelatihan yang memberikan pengetahuan tentang metode-metode yang digunakan dan struktur-struktur sistem yang ada. Kemampuan berkomunikasi baik secara lisan dan tertulis merupakan nilai tambah yang tidak dapat diabaikan bagi seorang analis.

Seorang analis sistem walaupun ia seorang sarjana, perlu memiliki pengalaman dalam mengikuti pelatihan mengenai stan-

Analisis sistem harus memiliki kemampuan berkomunikasi dengan pihak-pihak lain seperti pemakai komputer, manajemen, teknisi, bagian administrasi, programmer, penyedia *hardware* dan *software* dan *database administrator*

dar pengembangan sistem seperti standar sistem *database*, komunikasi *data*, pengembangan sistem, metode-metode pengembangan sistem, dan memahami tentang bagaimana menyusun studi kelayakan sistem dilihat dari segi biaya dan manfaat yang akan diperoleh dari sistem yang disusun.

Gambar 2.1 Uraian tugas dan tanggungjawab analis sistem

URAIAN TUGAS ANALIS SISTEM

Bertanggung jawab kepada : Manajer Pusat Pengembangan

Tugas utama:

1. Mengumpulkan dan menganalisis *data* untuk mengembangkan sistem informasi. Analisis sistem bertanggungjawab dalam mempelajari masalah-masalah dan kebutuhan kebutuhan organisasi/perusahaan untuk menentukan bagaimana teknologi komputer, prosedur-prosedur dan sumber daya manusia dapat bersama-sama memecahkan masalah dan mengembangkan sistem yang ada secara terpadu.
2. Mendesain sistem dan metode untuk dikomputerisasikan berdasarkan sistem informasi yang ada dan memberikan petunjuk penggunaannya.
3. Mempertanggungjawabkan temuan-temuan, rekomendasi-rekomendasi dan spesifikasi-spesifikasi secara formal baik lisan maupun dalam laporan resmi.

Tanggung jawab:

1. Melakukan evaluasi proyek
2. Menganalisis sistem yang ada untuk mengetahui masalah yang ada dan kemungkinan pemecahannya.
3. Mendefinisikan pengembangan atau perubahan sistem bila diperlukan.
4. Mengevaluasi alternatif pemecahan masalah
5. Memilih perangkat keras dan perangkat lunaknya (dengan persetujuan atasan).
6. Mendesain, alur dan prosedur sistem baru.
7. Melakukan supervisi untuk penerapan sistem baru.

Tugas-tugas:

1. Memperkirakan kebutuhan personil, anggaran biaya dan jadwal pembuatan sistem.
2. Mengembangkan dan menerapkan rencana pengembangan sistem sesuai dengan standar baik
3. Melakukan pengumpulan *data* melalui wawancara dan cara lainnya.
4. Menganalisa dan mendokumentasikan sistem yang telah berjalan.
5. Merumuskan perlengkapan teknologi terbaru untuk menangani masalah-masalah perusahaan.
6. Mempelajari pengetahuan teknologi manajemen yang akan menggunakan sistem.
7. Melakukan evaluasi terhadap berbagai teknologi yang mungkin digunakan berdasarkan pertimbangan teknis, operasi dan ekonomi.
8. Melakukan review terhadap sistem baru yang akan diajukan untuk persetujuan.
9. Membuat desain dan melakukan uji coba protipe sistem baru
10. Mendesain struktur *data* dan *file*
11. Mendesain *input*, *output* dan bahasa yang akan dipergunakan dalam sistem komputerisasi.
12. Mendesain teknik dan bentuk pengumpulan *data*.
13. Mendesain kontrol dan pengamanan sistem.
14. Mempersiapkan spesifikasi penerapan program.
15. Menerapkan, melakukan uji coba dan mengintegrasikan program.
16. Melakukan supervisi penerapan program

Melihat tanggungjawab, tugas-tugas, persyaratan pendidikan dan pengalaman serta wawasan yang harus dimiliki oleh seorang analis sistem, maka jelaslah bagi kita bahwa seorang analis sistem diharapkan dapat menjadi orang yang mampu memecahkan masalah-masalah yang dihadapi perusahaan terutama dalam mengatasi kurangnya manajemen yang memahami teknologi komputer dan penerapannya dalam aktivitas bisnis. Tugas memecahkan masalah ini menjadi kunci utama dari suksesnya karir seorang analis sistem. Jenjang karir seorang analis sistem adalah sebagai analis sistem pemula, analis sistem, analis sistem senior, dan kepala bagian analis. Bagaimana peranan analis sistem sebagai pemecah masalah, selanjutnya akan diuraikan secara lebih rinci berikut ini.

2.3 Perbedaan Tugas dan Tanggungjawab Analis Sistem dan Programmer

Seorang analis sistem informasi mempunyai tugas dan tanggungjawab yang berbeda dengan seorang programmer. Seorang analis sistem dari uraian tugas di atas diketahui bertanggungjawab atas pelaksanaan analisis sistem, merancang sistem, menerapkan sistem dan mendukung sistem berbasis komputer untuk aplikasi bisnis. Berikut ini salah satu uraian tugas seorang analis sistem secara garis besar bila menggunakan metode SDLC.

Analisis Sistem

Analisis sistem adalah studi tentang sistem bisnis yang sedang berjalan dan permasalahannya, menentukan kebutuhan aktivitas bisnis dan permintaan-permintaan pemakai sistem dan melakukan evaluasi terhadap berbagai alternatif solusi.

Perancangan Sistem

Perancangan sistem, adalah spesifikasi umum dan terinci dari pemecahan masalah berbasis komputer yang telah dipilih selama tahap analisis. Spesifikasi perancangan umumnya dikerjakan oleh programmer agar sistem informasi yang dirancang dapat diterapkan.

Penerapan Sistem

Penerapan sistem, adalah menerapkan sistem dalam operasi organisasi. Program komputer diterapkan dan diuji coba, manajer dan pemakai sistem dilatih untuk menggunakan sistem baru dan operasi organisasi dikonversikan pada sistem baru.

Dukungan Sistem

Dukungan sistem, adalah tindak lanjut yang diberikan terhadap sistem informasi akuntansi yang telah diterapkan dalam operasi perusahaan. Kegiatan ini mencakup pemeliharaan program dan meningkatkan kemampuan sistem.

Tugas programmer hanya berhubungan dengan penyusunan Program yang digunakan dalam pengembangan sistem informasi.

Karakteristik programmer berbeda dengan analis sistem. Secara umum karakteristik programmer adalah sebagai berikut:

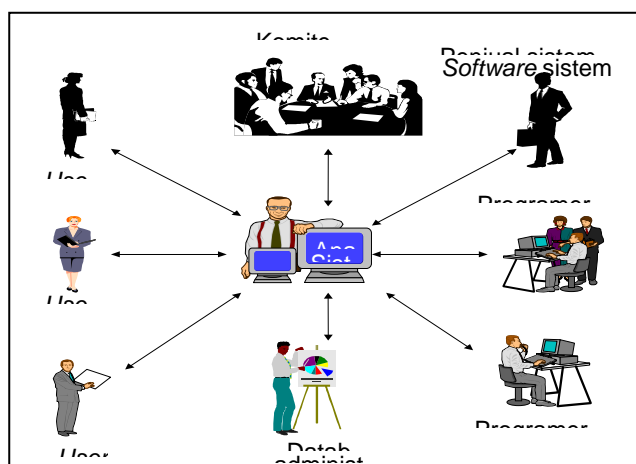
- Programmer hanya bertanggungjawab terhadap program komputer yang meliputi komputer, mengoperasikan program dan kelengkapannya serta bahasa pemrograman yang digunakan.
- Pekerjaan programmer suatu hal yang pasti, penilaiannya berkisar pada benar atau tidaknya instruksi-instruksi dan logika program.
- Pekerjaan programmer tidak banyak membutuhkan hubungan dengan pihak lain, umumnya hanya terbatas dengan sesama programmer dan analis sistem yang menyiapkan spesifikasi program.

Perbedaan tanggungjawab antara programmer dan analis sistem dapat diidentifikasi sebagai berikut:

- Sistem analis tidak hanya berhadapan dengan program komputer, ia juga bertanggungjawab dalam pemilihan perangkat komputer, orang-orang yang akan menggunakan sistem informasi akuntansi yang disusunnya, prosedur-prosedur sistem dan *file/database* sistem yang digunakan.
- Kerja yang dihasilkan analis sistem bukan merupakan hal yang pasti. Ada banyak kemungkinan jawaban betul atau salah. Solusi sistem merupakan suatu hal yang dapat dirundingkan.
- Hubungan yang harus dijalin oleh sistem analis lebih luas dan lebih kompleks. Analis sistem harus mampu menjalin hubungan dengan klien bisnisnya, pihak manajemen, programmer, manajer sistem informasi, auditor dan penjual sistem informasi akuntansi (kalau beli) yang masing-masing memiliki kepentingan yang berbeda-beda.

Untuk memberikan gambaran tentang hubungan yang harus dijalin oleh seorang analis sistem, dapat dilihat pada gambar 2.2. pada halaman berikut ini.

Gambar 2.2 Hubungan antara analis sistem dan pihak lain



2.4 Analisis Sistem Sebagai Pemecah Masalah

Kita telah mengetahui bahwa seorang analis sistem bertanggung jawab untuk mengembangkan sistem informasi melalui pengumpulan dan pelaksanaan analisis *data*, dengan tujuan untuk memperoleh solusi atas masalah-masalah yang dihadapi oleh suatu organisasi/perusahaan. Bagaimana peranan seorang analis sistem dalam memecahkan masalah-masalah organisasi/perusahaan berikut ini uraiannya.

Organisasi berada dalam lingkungan yang dinamis, dan dinamika itu sering mendorong terjadinya perubahan pada organisasi. Perubahan-perubahan umumnya terjadi karena berubahnya peraturan pemerintah, berubahnya tingkat dan jenis persaingan, adanya teknologi baru baik dalam bidang informasi maupun bisnis dan berubahnya aktivitas bisnis. Penyesuaian terhadap adanya perubahan tersebut, menuntut adanya sistem baru, baik dari hasil pengembangan maupun penyusunan sistem informasi akuntansi baru. Salah satu tahap dalam pengembangan sistem informasi akuntansi baru bagi suatu organisasi adalah melakukan analisis sistem, yaitu suatu aktivitas mempelajari masalah dan menemukan pemecahan yang terbaik bagi masalah yang dihadapi. Dengan kata lain proses analisis ini dikatakan juga sebagai proses pemecahan masalah.

Proses pemecahan masalah ini pada umumnya terbagi dalam tiga tahap yaitu:

- **Pertama-** mengidentifikasi masalah atau situasi yang perlu diperbaiki;
- **Kedua-** menentukan hasil yang akan dicapai; dan
- **ketiga-** mengidentifikasi dan memilih alternatif pemecahan masalah yang paling sesuai dengan masalah atau kondisi yang dihadapi.

Aktivitas pemecahan masalah yang bertujuan untuk mengembangkan sistem informasi akuntansi selain dibutuhkan oleh pengguna sistem (*end users*) juga didorong dengan adanya masalah (*problem*), kesempatan (*opportunity*) dan pengarahannya (*directive*).

Masalah (*problem*) adalah kondisi atau situasi yang tidak diharapkan terjadi yang menyebabkan terganggunya tujuan, sasaran, target dan penerapan kebijakan organisasi. Misalnya terganggunya target penjualan karena adanya keterlambatan dalam proses produksi.

Kesempatan (*opportunity*) adalah pengembangan sistem yang ada, walaupun sistem tersebut masih layak dipergunakan. Seperti gagasan manajemen untuk melakukan pengetatan jadwal produksi, walaupun pada saat tersebut target pencapaian produksi belum bermasalah.

Penerapan kesempatan ini lebih bersifat berjaga-jaga, sebelum masalah muncul. Pengarahannya (*directive*) adalah perubahan aktivitas atau prosedur berdasarkan permintaan pihak manajemen,

Pengembangan sistem informasi manajemen secara umum didorong oleh adanya:

1. Masalah
2. Kesempatan
3. Pengarahan

Pengarahan (*directive*) adalah perubahan aktivitas atau prosedur berdasarkan permintaan pihak manajemen, peraturan pemerintah atau pengaruh-pengaruh eksternal lainnya.

peraturan pemerintah atau pengaruh-pengaruh eksternal lainnya. Misalnya, adanya ketentuan pengurangan jam kerja yang diusulkan oleh pemerintah, yang berlaku bagi semua instansi baik pemerintah ataupun swasta. Contoh lain bila manajemen mengusulkan adanya penerapan kebijakan produksi baru untuk meningkatkan pelayanan kepada pelanggan. Pengarahan dapat juga bersifat teknis, seperti perubahan penerapan komputer dari penggunaan komputer perseorangan (*personal computer*) menjadi sistem jaringan (*network*). Pengarahan teknis ini biasanya dilakukan bila teknologi yang selama ini digunakan sudah tidak memenuhi lagi sasaran, target dan kebijakan perusahaan, seperti sistem berjalan dengan sangat lambat, sulit dikelola dan tidak memuaskan lagi.

Pada prakteknya banyak sekali, kemungkinan timbulnya masalah (*problem*), kesempatan (*opportunity*) dan pengarahan (*directive*). James Watherbe (1984) mengelompokan masalah, kesempatan dan pengarahan itu dalam kerangka kerja yang disebutnya PIECES yang terdiri dari enam kategori yaitu:

Kerangka kerja yang dijadikan dasar pemecahan masalah oleh analis sistem adalah:

1. Kinerja
2. Informasi/Data
3. Ekonomi
4. Pengendalian
5. Efisiensi
6. Pelayanan

- Kebutuhan untuk meningkatkan kinerja (*performance*).
- Kebutuhan untuk meningkatkan kualitas informasi atau *data (information)*.
- Kebutuhan untuk meningkatkan bidang ekonomi (*economy*) atau biaya.
- Kebutuhan untuk meningkatkan pengendalian (*control*) dan keamanan.
- Kebutuhan untuk meningkatkan efisiensi (*efficiency*) sumber daya manusia dan mesin.
- Kebutuhan untuk meningkatkan jasa/pelayanan (*service*) pada pelanggan, rekanan, pegawai dan pihak-pihak lainnya.

Keenam kerangka kerja tersebutlah yang dijadikan dasar oleh seorang analis sistem dalam memecahkan masalah yang dihadapi dengan melakukan analisis terhadap keenam komponen kerangka kerja tersebut di atas, berikut ini adalah uraiannya.

Analisis Kinerja

Kinerja suatu perusahaan bermasalah apabila dalam melaksanakan aktivitas bisnisnya, perusahaan dan komponen yang ada dalam perusahaan dianggap berjalan lambat dalam mencapai sasaran yang telah ditetapkan

Kinerja suatu perusahaan bermasalah apabila dalam melaksanakan aktivitas bisnisnya, perusahaan dan komponen yang ada dalam perusahaan dianggap berjalan lambat dalam mencapai sasaran yang telah ditetapkan. Sementara jika ada unsur manajemen yang mengusulkan untuk melakukan percepatan aktivitas bisnis agar sasaran organisasi/perusahaan segera tercapai, maka pada saat itu munculah apa yang dinamakan kesempatan dan untuk melaksanakan usulnya itu manajemen harus dapat memutuskan untuk mengganti sistem lama yang selama ini berjalan, maka proses tersebut dinamakan adanya pengarahan. Mungkin usulan tersebut berupa usulan perubahan dari sistem manual ke sistem komputer.

Kinerja sebuah organisasi/perusahaan diukur dari hasil kerja yang diperoleh selama periode tertentu (*throughput*) dan oleh rata-rata waktu penundaan yang terjadi antara transaksi dan pelaksanaan transaksi (*response time*). Misalnya, apabila sebuah perusahaan ingin meningkatkan penerimaan kreditnya seperti yang terjadi pada periode sebelumnya, maka perusahaan tersebut harus mempercepat proses transfer informasi setiap transaksi kredit kepada bagian kredit.

Analisis Data dan Informasi

Informasi merupakan hal penting bagi pemakai sistem atau manajemen. Sebuah sistem informasi akuntansi harus mampu menghasilkan informasi yang dapat digunakan untuk mengevaluasi apakah ada masalah dalam merealisasikan rencana dan apakah ada kesempatan untuk memperbaiki penyimpangan. Peningkatan kualitas informasi bukan berarti peningkatan jumlah informasi, karena informasi yang berlebihanpun dapat menjadi masalah bagi organisasi/perusahaan.

Keadaan dimana dibutuhkan peningkatan informasi apabila :

- Kurangnya informasi yang dapat dipergunakan untuk mengambil keputusan atau informasi mengenai situasi terbaru. Misalnya bagian manajemen tidak dapat merubah suatu prosedur penerimaan kas, apabila pada bagian tersebut tidak dapat membuktikan ada penyelewengan dalam penerimaan kas.
- Tidak tersedianya informasi yang relevan sehubungan dengan masalah yang dihadapi. Misalnya, bagian personalia tidak dapat segera memutuskan untuk menyetujui atau tidak usulan peningkatan biaya lembur, bagi departemen tertentu apabila *data* yang diajukan tidak mendukung permohonan tersebut.
- Ada Informasi tapi tidak memberikan manfaat bagi manajemen.
- Terlambatnya informasi yang dibutuhkan.
- Berlebihannya informasi yang tersedia.
- Tidak akuratnya informasi yang tersedia, misalnya terlalu banyak informasi yang salah sehingga tidak dapat dipergunakan untuk mengambil keputusan.

Informasi umumnya diperlukan untuk memberikan pengarah, seperti bentuk laporan keuangan yang ditetapkan oleh pemerintah. Analisis informasi dilakukan berdasarkan hasil yang diperoleh dari suatu aktivitas, sedangkan analisis *data* didasarkan pada *data* yang disimpan oleh sistem. Analisis *data* diperlukan apabila:

- Adanya kesimpangsiuran *data*, dimana *data* disimpan atau dikumpulkan di banyak bagian. Kondisi ini akan menyebabkan tidak terintegrasinya *data* yang disimpan dan membutuhkan banyak tempat.

Sistem informasi akuntansi harus mampu menghasilkan informasi yang dapat digunakan untuk mengevaluasi apakah ada masalah dalam merealisasikan rencana dan apakah ada kesempatan untuk memperbaiki penyimpangan.

- *Data* yang ada tidak fleksibel, hal ini terjadi apabila organisasi tetap berusaha mengumpulkan dan menyimpan *data*, walaupun *data* tersebut sulit diperoleh dan tidak mudah pengelolaannya.

Analisis Ekonomi

Keuntungan ekonomi merupakan pertimbangan paling umum yang mendorong dikembangkan proyek sistem informasi akuntansi

Keuntungan ekonomi merupakan pertimbangan paling umum yang mendorong dikembangkannya proyek sistem informasi manajemen. Misalnya, bagian pembelian diminta untuk melakukan pengurangan biaya bahan mentah. Ada dua cara yang dapat ditempuh, pertama membandingkan struktur berbagai harga yang ditawarkan oleh supplier yang berbeda, kedua dengan mengambil keuntungan dari potongan harga yang ditawarkan oleh supplier untuk pembelian jumlah yang besar. Bagaimanapun hal yang penting adalah mampu mengetahui tingkat keseimbangan antara pengurangan biaya pembelian yang dilakukan dengan peningkatan biaya penyimpanan persediaan bahan mentah (karena adanya biaya penyimpanan dan penanganan persediaan). Sehingga dapat diambil keputusan alternatif mana yang akan diambil

Analisis Pengendalian dan Keamanan

Pengendalian biasanya diterapkan untuk meningkatkan kinerja dari sistem, pencegah atau mendeteksi kecurangan atau kegagalan sistem dan menjamin keamanan dari *data*, informasi dan peralatan yang dimiliki oleh perusahaan.

Untuk menjaga agar tujuan dan sasaran perusahaan dapat dicapai sesuai dengan yang telah ditetapkan, maka semua aktivitas perusahaan perlu dipantau dan dikoreksi apabila terjadi ketidaksesuaian dengan ketentuan.

Pengendalian biasanya diterapkan untuk meningkatkan kinerja dari sistem, pencegah atau mendeteksi kecurangan atau kegagalan sistem dan menjamin keamanan dari *data*, informasi dan peralatan yang dimiliki oleh perusahaan. Ada dua kondisi yang mendorong diperlukannya analisis pengendalian dan keamanan, yaitu kontrol yang longgar dan kontrol yang ketat.

Pengendalian yang longgar pada sistem informasi akuntansi akan menyebabkan terjadinya perbedaan antara sistem informasi akuntansi dan sistem bisnis. Contoh yang umum terjadi pada pengendalian persediaan, adalah tidak cocoknya jumlah fisik dan jumlah yang tercatat dalam komputer. Hal ini terjadi karena setiap bagian persediaan mengambil atau menambah persediaan tidak pernah melakukan perhitungan ulang dan menyesuaikannya dengan catatan yang ada di kartu persediaan atau pun yang ada di komputer. Tetapi sistem yang terlalu ketat juga akan menyebabkan sistem berjalan terlalu lambat.

Analisis Efisiensi

Pengertian efisiensi adalah *output* dibandingkan dengan *input*, masalah-masalah dan kesempatan yang muncul dalam efisiensi adalah bagaimana meningkatkan *output* dengan meminimalkan *input*. Kita seringkali bingung dengan istilah efisien dan ekonomis,

pada hakekatnya ekonomis lebih menekankan pada berapa banyak sumber daya yang digunakan sedangkan efisiensi adalah bagaimana menggunakan sumber daya tersebut seminimal mungkin pemborosan.

Kita lihat contoh pada sebuah industri yang memiliki 145 bengkel kerja. Produk yang berbeda dihasilkan oleh masing-masing bengkel dalam satu periode produksi. Manajemen mempertimbangkan untuk melakukan ekspansi produk, tetapi tidak mempunyai cukup dana. Apa yang harus dilakukan menghadapi situasi seperti ini. Ada dua hal yang harus dianalisis pertama, bagaimana konsolidasi dari order-order untuk produk yang sama? Kedua, bagaimana pembagian jadwal masing-masing bengkel kerja, apakah ada hari-hari dimana banyak sekali pekerjaan, sementara dihari lainnya tidak ada yang dilakukan?. Hasil dari analisis terhadap kedua masalah tersebut akan memberikan jawaban kepada manajemen, bagaimana ia harus menjadwalkan dan mengontrol produksinya sehingga tercipta efisiensi dari masing-masing bengkel kerja.

Analisis Jasa/Pelayanan

Proyek pengembangan sistem pada umumnya juga didorong oleh keinginan manajemen untuk meningkatkan kualitas pelayanan kepada konsumen didalam aktivitas usahanya. Analisis terhadap tingkat pelayanan yang diberikan perusahaan, umumnya dijadikan dasar dalam mengevaluasi bagaimana pelaksanaan pelayanan yang dilakukan oleh perusahaan selama ini. Meningkatkan kualitas pelayanan juga berarti memberikan kepuasan yang optimal, baik kepada pelanggan, karyawan dan juga pihak manajemen.

Peningkatan pelayanan biasanya dilakukan untuk menciptakan kondisi-kondisi sebagai berikut:

- **Memperoleh akurasi data, proses dan hasilnya**- misalnya untuk mengurangi kesalahan pada penulisan nomor tagihan pada *invoice* pelanggan
- **Mencapai reabilitas**- dengan tujuan untuk menciptakan konsistensi dari proses dan hasilnya, misalnya penentuan pemberian kredit pada pelanggan yang mempunyai peringkat yang baik dan sejarah pembayaran yang baik pula.
- **Memudahkan penggunaan** - saat ini penggunaan sistem informasi akuntansi berbasis komputer semakin dikenal oleh pemakai sistem. Sistem informasi akuntansi baik manual maupun berbasis komputer, harus mudah digunakan. Beberapa proyek pengembangan sistem informasi akuntansi saat ini banyak ditunjukkan untuk meningkatkan kemudahan dalam pemakaian sistem informasi akuntansi berbasis komputer tersebut.
- **Menciptakan fleksibilitas** artinya sistem yang diterapkan dapat menerima pengecualian pada kasus-kasus tertentu, misalnya bila ada pembelian yang dibayarkan secara kredit, tetapi

Pengertian efisiensi adalah *output* dibandingkan dengan *input*; masalah-masalah dan kesempatan yang muncul dalam efisiensi adalah bagaimana meningkatkan *output* dengan meminimalkan *input*.

Analisis terhadap tingkat pelayanan yang diberikan perusahaan menjadi dasar dalam mengevaluasi bagaimana pelaksanaan pelayanan yang dilakukan oleh perusahaan selama ini.

dengan jangka waktu yang lebih pendek dari ketentuan umumnya. Pada kondisi ini diharapkan sistem informasi akuntansi yang diterapkan dapat menerimanya tanpa melalui proses yang berbelit-belit sehingga menimbulkan pembatalan order.

- **Menciptakan koordinasi-** dalam menjalankan aktivitasnya suatu organisasi umumnya didukung oleh beberapa fungsi. Setiap fungsi harus dapat berkoordinasi dengan fungsi lainnya untuk mencapai sasaran dan tujuan organisasi, atau dengan kata lain harus dapat menciptakan satu sinergi dimana seluruh bagian organisasi menerima manfaat yang sama dari aktivitas yang dilakukan. Misalnya perlunya ada kesesuaian antara jadwal produksi dengan jadwal penyediaan bahan mentah, sehingga tidak menimbulkan keterlambatan produksi di satu bagian atau menimbulkan penimbunan bahan mentah di bagian lain.

Enam kerangka kerja tersebut di atas masing-masing saling berhubungan satu sama lain dalam proses pemecahan masalah melalui pengembangan sistem informasi, tetapi walaupun demikian analisis mungkin saja hanya memerlukan satu kategori saja.

Dengan berpedoman pada keenam kerangka kerja tersebut, seorang analis diharapkan dapat memecahkan masalah-masalah yang dihadapi organisasi melalui pengidentifikasian, pelaksanaan analisis dan pemecahan masalah dengan menggunakan tahapan-tahapan dan tugas-tugas analis sistem. Berikut adalah uraian ringkasnya.

Seperti telah disebutkan di muka proses pemecahan masalah ini pada umumnya terbagi dalam tiga tahap yaitu: pertama, mengidentifikasi masalah atau situasi yang perlu diperbaiki; kedua, mendefinisikan hasil yang akan dicapai dan ketiga, mengidentifikasi dan memilih alternatif pemecahan masalah yang paling sesuai dengan masalah atau kondisi yang dihadapi.

Mengidentifikasi Masalah

Mengidentifikasi masalah dilakukan dengan cara mempelajari dan mengetahui sistem informasi akuntansi yang tengah berjalan pada saat itu dan permasalahannya

Seperti pembangunan sebuah kawasan real estate, seorang arsitek tidak akan merancang bentuk bangunan di kawasan tersebut sebelum mengetahui dengan pasti bagaimana karakteristik dari tanah yang akan dipergunakan, bagaimana legalitasnya dan bagaimana pemerintah mengatur tata kota untuk kawasan itu. Begitu juga seorang pelaku bisnis yang baik tidak akan melakukan perubahan kebijaksanaan sebelum ia tahu persis apa yang dibutuhkan dan apa masalah yang dihadapi oleh organisasinya. Demikian juga halnya seorang analis sistem, sebelum ia menyelesaikan masalah yang dihadapi organisasi melalui pengembangan sistem informasi akuntansi, maka ia terlebih dahulu harus mempelajari dan mengetahui sistem informasi akuntansi yang tengah berjalan pada saat itu dan permasalahannya, kemudian memilih alternatif pemecahan masalah yang paling sesuai dengan kebutuhan dan kondisi yang ada baru ia dapat merancang sistem baru berdasarkan keinginan pemakai sistem dan selanjutnya.

Survei dan mempelajari sistem yang tengah berjalan memiliki tujuan yang sama yaitu untuk memahami sistem yang tengah berjalan dan masalahnya sehingga dapat diperoleh informasi yang dapat digunakan untuk menentukan langkah pemecahan apa yang perlu dilakukan. Perbedaan dari kedua aktivitas tersebut adalah pada tingkat pemahaman dan waktu yang diperlukan untuk melakukannya.

Survei lebih bertujuan untuk melakukan penyelidikan awal terhadap sistem yang sedang berjalan, sedangkan tahap mempelajari sistem adalah untuk penyelidikan yang lebih terinci. Langkah-langkah yang perlu dilaksanakan dalam kedua tahap ini ada-lah :

- **Mengidentifikasi semua pemakai sistem (pemakai)**- pada sistem yang sedang berjalan, hal ini penting dilakukan karena masing-masing pemakai memiliki kepentingan yang berbeda terhadap sistem yang diterapkan.
- **Menganalisis aspek bisnis dalam sistem yang sedang berjalan**- hal ini dilakukan untuk mengetahui sampai sejauhmana sistem yang sedang berjalan memberikan manfaat pada aktivitas bisnis organisasi dalam mencapai tujuan, sasaran dan menerapkan kebijakannya.
- Mengidentifikasi dan menganalisis fungsi-fungsi sistem informasi akuntansi yang sedang berjalan.
- Mengidentifikasi dan menganalisis komponen-komponen dari sistem informasi akuntansi yang sedang berjalan.

Setelah tahapan tersebut selesai dilakukan maka, analisis sistem harus mulai menentukan tugas-tugas yang perlu dilaksanakan diantaranya,

- Melakukan survei kelayakan proyek pengembangan sistem;
- Mempelajari bagaimana sistem yang ada dilaksanakan;
- Menganalisa masalah, keterbatasan, dan kendala pada sistem yang sedang berjalan,
- Memperkenalkan alternatif pemecahan masalah,
- Memperbaharui lingkup dan kelayakan pengembangan sistem dan Mempresentasikan temuan-temuan dan menyusun laporan dari tahapan ini dalam sebuah laporan formal.

Bagaimana Menentukan Hasil yang akan Dicapai

Untuk dapat menentukan hasil yang akan dicapai terlebih dahulu analisis sistem harus mengetahui apa yang dibutuhkan dan diinginkan oleh pemakai sistem (pemakai) dari pengembangan sistem yang akan dilakukan. Seringkali analisis sistem melupakan tahapan ini, sehingga sistem yang disusun menjadi sia-sia karena tidak dapat dipergunakan oleh pemakai. Oleh karena itu menentukan

Survei lebih bertujuan untuk melakukan penyelidikan awal terhadap sistem yang sedang berjalan.

Hasil yang akan dicapai harus diketahui oleh analisis sistem dari pemakai agar apa yang dibutuhkan dan diinginkan sesuai.

keinginan pemakai sistem adalah tahapan penting, bila ingin meraih sukses dalam menyusun sebuah sistem informasi akuntansi.

Tujuan dari tahap penentuan permintaan pemakai adalah untuk mengetahui apa yang harus dikembangkan dari sistem yang ada tanpa harus mengetahui secara khusus bagaimana sistem tersebut dikembangkannya. Langkah-langkah yang perlu dilakukan dalam tahapan ini adalah :

- Libatkan semua pemakai dalam menentukan sistem informasi akuntansi yang diinginkan.
- Kaji ulang dan perbaiki kebutuhan aktivitas bisnis untuk sistem informasi akuntansi yang baru.
- Tentukan fungsi-fungsi sistem informasi akuntansi yang harus tersedia pada sistem informasi akuntansi yang baru.
- Tentukan komponen sistem informasi akuntansi yang baru, diluar komponen-komponen yang dikomputerisasikan.

Untuk melengkapi tahapan ini, maka harus dilakukan tugas-tugas sebagai berikut:

- Menentukan tujuan dan prioritas sistem informasi akuntansi.
- Membuat garis besar keinginan/kebutuhan pemakai dari sistem informasi akuntansi yang baru, dan
- Tentukan secara terinci permintaan pemakai atas sistem informasi akuntansi baru yang akan disusun.

Seluruh tahap-tahap dan tugas-tugas yang telah dilaksanakan, hasilnya dilaporkan dalam bentuk laporan permintaan pemakai atau *requirements statement*, untuk lebih memahami apa yang dibutuhkan oleh pemakai.

Bagaimana Memilih Alternatif Pemecahan Masalah

Setelah ketiga tahapan yaitu survei, study dan tahap penentuan dilaksanakan, selanjutnya analisis sistem harus menentukan langkah pemecahan apa yang akan dilakukan. Jangan terjebak dengan ide pemecahan yang pertama muncul dalam kepala anda, karena anda akan terperosok pada hasil yang kurang memuaskan. Seorang analisis sistem yang baik tentu akan melakukan identifikasi solusi, analisis solusi dan menentukan solusi yang paling memadai berdasarkan hasil analisis tersebut.

Tujuan yang ingin dicapai dalam pemilihan alternatif solusi ini adalah: dapat menentukan solusi atau pemecahan apa yang paling memadai dan bermanfaat bagi pemakai. Sedangkan sasaran yang ingin dicapai adalah :

- Dapat menentukan alternatif-alternatif pemecahan masalah berdasarkan penggunaan sistem informasi akuntansi berbasis komputer. Tahap ini melengkapi tahap pengidentifikasian yang tidak dilakukan pada saat menentukan permintaan pemakai.

- Mengevaluasi dampaknya pada pemakai.
- Mengevaluasi dampaknya pada aktivitas usaha/bisnis.
- Mengevaluasi sejauhmana alternatif pemecahan masalah ini dapat memenuhi fungsi sistem informasi akuntansi yang di-minta.

Untuk mendapatkan alternatif pemecahan yang paling baik, maka perlu dilakukan beberapa aktivitas dengan lebih spesifik. Aktivitas-aktivitas tersebut adalah :

- Menentukan cara yang dapat dilakukan dalam menerapkan sistem informasi akuntansi, dan untuk itu perlu ditentukan spesifikasi alternatif pemecahan apa yang akan dilakukan. Penekanan spesifikasi umumnya diterapkan pada pemanfaatan orang dan mesin yang secara spesifik menentukan apa yang akan dikerjakan oleh manusia dan mana yang akan dilakukan oleh mesin.
- Melakukan analisis kelayakan dari alternatif pemecahan yang dipilih. Kriteria-kriteria yang harus diperhatikan tanpa dibatasi oleh biaya dan manfaat adalah: bagaimana kelayakan operasionalnya. bagaimana kelayakan teknisnya dan bagaimana kelayakan secara ekonomi.
- Persiapkan jadwal perancangan dan teknik pemecahan masalah yang direkomendasikan.
- Langkah terakhir adalah memperkenalkan sistem yang telah dihasilkan. Agar mencapai hasil yang optimal maka jangan dilupakan keenam kerangka kerja sistem, sehingga pada saat memperkenalkan sistem baru kita dapat mengatakan bahwa sistem ini dapat meningkatkan kinerja, informasi dan *data*, mengurangi biaya, meningkatkan pengendalian dan keamanan, menciptakan efisiensi dan meningkatkan pelayanan.

Rekomendasi dari pemecahan masalah yang dipilih disusun dalam sebuah proposal pengembangan sistem atau laporan kelayakan proyek. Uraian di atas menggambarkan bagaimana seorang analis sistem bekerja memecahkan masalah yang dihadapi oleh organisasi. Lalu bagaimana kedudukan seorang analis sistem dalam organisasi?. Bagi perusahaan yang memiliki divisi pengembangan sistem secara khusus, umumnya akan mempekerjakan analis-analis profesional secara khusus, uraian berikut ini akan memberikan gambaran dimana seorang analis berada pada struktur organisasi suatu perusahaan.

2.5 Kedudukan Analis Sistem Dalam Organisasi

Kita mungkin akan bertanya-tanya dengan tanggungjawab dan tugas yang begitu penting serta harus ditunjang oleh pendidikan dan pengalaman yang memadai, dimanakah seorang analis sistem ditempatkan? Apakah sebagai tenaga ahli khusus atau merupakan

bagian dari organisasi itu sendiri? Pertanyaan ini timbul karena seringkali dalam industri pengertian analisis sistem ini disamakan dengan seorang programmer komputer, tetapi pada kenyataannya seorang analisis sistem bekerja lebih dari seorang programmer komputer. Analisis sistem harus mampu melakukan pemilihan perangkat komputer, menentukan orang yang akan menggunakan sistem informasi akuntansi, menyusun prosedur dari sistem informasi akuntansi tersebut dan menyusun *file-file* atau *database* dari sistem informasi akuntansi yang dikembangkan.

Seorang analisis sistem juga tidak memiliki pekerjaan dengan jawaban yang pasti antara salah dan benar karena terdapat banyak pilihan pemecahan masalah. Solusi sistem merupakan solusi yang dapat dirundingkan, oleh karenanya seorang analisis harus memiliki kemampuan untuk berkomunikasi dengan pemakai sistem (*users*), manajemen, programmer, manajer sistem informasi akuntansi, auditor dan penjual *software* sistem informasi akuntansi.

Analisis sistem biasanya bekerja pada divisi Sistem Informasi Komputer (*Computer Information System/CIS*), pada beberapa perusahaan divisi SIK biasa disebut juga sebagai divisi sistem informasi, divisi sistem informasi akuntansi atau divisi Pelayanan *Data* dan Informasi tergantung kepada situasi dan kondisi perusahaan. Kedudukan manajer sistem informasi atau sistem informasi manajemen/akuntansi berbasis komputer ini pada perusahaan tertentu berada pada jajaran eksekutif, seperti wakil presiden dan direktur. Tapi ini semua kembali tergantung kepada situasi dan kondisi perusahaan.

Dalam organisasi analisis sistem bekerja pada divisi Sistem Informasi Komputer, yang berfungsi sebagai pusat pengembangan, pusat informasi, pusat *database* dan pusat komputer.

Walaupun memiliki istilah yang berbeda-beda tetapi pada umumnya bagian atau divisi sistem informasi mempunyai empat kegiatan utama yaitu sebagai: pusat pengembangan, pusat informasi, pusat *database* dan pusat komputer. Pusat Pengembangan merupakan bidang baru biasanya terdiri dari analisis sistem dan programmer yang mengembangkan dan mendukung sistem bagi pemakai sistem dan manajemen.

Pusat Informasi juga merupakan divisi yang relatif baru, biasanya terdiri dari analisis, programmer yang membantu perusahaan memelihara dan mengembangkan sistem yang diterapkan oleh perusahaan. Biasanya pusat informasi mengembangkan sistem informasi dengan strategi yang lebih sempit dibandingkan dengan pusat pengembangan. Pusat informasi lebih menekankan pada pendayagunaan komputer baik oleh perusahaan maupun manajemen. Pusat *database* juga disebut sebagai Administrasi *data*, berfungsi melacak dan mengelola sejumlah *data* yang disimpan oleh organisasi/perusahaan. Pusat Komputer merupakan pusat pelayanan dan penyediaan jasa komputerisasi seperti pemasukan *data*, pengoperasian sistem informasi akuntansi, penyusunan program sistem informasi akuntansi dan mengkomunikasikan *data* (antara komputer pusat, pengendali dan komputer-komputer perorangan).

Umumnya analisis dan programmer bekerja di satu divisi pengembangan. Analisis dan programmer ini kadangkala bekerja dalam satu

tim yang biasa disebut proyek, tim ini dibentuk dan dibubarkan seiring dengan adanya proyek penyusunan sistem informasi, baik sistem informasi akuntansi atau yang lainnya. Umumnya tim proyek ini terdiri dari pemimpin proyek yang dipegang oleh seorang analis yang berpengalaman, analis sistem dan programmer.

Peranan analis sistem dalam proyek adalah sebagai fasilitator. Analis bekerja sebagai perantara diantara berbagai kalangan dan fasilitas komputer yang dibutuhkan oleh kalangan tersebut. Analis sistem spesialis sistem informasi akuntansi adalah orang yang paling mengetahui tentang sistem informasi akuntansi, dengan tugas sebagai analis sistem, berbagai beban tanggung jawab mungkin saja dipikulnya.

Seorang analis tidak hanya dapat bekerja di pusat informasi berbasis komputer dalam suatu perusahaan, ia juga dapat bekerja sebagai pembuat *software* pada perusahaan komputer atau bekerja sebagai seorang konsultan. Peluang kerja sebagai analis sistem ini kini benar-benar terbuka lebar, tinggal bagaimana ia mampu mengembangkan keahliannya sehingga ia dapat memperoleh kedudukan yang memuaskan dan sesuai dengan wawasan dan kemampuannya. Pada uraian selanjutnya akan dijelaskan bagaimana mempersiapkan diri menjadi seorang analis sistem, lebih spesifik lagi analis sistem informasi akuntansi.

2.6 Persiapan Karier Sebagai Analis Sistem

Karir sebagai seorang analis sistem bukan suatu hal yang mudah diraih, untuk bekerja sebagai seorang analis perlu persiapan yang matang dan pengalaman yang luas, khususnya mengenai sistem informasi akuntansi berbasis komputer. Seperti kita ketahui bahwa seorang analisis sistem berkaitan erat dengan bagian akuntansi, pelaku bisnis dan programmer komputer. Lalu hal apa saja yang harus dipersiapkan untuk menjadi seorang analis sistem yang sukses?.

Banyak perusahaan yang menganggap seorang programmer komputer yang berpengalaman akan dapat menjadi seorang analis sistem yang baik pula atau sebaliknya seorang programmer yang tidak berpengalaman tidak akan menjadi seorang analis sistem yang baik. Pada kenyataannya kedua kondisi tersebut sama sekali tidak berpengaruh terhadap berhasil tidaknya untuk menjadi seorang analis sistem. Walaupun cukup sulit untuk menjelaskan kemampuan apa dan gelar apa yang harus dimiliki oleh seorang analis sistem untuk mencapai sukses. Secara umum seorang analis sistem harus memiliki pengetahuan teknik dibidang teknologi sistem informasi akuntansi, pengalaman dan pengetahuan dalam penyusunan program dengan komputer, memiliki pengetahuan bisnis secara umum, memiliki kemampuan memecahkan masalah, mampu berkomunikasi dengan baik, memiliki hubungan/relasi yang luas, memiliki kemampuan menganalisis dan merancang sistem secara formal dan memiliki pengalaman dalam menyusun sebuah sistem informasi akuntansi.

Memiliki Pengetahuan Teknik dan Teknologi Sistem Informasi Akuntansi

Analisis sistem harus memiliki pengetahuan teknis dan teknologi sistem informasi manajemen saat ini dan trend teknologi dimasa yang akan datang

Analisis sistem juga sering dikatakan sebagai “pembawa perubahan”. Analisis sistem harus memiliki pengetahuan teknis dan teknologi sistem informasi akuntansi saat ini dan trend teknologi dimasa yang akan *datang*, karena dengan demikian dapat menerangkan kepada pemakai sistem bagaimana sebuah teknologi baru dapat memberikan manfaat bagi aktivitas usaha pemakaian sistem maupun kegiatan pemakai sistem. Teknologi yang harus dikuasai oleh seorang analis sistem diantaranya teknologi *data-base*, komunikasi *data*, pengenalan komputer, pengetahuan mengenai *software* komputer, pengetahuan tentang komputer grafik, jaringan komputer, generasi-generasi komputer dan pengamanan komputer.

Pengetahuan-pengetahuan tersebut dapat diperoleh baik melalui pendidikan formal maupun non formal, seperti kursus-kursus atau pendidikan jenjang diploma. Selain pendidikan formal, seorang analis juga harus memperluas wawasannya dengan banyak membaca berbagai literatur yang berhubungan dengan sistem informasi akuntansi dan komputer.

Ada baiknya juga bila anda ingin menjadi seorang analis sistem yang profesional, bergabunglah dengan organisasi-organisasi yang bergerak dalam pengembangan sistem informasi akuntansi dan komputer.

Memiliki Pengetahuan dan Pengalaman dalam Pemrograman Komputer

Analisis sistem harus dapat menggunakan bahasa komputer tingkat tinggi untuk diaplikasikan dalam aktivitas bisnis

Seorang analis harus memahami bahasa pemrograman komputer tingkat tinggi untuk diaplikasikan dalam aktivitas bisnis, seperti *COBOL*, atau *FORTRAN* untuk bidang teknik dan lain-lain. Ada beberapa bahasa pemrograman generasi ke empat yang perlu dikenal oleh seorang analis yaitu, *FOCUS*, *IDEAL*, *NATURAL*, *ADS/O*, *RAMIS*, *SAS*, *MANTIS*, *RBASE*, *dBASE*, *Cliper*, *Foxpro* dan *ORACLE*. Beberapa perusahaan *software* komputer kini banyak yang mengembangkan bahasa C, terutama untuk paket-paket *software* untuk PC. Bahasa-bahasa pemrograman ini digunakan untuk mendukung pembuatan prototipe sistem. Bahasa pemrograman ini juga akan membantu analis dalam merancang sistem yang akan disusunnya.

Memiliki Pengetahuan Bisnis Secara Umum

Analisis sistem harus memiliki pengetahuan aplikasi dan fungsi-fungsi dalam aktivitas bisnis

Karena sistem informasi akuntansi lebih banyak diterapkan dalam aktivitas bisnis, maka penting sekali bagi seorang analis memiliki pengetahuan aplikasi dan fungsi-fungsi dalam aktivitas bisnis, seperti pengetahuan tentang manajemen atau produksi. Bila

penekanan sistem ini untuk mengembangkan sistem informasi akuntansi pada sebuah perusahaan, maka hal ini merupakan suatu tantangan bagi para akuntan untuk menambah wawasannya dengan pengetahuan aplikasi komputer untuk bisnis.

Beberapa aplikasi bisnis umum yang perlu diketahui diantaranya adalah: manajemen keuangan, manajemen biaya atau manajemen manajemen, metode-metode kuantitatif seperti statistik, pemasaran, operasi produksi dan persediaan, manajemen personalia, keuangan, tingkah laku organisasi dan hukum-hukum dan etika bisnis.

Memiliki Kemampuan Memecahkan Masalah

Kemampuan memecahkan masalah adalah hal yang paling penting dimiliki oleh seorang analis, karena dalam aktivitas kerjanya analis dituntut untuk mampu memecahkan berbagai masalah yang dihadapi organisasi/perusahaan. Analis harus mampu memilah-milah masalah yang terjadi pada kelompok-kelompok tersendiri sesuai dengan kategori masalahnya, menganalisis berbagai aspek yang menyebabkan masalah itu muncul, dan kemudian menggunakan sistem untuk memecahkan masalah yang dihadapi. Analis sistem juga harus mampu melakukan analisis terhadap sebab dan akibat yang mungkin terjadi, dan bukan hanya sekedar memecahkan masalahnya saja. Metodologi seperti analisis struktur, merupakan salah satu yang harus dilakukan oleh analis sistem dalam proses pemecahan masalah. Mampu berorganisasi dengan baik juga merupakan salah satu langkah yang baik untuk meningkatkan kemampuan memecahkan masalah.

Kreativitas adalah segi lain yang harus dimiliki seorang analis sistem, terutama kreativitas dalam menentukan alternatif pemecahan masalah dan pemenuhan kebutuhan pemakai sistem. Kreativitas ini harus dilatih sedemikian rupa sehingga dapat terus terjaga dan ditingkatkan setiap saat.

Mampu Berkomunikasi Dengan Baik

Berkomunikasi dengan efektif baik lisan maupun tulisan penting dimiliki oleh seorang analis. Tanpa mampu berkomunikasi dengan baik, sulit bagi seorang analis mengkomunikasikan gagasan-gagasan dalam mengembangkan sistem informasi akuntansi pada pemakai bagian akuntansi atau pada manajemen. Mampu berbahasa asing, terutama bahasa Inggris juga merupakan hal penting lainnya, karena seperti disebutkan terdahulu, bahwa seorang analis harus mampu berhubungan dengan berbagai kalangan.

Kemampuan berkomunikasi secara efektif baik lisan maupun tulisan, harus meliputi kemampuan menyusun laporan, menguasai teknis menulis, melakukan wawancara, melakukan presentasi dan kemampuan mendengarkan dengan cermat.

Analisis sistem harus memiliki kemampuan memecahkan masalah dengan memilah-milah masalah yang terjadi pada kelompok-kelompok tersendiri sesuai dengan kategori masalahnya,

Analisis sistem harus dapat berkomunikasi dengan efektif baik lisan maupun tulisan.

Mempunyai Relasi Yang Luas

Analisis sistem harus mempunyai relasi yang luas

Orientasi kerja sebuah pengembangan sistem informasi akuntansi melihat orang sebagai pengguna bukan alat yang digunakan, oleh karenanya seorang analis sistem harus bersifat terbuka atau mampu berorientasi pada orang-orang disekitarnya. Kemampuan menjalin hubungan atau membina hubungan akan menolong analis bekerja efisien. Strategisnya posisi seorang analis juga menuntutnya untuk mampu membina hubungan yang luas dengan berbagai kalangan.

Tanggungjawab analis yang utama adalah kepada manajemen dan pekerjanya/karyawan. Konflik antar personal seringkali terjadi dalam sebuah organisasi/perusahaan, dalam hal ini analis harus mampu menjadi penengah antara berbagai kepentingan dan menciptakan manfaat bagi bisnis secara keseluruhan.

Sebagai pembawa perubahan, analis tentu harus mampu mengkomunikasikan perubahan yang dibawanya pada setiap orang, untuk itulah kemampuan membina hubungan menjadi hal penting yang harus dimiliki oleh seorang analis. Pelajarilah teknik penjualan, maka anda sebagai seorang analis akan mampu menawarkan perubahan pada setiap anggota organisasi atau pada setiap unsur perusahaan.

Analisis sistem tidak bekerja sendiri, tetapi bekerja dalam satu tim. Sebagai bagian dari tim, maka tentu ia harus mampu menjalin hubungan yang baik dengan anggota tim yang lainnya. Bekerjasama dan berunding adalah kunci sukses dari pelaksanaan suatu proyek, khususnya dalam proyek pengembangan sistem informasi akuntansi.

Memiliki Kemampuan Formal dalam Menganalisis dan Merancang Sistem Informasi Akuntansi

Analisis sistem harus mengikuti pelatihan menganalisis dan merancang sistem

Secara formal, seorang analis sistem harus mengikuti pelatihan menganalisis dan merancang sistem. Keterampilan ini biasanya meliputi tiga aspek yaitu, prinsip-prinsip dan konsep dasar sistem, alat-alat dan teknik serta metodologi analisis dan perancangan sistem.

Sukses akan diraih oleh seorang analis jika selalu bekerja sesuai dengan prinsip dan konsep dasar kerja sebuah sistem. Prinsip dan konsep dasar yang terdiri dari enam dasar prinsip dimana analis sistem harus mementingkan kebutuhan pemakai (*user*) karena sistem dibuat untuk pemakai; analis juga harus menetapkan fase-fase dan tugas-tugas agar pengembangan sistem dapat dikelola dengan baik; kreativitas diperlukan oleh seorang analis karena tugas pengembangan sistem tidaklah terbatas, tetapi dapat meluas keberbagai bidang, oleh karenanya mungkin saja diperlukan perbaikan-perbaikan; sistem merupakan investasi modal, oleh karenanya analis harus mampu melakukan pertimbangan ekonomi; analis harus mampu menetapkan poin-poin pengecekan untuk mengevaluasi kelayakan dan jangan takut untuk

membuang hal-hal yang tidak layak dan yang perlu diingat segera dokumentasikan apa yang dihasilkan dalam pengembangan sistem. Keenam prinsip tersebut ini akan mudah diadaptasikan pada berbagai situasi yang berbeda-beda, karena metodologinya telah tersedia dan banyak pilihannya.

Banyaknya aktivitas yang harus dilakukan seorang analis dalam mengembangkan sistem, tentunya perlu didukung dengan pengetahuan, pendidikan formal dan wawasan yang luas, karena jika tidak terpenuhi maka proses pemecahan masalah melalui pengembangan sistem tidak dapat tercapai.

Berpengalaman

Berpengalaman sebagai analis artinya pernah terlibat secara langsung dalam proyek pengembangan sistem informasi. Pengalaman ini berguna untuk meningkatkan keterampilan dan kualitas dari analis sistem, agar menjadi seorang analis yang sukses maka diperlukan kesabaran. Seperti pepatah mengatakan “pengalaman adalah guru yang baik”. Maka teruslah mencari pengalaman, karena tidak ada buku pedoman yang akan memberikan pengalaman. Keterampilan dan kemampuan yang disyaratkan terdahulu dapat dipergunakan sebagai alat untuk menimba pengalaman, bila diarahkan dan diasah terus menerus.

Perkembangan teknologi komputer, memungkinkan seorang akuntan meningkatkan produktivitasnya dengan lebih luas lagi, hal ini menjadi tantangan dan kesempatan yang berharga bagi pengembangan karir akuntan. Pengembangan sistem informasi akuntansi berbasis komputer merupakan salah satu pendukung dalam peningkatan produktivitas pegawai dengan mengandalkan teknologi yang tersedia melalui otomatisasi aktivitas perusahaan. Paket-paket aplikasi komputer untuk sistem informasi akuntansi telah banyak beredar, walaupun dalam kenyataannya paket-paket tersebut sulit diterapkan karena paket tersebut dibuat bukan berdasarkan kebutuhan user dimana paket tersebut diterapkan. Penyusunan aplikasi komputer untuk sistem informasi manajemen ini biasanya memiliki fungsi untuk memproses seluruh transaksi manajemen secara umum. Akuntan bila bekerja sebagai analis sistem dalam menganalisis dan merancang sistem informasi akuntansi berbasis komputer berperan dalam memilih *software* apa yang harus digunakan dan dapat menjadi bagian dalam proyek pengembangan sistem informasi akuntansi sebagai anggota atau pimpinan tim proyek tersebut, atau sebagai konsultan sistem informasi akuntansi.

Dalam pengembangan sistem informasi akuntansi berbasis komputer langkah-langkah dalam analisis dan perancangannya akan sama dengan langkah-langkah pengembangan sistem informasi akuntansi umumnya,

Analisis sistem harus berpengalaman artinya harus pernah terlibat langsung dalam pengembangan sistem informasi akuntansi.

Rangkuman

Analisis sistem informasi akuntansi adalah seorang profesional yang bertanggung jawab untuk mempelajari masalah-masalah dan kebutuhan-kebutuhan dalam aktivitas bisnis dengan mendayagunakan manusia, metodologi dan teknologi komputer agar dapat memberikan manfaat yang optimal bagi peningkatan aktivitas bisnis. Teknologi komputer digunakan analisis untuk mengumpulkan, memproses dan menyimpan *data* agar dapat memberikan informasi akuntansi yang akurat dan tepat waktu bagi manajemen dan pihak luar perusahaan .

Dalam melaksanakan tugasnya analisis melakukan serangkaian kegiatan seperti analisis sistem informasi akuntansi, merancang sistem informasi akuntansi, dan menerapkan sistem informasi akuntansi tersebut disamping melakukan pengawasan agar sistem informasi akuntansi itu tetap dapat dimanfaatkan sesuai dengan kebutuhan pemakai. Peranan analisis sistem dalam aktivitas bisnis/organisasi adalah sebagai pemecah masalah karena aktivitas yang dilakukan dalam mengembangkan sistem merupakan aktivitas pemecahan masalah yang dihadapi oleh perusahaan.

Aktivitas pemecahan masalah dapat dilakukan berdasarkan pada enam kerangka kerja yang disebut PIECES yaitu, *performance* (kinerja), *information and data* (informasi dan *data*, *economy and cost* (ekonomi dan biaya), *control and security* (pengendalian dan keamanan), *efficiency* (efisiensi) dan *service* (jasa/ pelayanan). Langkah-langkah aktivitas pemecahan masalah yang dilakukan adalah mengidentifikasi, melakukan analisis, dan memilih alternatif pemecahan masalah melalui tahapan-tahapan tersebut.

Soal

1. Siapakah dan apa tugas analisis sistem?
2. Apa beda tanggung jawab analisis sistem dengan programmer?
3. Sebutkan tiga tahap pemecahan masalah?
4. Sebutkan tiga kondisi yang mendorong dikembangkannya sistem informasi akuntansi?
5. Sebutkan kerangka kerja yang menjadi dasar pemecahan masalah analisis sistem?

Tugas

1. Sebutkan pengetahuan dan kemampuan yang harus dimiliki oleh seorang analisis sistem?
2. Apakah seorang analisis sistem yang tidak memiliki pengetahuan dibidang akuntansi dapat menganalisis masalah dalam sistem informasi akuntansi? Berikan penjelasan.
3. Apakah seorang programmer yang tidak memiliki pendidikan formal dapat menjadi analisis sistem ?
4. Apakah seorang programmer karena pengalamannya sudah puluhan tahun dapat menjadi analisis sistem?
5. Bagaimana analisis sistem memecahkan masalah gap komunikasi antara dirinya sebagai pengembang sistem dan manajemen perusahaan terutama bagian akuntansi?

3

Siklus, Metode dan Teknik dalam Membangun Sistem Informasi Akuntansi

Pendahuluan

Seperti tiga ekor burung unta yang memiliki cara sendiri-sendiri dalam mempertahankan diri dari serangan musuh, demikian pula dengan pengembangan sistem informasi akuntansi. Setiap perusahaan atau pengembang sistem informasi akuntansi memiliki teknik dan metode sendiri-sendiri yang dirasanya paling cocok untuk menghadapi setiap masalah yang dihadapinya.

Pada awal berkembangnya sistem informasi akuntansi, pengembangan sistem informasi akuntansi dilakukan oleh programmer. Manajemen perusahaan (*user*) meminta kepada programmer untuk membuat program tertentu yang bisa membantu aktivitasnya. Dengan permintaan tersebut programmer akan meminta *data* yang harus dimasukkan dan laporan keuangan atau informasi akuntansi yang ingin dikeluarkan, berdasarkan *data* dan laporan inilah programmer mulai dan bekerja. Hasil akhir dari pekerjaan ini ternyata informasi akuntansi yang dihasilkan tidak memuaskan dan saat itulah muncul pemikiran perlu adanya analisis sistem informasi akuntansi sebelum sistem informasi akuntansi dirancang.

Dengan adanya kebutuhan untuk menganalisis dulu permasalahan yang ada dan kebutuhan informasi akuntansi para manager maka lahirlah satu siklus pengembangan sistem informasi yang dikenal sebagai siklus *System Development Life Cycle (SDLC)* yang merupakan tahap-tahapan pengembangan sistem informasi yang didalamnya terdiri dari beberapa tahapan yang terstruktur.

System Development Life Cycle memiliki aktifitas yang lebih detail atau rinci yang dikenal sebagai metode SDLC. Metode SDLC juga merupakan metode pertama yang digunakan dalam membangun sistem informasi akuntansi untuk selanjutnya metode SDLC tradisional ini berkembang menjadi metode SDLC modern dan juga lahir turunan lainnya seperti metode *Water Fall* dan metode *FAST*.

Disamping lahir metode metode dari turunan metode SDLC juga lahir ratusan metode baru, diantaranya metode *Prototyping*, *Rapid Application Development (RAD)*, metode *Dynamic System Development Methode (DSDM)* dan metode *Soft Systems*. Semua metode hanya diatas kertas dan konseptual sedangkan untuk implementasinya kita kenal sebagai teknik,

Bab ini akan membahas tentang siklus, metode dan teknik dalam membangun sistem informasi akuntansi.

3.1 Siklus, Metodologi dan Teknik

Dengan berkembangnya teknologi yang sangat pesat dewasa ini dimana hampir semua sektor kehidupan memanfaatkan dan tergantung kepada kemajuan teknologi khususnya teknologi komputer, para pengembang sistem informasi dituntut untuk menyajikan *software* aplikasi sistem informasi yang lebih kompleks dan berkualitas tinggi untuk mendukung perkembangan dunia usaha yang terus berkembang saat ini.

Tetapi sayangnya di Indonesia tuntutan ini belum sepenuhnya didukung dengan tersedianya sumber daya manusia yang memadai sehingga lamban dalam mengantisipasi terhadap perkembangan teknologi baru serta tidak dimilikinya metode dan prosedur yang dapat memenuhi tuntutan kebutuhan yang semakin hari semakin kompleks. Seringkali antara metode, prosedur dan teknologi tidak dapat diintegrasikan secara optimal. Kondisi-kondisi seperti ini menghasilkan sistem informasi yang kurang mendukung peningkatan produktivitas, sehingga memaksa manajemen dihadapkan kepada dua alternatif keputusan antara memiliki sistem informasi yang berkualitas atau melakukan efisiensi pengembangan.

Dalam pengembangan sistem informasi kita mengenal adanya siklus pengembangan sistem informasi (*life cycle*). Pada perkembangan selanjutnya banyak profesional sistem informasi yang mengatakan bahwa siklus pengembangan sistem informasi ini sudah tidak dapat dipergunakan lagi dan diganti kedudukannya dengan diperkenalkannya teknik-teknik dan metode pengembangan sistem informasi yang baru, sedangkan sebagian lagi mengatakan bahwa siklus sistem informasi masih tetap ada dan keberadaannya dilengkapi dengan adanya teknik dan metode lainnya. Uraian selanjutnya pada bab ini akan menjelaskan bagaimana pengertian dari terminologi-terminologi yang digunakan diatas.

Siklus (*life cycle*) adalah tahapan-tahapan dan tugas-tugas yang harus dilakukan dalam mengembangkan sistem informasi.

3.2 Siklus

Siklus (*life cycle*) dalam hal ini siklus pengembangan sistem informasi adalah tahapan-tahapan dan tugas-tugas yang harus dilakukan dalam mengembangkan sistem informasi, tanpa memperhatikan sistem informasi jenis apa yang akan dibuat dan seberapa luas yang harus dihasilkannya. Contoh dari siklus adalah siklus pengembangan sistem atau *systems development life cycle (SDLC)*.

Tabel 15.1 Tahapan dalam SDLC tradisional dan modern

SDLC Tradisional	SDLC Modern
1. Analisis (<i>Analysys</i>)	1. Perencanaan (<i>Planning</i>)
2. Perancangan (<i>Design</i>)	2. Analisis (<i>Analysis</i>)
3. Penerapan (<i>Implementation</i>)	2. Perancangan (<i>Design</i>)
4. Pemeliharaan (<i>Maintenance</i>)	3. Penerapan (<i>Implementation</i>)
	4. Pemeliharaan (<i>Maintenance</i>)

3.3 Metodologi

Metodologi (metode metode) adalah rincian secara menyeluruh dari siklus pengembangan sistem informasi yang mencakup :

- 1) Langkah demi langkah tugas dari masing-masing tahapan
- 2) Aturan yang harus dijalankan oleh individu dan kelompok dalam melaksanakan setiap tugas
- 3) Standar kualitas dan pelaksanaan dari masing-masing tugas
- 4) Teknik-teknik pengembangan yang digunakan untuk masing-masing tugas ini berkaitan dengan teknologi yang digunakan oleh pengembang.

Metodologi (metode-metode) adalah rincian secara menyeluruh dari siklus pengembangan sistem informasi

Berikut ini beberapa metode yang populer digunakan dalam membangun sistem informasi akuntansi atau sistem informasi lainnya.

3.3.1 Metode System Development Life Cycle (SDLC)

SDLC adalah salah satu metode pengembangan sistem informasi yang populer pada saat sistem informasi pertama kali berkembang. Metode SDLC merupakan metode pengembangan sistem informasi yang pertama kali digunakan. karena itulah metode ini dikatakan sebagai metode tradisional dalam pengembangan sistem informasi. Selanjutnya berbagai metode pengembangan sistem informasi dikembangkan. Metode SDLC hanyalah salah satu dari ratusan metode pengembangan sistem informasi yang ada di dunia saat ini. Bagi seorang analis profesional yang bergerak dalam bidang pengembangan sistem informasi, maka mengetahui berbagai metode yang berkembang seputar pengembangan sistem informasi merupakan hal yang sangat penting, agar dapat memilih metode mana yang paling tepat untuk dapat digunakan dalam menangani masalah sistem informasi akuntansi suatu perusahaan.

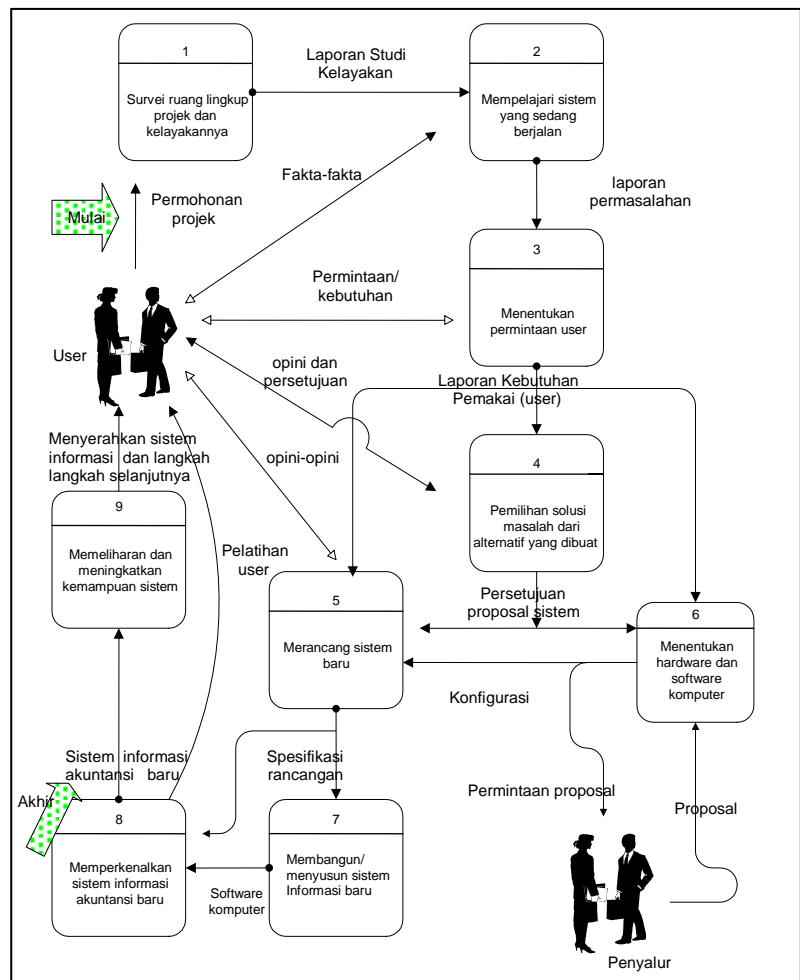
Metode *System Development Life Cycle (SDLC)* adalah tahap-tahapan pekerjaan yang dilakukan oleh analis sistem informasi dan programmer dalam membangun sistem informasi. SDLC juga merupakan alat untuk manajemen proyek yang bisa digunakan untuk merencanakan, memutuskan dan mengontrol proses pengembangan sistem informasi. Metode SDLC ini seringkali dinamakan juga sebagai proses pemecahan masalah, yang langkah-langkahnya meliputi:

SDLC adalah tahap-tahapan pekerjaan yang dilakukan oleh analis sistem dan programmer dalam membangun sistem informasi.

- Melakukan survei dan menilai kelayakan proyek pengembangan sistem informasi,
- Mempelajari dan menganalisis sistem informasi yang sedang berjalan,
- Menentukan permintaan pemakai sistem informasi,
- Memilih solusi atau pemecahan masalah yang paling baik,

- Menentukan perangkat keras (*hardware*) dan perangkat lunak (*software*) komputer,
- Merancang sistem informasi baru,
- Membangun sistem informasi baru,
- Mengkomunikasikan dan mengimplementasikan sistem informasi baru,
- Memelihara dan melakukan perbaikan/peningkatan sistem informasi baru bila diperlukan.

Gambar 3.1 Tahapan metode *systems development life cycle*



Melakukan Survei dan Menilai Kelayakan Projek

Tahap ini disebut juga sebagai tahap penelaahan awal atau tahap studi kelayakan yang diperlukan untuk mengetahui memadai atau tidaknya sumber daya yang akan dipergunakan pada fase-fase pengembangan selanjutnya. Pada tahap ini akan ditentukan ruang lingkup projek bagi semua pemakai sistem informasi dari berbagai tingkat pertanggungjawaban, meneliti masalah dan berbagai kemungkinan adanya kendala dari segi teknik dan bisnis, menentukan sasaran projek dan menentukan solusi yang mungkin diterapkan. Hasil dari survey adalah laporan kelayakan yang berisi temuan-temuan, rekomendasi, pertimbangan biaya dan manfaat yang akan diperoleh. Temuan-temuan ini sebelum dilaksanakan harus diketahui oleh komite pengawas (*steering committee*).

Masukan: permohonan Projek

Keluaran: Laporan studi kelayakan

Mempelajari dan Menganalisis Sistem informasi yang Sedang Berjalan

Tahap mempelajari (studi) sistem informasi yang sedang berjalan sangat berguna untuk mengetahui sebab dan akibat yang ditimbulkan oleh masalah, kesempatan dan pengarahannya yang terjadi. Jadi untuk menganalisis ketiga unsur tersebut maka mempelajari sistem informasi yang sedang berjalan sangat diperlukan. *Output* dari tahap ini akan menghasilkan laporan yang mengungkapkan adanya berbagai permasalahan (*problem statement*).

Masukan: -Laporan studi kelayakan
-Fakta-fakta

Keluaran: Laporan permasalahan

Menentukan Permintaan Pemakai Sistem informasi

Analisis seringkali melupakan tahapan ini. Umumnya setelah melakukan survei dan mempelajari masalah yang terjadi dari sistem informasi yang tengah berjalan, analisis langsung saja membuat alternatif-alternatif solusi dan menterjemahkannya dalam program komputer. Hal inilah yang sering menimbulkan keluhan-keluhan dari pemakai sistem informasi bahwa sistem informasi yang disusun analisis tidak dapat digunakan dan tidak sesuai dengan permintaan. Jelaslah bahwa mengetahui keinginan pemakai sistem informasi merupakan kunci sukses dari pembuatan atau pengembangan sistem informasi. Tujuan dari penentuan keinginan pemakai sistem informasi adalah untuk mengetahui apa yang diharapkan pemakai sistem dari sistem informasi yang baru. Secara umum keinginan para pemakai sistem informasi dari sistem informasi baru meliputi *data (input)*, pemrosesan (*processing*) dan hasil (*output*). Hal terpenting adalah bahwa garis besar sistem informasi baru tersebut di atas sebelum dibuat detail sistem informasinya harus mendapat persetujuan dari para pemakai sistem informasi. Pada tahapan ini analisis sistem informasi akan mengeluarkan laporan permintaan (*requirement statement*) dari pemakai sistem informasi yang akan dijadikan dasar untuk pembuatan alternatif pemecahan masalah.

Masukan: -Laporan permasalahan
-Permintaan/kebutuhan

Keluaran: Laporan kebutuhan pemakai

Memilih Pemecahan Masalah Sistem yang Terbaik.

Pemecahan masalah harus berdasarkan kepada permintaan-permintaan dan hasil analisis terhadap permintaan tersebut.

Masukan: -Laporan kebutuhan pemakai
-Opini dan persetujuan

Keluaran: Persetujuan proposal sistem

Mengembangkan sistem informasi dilakukan berdasarkan keinginan pemakai sistem. Para pemakai sistem akan membantu analis sistem dalam menentukan bagaimana sebaiknya sistem informasi berbasis komputer harus dibuat dan dioperasikan agar sesuai dengan kebutuhan pemakai. Seorang analis yang baik sebelum memberikan berbagai alternatif pemecahan masalah, akan memperhatikan permintaan-permintaan, menganalisis permintaan dan menawarkan pemecahan masalah yang baik berdasarkan hasil analisisnya. Pemakai (*End-user*) selanjutnya akan mengevaluasi berbagai alternatif pemecahan masalah yang diajukan secara teknik, operasional dan ekonomis. Secara formal tahap pemilihan alternatif solusi ini akan dibuat dalam proposal pengembangan sistem informasi, dan proposal yang telah disetujui akan dipergunakan sebagai dasar menuju pada langkah pengembangan sistem informasi selanjutnya.

Merancang Sistem Informasi Baru

Masukan: -Persetujuan proposal sistem
-Opini-opini

Keluaran: Spesifikasi rancangan

Setelah memahami apa yang diinginkan pemakai (*end-user*) sistem informasi yang akan dibangun, analis sistem informasi harus memahami bagaimana menterjemahkan keinginan pemakai sistem informasi tersebut kedalam bahasa komputer, untuk selanjutnya mulailah ia dapat merancang sistem informasi baru. Perancangan sistem informasi baru umumnya meliputi, *output, input, file-file, database*, komputer dan bahasa yang digunakan, metode dan prosedur serta pengendalian intern. Analis sistem informasi dalam merancang sistem informasi harus memiliki kemampuan dalam hal pemrograman komputer, hasil dari tahap perancangan ini adalah spesifikasi desain. Alternatif lain yang dapat digunakan adalah dengan membuat prototipe dari sistem informasi yang akan disusun. Penyusunan prototipe ini biasanya menggunakan bahasa komputer tingkat tinggi yang biasa digunakan untuk aplikasi pemrograman seperti dBase, Visual Foxpro, Oracle atau bahasa lainnya.

Menentukan Hardware dan Software Komputer

Masukan: -Laporan kebutuhan pemakai
-Persetujuan proposal sistem

Keluaran: Konfigurasi yang sesuai

seperti seorang arsitek yang telah disetujui membangun sebuah rumah, maka selanjutnya ia akan menentukan material yang akan dipergunakan. Demikian juga dengan seorang analis sistem informasi, setelah proposal pengembangan sistem informasi yang diajukan disetujui, maka akan menentukan *hardware* dan *software* yang akan digunakan, dan bagaimana cara mendapatkannya .

Membangun/Menyusun Sistem Informasi Baru

Apabila pemakai sistem informasi telah menyetujui rancangan yang diajukan maka mulailah analisis membangun/menyusun sistem informasi baru. Waktu yang dibutuhkan untuk membangun sistem informasi baru ini biasanya cukup lama. Tahap pembangunan sistem informasi ini dapat dilakukan oleh programmer dan peranan analisis sistem informasi pada tahap ini lebih banyak memberikan pengarahan.

Masukan: Spesifikasi rancangan

Keluaran: *Software* komputer yang sesuai dengan kebutuhan.

Memperkenalkan Sistem Informasi Akuntansi Baru

Hasil dari penyusunan sistem informasi (akuntansi/manajemen) adalah sebuah *software* komputer yang siap digunakan dan sesuai dengan kebutuhan *user*, untuk selanjutnya analisis harus memperkenalkan paket sistem informasi baru tersebut untuk dioperasikan. Pada penerapan sistem informasi baru, analisis harus benar-benar berperan sebagai perantara yang dapat membantu pemakai sistem untuk berpindah dari sistem informasi lama ke sistem informasi baru. Pada tahap ini pelatihan secara tertulis maupun praktek harus diberikan oleh analisis, agar pemakai dapat mengoperasikan sistem informasi baru.

Masukan: *Software* komputer yang sesuai dengan kebutuhan.

Keluaran: Sistem informasi akuntansi baru

Memelihara dan Meningkatkan Sistem Informasi

Tugas analisis sistem informasi dalam memecahkan masalah yang dihadapi organisasi belumlah selesai walaupun sistem informasi baru telah disusun dan telah diterapkan dalam aktivitas organisasi, meskipun masalah yang dihadapi oleh organisasi telah terpecahkan, tetapi analisis sistem informasi memiliki tanggungjawab untuk terus melakukan pengawasan dan pengembangan, melalui pemeliharaan dan peningkatan sistem informasi. Pemeliharaan yang dilakukan analisis adalah dengan melakukan perbaikan-perbaikan pada kesalahan-kesalahan atau kegagalan-kegagalan yang timbul dalam penggunaan sistem informasi. Banyak kegagalan yang muncul pada saat diterapkan dan tidak muncul pada tahap-tahap sebelumnya. Sedangkan yang dimaksud dengan peningkatan adalah penambahan fasilitas sistem informasi seperti pada penyajian laporan, penggabungan dengan sistem informasi lain dan tampilan-tampilan baru di layar.

Masukan: Sistem informasi akuntansi baru

Keluaran: Sistem informasi akuntansi dan langkah selanjutnya

Tahap-tahapan pada pengembangan sistem informasi tidak dapat dilakukan secara terpisah atau sendiri-sendiri, tetapi satu sama lain harus saling berkaitan, dan dapat saja terjadi pada saat suatu tahap belum selesai tahap berikutnya sudah dapat dilaksanakan. Misalnya pada saat suatu permintaan dari pemakai telah ditentukan maka analisis sudah dapat mulai mencari solusi, melalui pengidentifikasi-an dan pengevaluasian alternatif solusi yang diusulkan. Tetapi perlu diingat bahwa walaupun tahap-tahap tersebut bisa berjalan bersamaan, sistem informasi baru tersebut tidak dapat diterapkan, sebelum teknologi yang mendukungnya dimiliki.

3.3.2 Metode Prototyping

Secara umum tujuan pengembangan sistem informasi adalah untuk memberikan kemudahan dalam penyimpanan informasi, mengurangi biaya dan menghemat waktu, meningkatkan pengendalian, mendorong pertumbuhan, meningkatkan produktivitas serta profitabilitas organisasi. Dalam beberapa tahun terakhir ini peningkatan produktivitas organisasi ini dibantu dengan berkembangnya teknologi komputer, baik *hardware* maupun *softwarena*. Tetapi tidak semua kebutuhan sistem informasi dengan komputer itu dapat memenuhi kebutuhan dan menyelesaikan masalah yang dihadapi organisasi. Keterbatasan sumber daya dan anggaran pemeliharaan memaksa para pengembang sistem informasi untuk menemukan jalan untuk mengoptimalkan kinerja sumber daya yang telah ada.

Karakteristik sistem informasi akuntansi yang lengkap tergantung dari masalah yang dihadapi, proses pengembangannya dan tenaga kerja yang akan dikembangkannya

Karakteristik dari suatu sistem informasi akuntansi yang lengkap tergantung dari masalah yang dihadapi, proses pengembangannya dan tenaga kerja yang akan dikembangkannya. Seiring dengan perkembangan permasalahan karena berubahnya lingkungan yang berdampak kepada perusahaan maka yang menjadi parameter proses pengembangan sistem informasi yaitu masalah yang dihadapi, sumber daya yang tersedia dan perubahan, sehingga hasil pengembangan sistem informasi akuntansi baik yang diharapkan oleh perorangan maupun oleh organisasi turut berubah.

Metode prototyping sebagai suatu paradigma baru dalam pengembangan sistem informasi akuntansi, tidak hanya sekedar suatu evolusi dari metode pengembangan sistem informasi yang sudah ada, tetapi sekaligus merupakan revolusi dalam pengembangan sistem informasi akuntansi

Perubahan tersebut pada akhirnya menimbulkan ketidakpastian dan menambah kompleks/rumit masalah yang dihadapi oleh para analis sistem informasi. Metode tradisional seperti SDLC dianggap tidak lagi mampu memenuhi tantangan perubahan dan kompleksnya masalah yang dihadapi tersebut. Sekitar awal tahun delapan puluhan, para profesional dibidang sistem informasi memperkenalkan satu metode pengembangan sistem informasi baru, yang dikenal dengan nama metode *prototyping*.

Metode *prototyping* sebagai suatu paradigma baru dalam pengembangan sistem informasi akuntansi, tidak hanya sekedar suatu evolusi dari metode pengembangan sistem informasi yang sudah ada, tetapi sekaligus merupakan revolusi dalam pengembangan sistem informasi akuntansi. Metode ini dikatakan revolusi karena merubah proses pengembangan sistem informasi yang lama (SDLC).

Menurut literatur, yang dimaksud dengan prototipe (*prototype*) adalah "model pertama", yang sering digunakan oleh perusahaan industri yang memproduksi barang secara masal. Tetapi dalam kaitannya dengan sistem informasi definisi kedua dari Webster yang menyebutkan bahwa "*prototype is an individual that exhibits the essential features of later type..*", yang bila diaplikasikan dalam pengembangan sistem informasi akuntansi dapat berarti bahwa prototipe tersebut adalah sistem informasi yang menggambarkan hal-hal penting dari sistem informasi yang akan datang. Prototipe sistem informasi bukanlah merupakan sesuatu yang lengkap, tetapi sesuatu yang harus dimodifikasi kembali, di-

kembangkan, ditambahkan atau digabungkan dengan sistem informasi yang lain bila perlu.

Dalam beberapa hal pengembangan *software* berbeda dengan produk-produk manufaktur, setiap tahap atau fase pengembangan sistem informasi merupakan bagian yang tidak terpisahkan dari seluruh proses yang harus dilakukan. Proses ini umumnya hanya untuk satu produk dan karakteristik dari produk tersebut tidak dapat ditentukan secara pasti seperti produk manufaktur, sehingga penggunaan "model pertama" bagi pengembangan *software* tidaklah tepat. Istilah *prototyping* dalam hubungannya dengan pengembangan *software* sistem informasi akuntansi lebih merupakan suatu proses bukan prototipe sebagai suatu produk.

Prototipe merupakan model kerja dari sebuah sistem informasi akuntansi yang belum lengkap. Para pengembang sistem informasi, melakukan pertemuan-pertemuan intensif dengan *user* untuk menampung informasi yang akan dijadikan dasar dalam menyusun prototipe dari sistem informasi akuntansi yang akan disajikan kelak. Prototipe yang dihasilkan kemudian dipresentasikan kepada *user* dan *user* diberikan kesempatan untuk memberi masukan-masukan sehingga sistem informasi akuntansi yang dihasilkan betul-betul sesuai dengan keinginan dan kebutuhan *user*. Perubahan dan presentasi prototipe ini dapat dilakukan berkali-kali sampai dicapai kesepakatan bentuk sistem informasi akuntansi yang akan diterapkan.

Istilah prototipe, pada saat pengembangan sistem informasi, menghasilkan suatu demonstrasi praktis dari *software* komputer yang akan dihasilkan. Jadi seperti dijelaskan diatas, prototipe merupakan model kerja dari sebuah sistem informasi akuntansi yang belum lengkap tapi telah menampung hal-hal yang penting dari suatu sistem informasi akuntansi. Metode *prototyping* dirancang agar dapat menerima perubahan-perubahan dalam rangka menyempurnakan prototipe yang sudah ada sehingga pada akhirnya dapat menghasilkan sistem informasi akuntansi yang dapat diterima dan perubahan-perubahan yang terjadi dianggap merupakan bagian dari proses pengembangan itu sendiri.

Para pengembang sistem informasi melakukan pertemuan-pertemuan intensif dengan pemakai untuk menampung informasi yang akan dijadikan dasar dalam menyusun model prototipe dari sistem informasi (akuntansi) yang akan disajikan kelak. Prototipe yang dihasilkan kemudian dipresentasikan kepada *user* dan *user* diberi kesempatan untuk memberikan masukan-masukan sehingga hasilnya diharapkan betul-betul sesuai dengan keinginan dan kebutuhan *user*. Perubahan dan presentasi prototipe ini dapat dilakukan berkali kali sampai dicapai kesepakatan bentuk sistem informasi yang akan diterapkan.

Keterlibatan *user* dalam proses pengembangan sistem informasi merupakan bagian dari proses pengembangan yang akan mempengaruhi kualitas akhir dari sistem informasi akuntansi yang akan dihasilkan. Untuk mempercepat proses pengembangan sistem informasi akuntansi dengan menggunakan metode *proto-*

typing, dalam pelaksanaannya perlu dibantu oleh perangkat lunak yang mendukung pembuatan *software* seperti *DBMS (4GL)* dan prosesor berkemampuan cukup tinggi.

Karakteristik Metode Prototyping

Ada empat langkah yang menjadi karakteristik metode *prototyping* yaitu, pemilihan fungsi (*function selection*), penyusunan sistem informasi (*construction*), evaluasi (*evaluation*) dan penggunaan selanjutnya (*further use*), yang uraian lengkapnya sebagai berikut:

Empat langkah yang menjadi karakteristik metode *prototyping* :

- Pemilihan fungsi
 - Penyusunan sistem informasi
 - Evaluasi, dan
 - Penggunaan selanjutnya
- Pemilihan fungsi (*functional selection*) mengacu pada pemilihan fungsi yang harus ditampilkan oleh *prototyping*. Pemilihan harus selalu dilakukan berdasarkan pada tugas-tugas yang relevan yang sesuai dengan contoh kasus yang akan diperagakan. Walaupun luasnya gambaran yang disajikan oleh prototipe tidak selamanya sama dengan hasil akhir dari sistem informasi akuntansi yang akan disusun, tetapi prototipe ini harus mewakili bentuk yang akan disusun. Terdapat dua perbedaan antara cakupan fungsi prototipe dengan hasil akhir yaitu :
 - Fungsi sistem informasi yang diterapkan mendekati bentuk akhirnya, tetapi hanya fungsi yang terpilih saja yang digunakan (*vertical prototipe*);
 - Fungsi-fungsi tidak ditampilkan secara terinci seperti yang akan digunakan pada hasil akhir, tetapi dapat diperagakan (*horizontal prototipe*).

Pada beberapa bagian dari prototipe, seringkali kedua elemen tersebut disajikan secara bersama-sama.
 - Penyusunan sistem informasi (*construction*) bertujuan untuk memenuhi permintaan akan tersedianya prototipe. Secara umum, prototipe ini harus lebih kecil dari yang dibutuhkan dalam mengembangkan hasil akhir sistem informasi akuntansi yang akan digunakan. Hal ini dapat dilakukan dengan menyesuaikan baik pemilihan fungsi maupun teknik dan perangkat yang tersedia untuk penyusunan prototipe. Dalam menyusun prototipe, kualitas dari sistem informasi akuntansi akhir dapat diabaikan seperti kelayakan dan efisiensi keamanan *data*, asal dapat diperagakan misalnya prototipe ini dapat digunakan untuk melihat apakah waktu yang dibutuhkan sistem informasi tersebut sesuai dengan ritme kerja *user*, efisiensi dalam hal ini merupakan hal yang tidak dapat diabaikan.
 - Evaluasi (*evaluation*) harus dipertimbangkan agar menerima masukan-masukan untuk proses pengembangan selanjutnya. Seluruh unsur dalam pengembangan sistem informasi harus

yakin bahwa prototipe ini dapat dievaluasi, termasuk adanya partisipasi dari kelompok-kelompok yang relevan yang mungkin akan menggunakan sistem informasi akuntansi ini. Evaluasi harus dilakukan setelah dilaksanakannya pelatihan. Pelaksanaan evaluasi harus didasarkan pada dokumen yang menjelaskan tentang kriteria hal-hal yang perlu dievaluasi dan langkah-langkah kerja yang terinci dari sistem informasi manajemen yang akan digunakan. Evaluasi juga harus dilakukan baik untuk penggunaan sistem informasi secara individual maupun secara bersama-sama antara beberapa *user*, sehingga tidak mengabaikan timbulnya masalah dalam komunikasi antara para pemakai dan kendala dalam penggunaan perangkat komputer.

Ada beberapa kemungkinan dalam penggunaan selanjutnya (*further use*) dari prototipe. Hal ini sangat tergantung pada pengalaman yang dimiliki dengan prototipe ini pada kondisi lingkungan tertentu, seringkali prototipe ini digunakan sebagai alat untuk belajar dan selanjutnya tidak dipergunakan lagi, tetapi mungkin juga prototipe ini merupakan bagian dari sistem informasi akuntansi yang akan digunakan kelak.

Jika prototipe ini digunakan sebagai alat untuk belajar, maka penyusunan prototipe ini harus benar-benar dilakukan dengan hati-hati. Ada beberapa aspek yang harus diperhatikan yaitu :

- **Manfaat awal** - prototipe harus memberikan manfaat pada pengembangan *software* sehingga dapat ditawarkan pada semua unsur yang berkepentingan seperti pengembang, konsumen, dan *user*.
- **Peragaan, Evaluasi dan Modifikasi** - prototipe harus dapat diperagakan kepada *user*. Peragaan ini harus dapat menyajikan bagaimana proses kerja *user*, seperti harus otentik dan tidak menimbulkan masalah sehingga evaluasi yang dilakukan pun relevan. Prototipe harus mudah dirubah baik direvisi maupun ditambah setelah dilakukan evaluasi, sehingga memungkinkan dilakukannya berbagai modifikasi.
- **Pengajaran dan pelatihan** - setelah dilakukan evaluasi dan modifikasi, prototipe yang baik harus dapat dipelajari untuk persiapan *user* saat bekerja dengan sistem informasi manajemen yang sudah jadi.
- **Kesepakatan** - perlu diperhatikan bila selama peragaan prototipe ada masukan-masukan dari *user*, maka harus diarahkan untuk membuat kesepakatan pada sistem informasi yang akan dihasilkan kelak. Kesepakatan yang dibuat sangat tergantung pada pendekatan yang dipergunakan dalam menyusun *prototyping* ini.

Jenis-jenis Prototyping

Jenis-jenis *prototyping*:

- *Feasibility*
- *Requirement*
- *Design*
- *Implementation*

Feasibility prototyping digunakan untuk menguji kelayakan teknologi yang akan digunakan untuk sistem informasi akuntansi yang akan disusun.

Requirement prototyping juga disebut sebagai *discovery prototyping*, digunakan untuk mengetahui kebutuhan aktivitas bisnis *user*.

Desain prototyping digunakan untuk mendorong perancangan sistem informasi akuntansi yang akan digunakan.

Bila dikaitkan dengan siklus pengembangan sistem informasi akuntansi secara umum, beberapa pakar berpendapat ada empat tahap dalam siklus pengembangan sistem informasi yang dapat menggunakan metode *prototyping*. Penggunaan metode *prototyping* dalam beberapa siklus sistem informasi akuntansi ini dikelompokkan menjadi empat jenis yaitu:

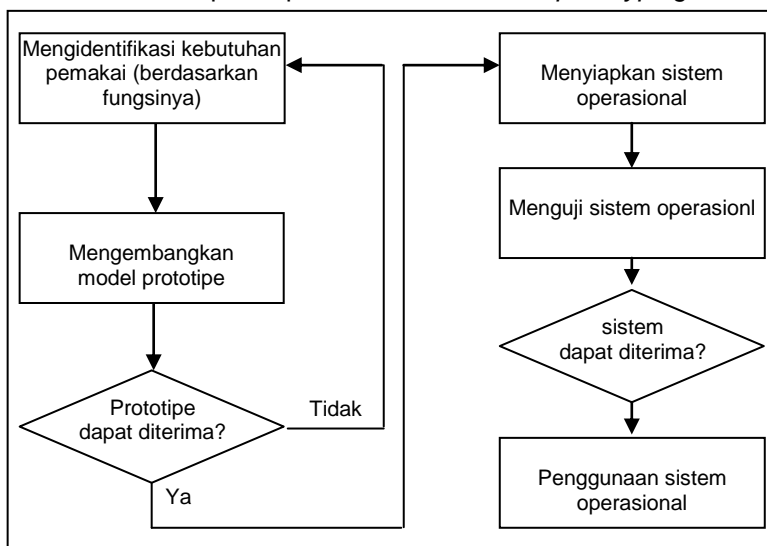
- **Feasibility prototyping** - digunakan untuk menguji kelayakan dari teknologi yang akan digunakan untuk sistem informasi akuntansi yang akan disusun. Misalnya bila ingin menguji apakah komputer dapat digunakan untuk mencatat sendiri permintaan pembelian oleh salesman untuk menguji hal tersebut analis harus membuat satu prototipe yang dapat digunakan untuk melihat bagaimana reaksi salesman bila mereka dapat memasukan sendiri permintaan pembelian pada jaringan komputer. Pengisian *data* ini biasanya dilakukan oleh bagian gudang. Dari hasil pengujian itu dapat diambil keputusan apakah teknik yang diperkenalkan dapat digunakan atau tidak.
- **Requirement prototyping** - juga disebut sebagai *discovery prototyping*, digunakan untuk mengetahui kebutuhan aktivitas bisnis *user*. Jenis ini ditunjukkan untuk merangsang pola berfikir *user*. Konsepnya adalah, *user* akan mengetahui apa yang mereka inginkan, bila mereka melihatnya. Pada tahap penentuan kebutuhan, analis sistem informasi membuat satu tampilan layar dan meminta *user* untuk memberikan reaksinya. Hal penting yang perlu diperhatikan pada saat melaksanakan siklus ini, karena *user* mungkin akan berfikir bahwa ini adalah sistem informasi akuntansi yang kelak akan digunakan. Bila hal ini terjadi maka kemungkinan masalah yang akan timbul adalah: 1) perhatian *user* terfokus pada format tampilan atau laporan yang disajikan; dan/atau 2) *user* mempertimbangkan untuk menjadikan prototipe ini sebagai sistem informasi manajemen yang akan diterapkan.
- **Desain prototyping** - juga sering disebut *behaviorial prototyping*, digunakan untuk mendorong perancangan sistem informasi akuntansi yang akan digunakan. Dengan menggunakan rancangan prototipe, *user* diharapkan dapat mengevaluasi apakah prototipe ini dapat digunakan sebagai bagian dari sistem informasi akuntansi yang akan digunakan. *User* dalam hal ini diharapkan untuk mengevaluasi kemudahan mempelajari dan menggunakannya serta bagaimana dan prosedur apa yang akan diterapkan dalam menggunakan sistem informasi akuntansi ini.
- **Implementation prototyping** - atau disebut juga *production prototyping*, adalah lanjutan dari rancangan prototipe, prototipe ini langsung disusun sebagai sistem informasi manaje-

men yang akan digunakan. Pada umumnya prototipe yang diterapkan ini membutuhkan *data* yang terperinci, fasilitas pengeditan *data*, keamanan dan pesan-pesan penolong. Rincian lainnya akan ditambahkan kemudian jika prototipe ini telah menjadi satu produk sistem informasi akuntansi yang lengkap. Dengan penggunaan bahasa pemrograman yang semakin canggih yaitu *4GLs*. Jenis prototipe ini menjadi semakin populer karena dapat menghasilkan perangkat lunak yang menambah kecepatan pengolahan *data*.

Implementation prototyping atau disebut juga *production prototyping*, adalah lanjutan dari rancangan prototipe, prototipe ini langsung disusun sebagai sistem informasi mana jemen yang akan digunakan.

Beberapa pakar lain berpendapat bahwa metode *prototyping* merupakan metode pengembangan sistem informasi untuk membangun sistem informasi akuntansi perusahaan secara keseluruhan. Berikut ini tahap-tahapannya.

Gambar 3.2 Tahap-tahapan dalam metode *full prototyping*



Teknik dalam Metode Prototyping

Ada beberapa teknik yang relevan dan dapat dipergunakan dalam penerapan metode *prototyping*, Teknik-teknik tersebut adalah perancangan model, perancangan dialog dan simulasi.

- **Perancangan model** adalah bagian yang terpenting dalam seluruh strategi *prototyping* yang digunakan sebagai alat untuk menjadikan model menjadi sistem informasi yang sebenarnya. Strategi ini meliputi simulasi interaktif, struktur pemrograman dan pengembangan sistem informasi secara bertahap. Dari semua kegiatan, pembuaan model membantu memperoleh perubahan-perubahan secara bertahap bagian-bagian model/prototipe dengan komponen sistem informasi yang sebenarnya.

Teknik-teknik Prototyping:

- Perancangan model
- Perancangan dialog
- Simulasi

- **Perancangan dialog** disusun agar keterlibatan *user* menjadi jelas dan fleksibel. Aspek dalam perancangan dialog mencakup keseluruhan unsur yang harus didialogkan, seperti pemilihan perintah-perintah sistem informasi, *layout* tampilan layar, penanganan masalah-masalah khusus dan kemungkinan disediakan prosedur untuk menjalankan sistem yang diterapkan.
- **Simulasi** dalam *prototyping* dilakukan untuk menunjukkan bagaimana cara kerja sebuah sistem informasi akuntansi yang akan diterapkan kelak. Unsur-unsur yang disimulasikan biasanya menunjukkan bagaimana *data* diorganisasikan serta waktu yang diperlukan untuk mengoperasikan sistem informasi yang akan digunakan. Sebagai suatu teknik dan perangkat dari *prototyping*, simulasi sangat tergantung pada model yang dirancang, yang memungkinkan dilakukannya simulasi bagian tertentu tanpa mengganggu sistem informasi secara keseluruhan.

Metode *prototyping* tidak membutuhkan satu perangkat khusus atau tambahan investasi. Metode ini dapat menggunakan perangkat yang sebelumnya telah digunakan. Perangkat pendukung kerja metode *prototyping* adalah bahasa pemrograman tingkat tinggi *4th generation languages (4GL)*, sistem manajemen *database (DBMS)*, sistem dialog (*user interface*), penggunaan bahasa instruksi yang spesifik dan simbol-simbol pelaksanaan sistem informasi, dan tentunya hal yang paling utama adalah tersedianya perangkat komputer yang memadai.

Seperti metode pengembangan sistem informasi lainnya, metode *prototyping* juga selain memberikan dampak yang menguntungkan juga mengundang masalah. Profesional dibidang pengembangan sistem informasi akuntansi melihat beberapa keuntungan penggunaan metode *prototyping* diantaranya adalah :

- **End-user** dapat berpartisipasi secara lebih aktif dalam pengembangan sistem informasi. Sehingga jadwal pelatihan bisa dihilangkan.
- **Penentuan kebutuhan/keinginan** lebih mudah diwujudkan.
- **Mengurangi proses** persetujuan rancangan sistem informasi.
- **Perancangan dengan menggunakan metode prototyping** mempersingkat waktu pengembangan sistem informasi akuntansi sehingga dapat menghemat biaya.

Selain beberapa keunggulan diatas *prototyping* juga mempunyai beberapa kelemahan. Biasanya kelemahan-kelemahan ini dimunculkan oleh pihak-pihak yang kontra terhadap penggunaan metode *prototyping*. Kelemahan tersebut diantaranya yaitu :

- Metode *prototyping* membuat proses analisis dan perancangan menjadi terlalu singkat. Analisis sistem informasi dalam hal ini mungkin saja kurang memahami masalah-masalah dan kebutuhan-kebutuhan yang sebenarnya dari organisasi.
- Metode *Prototyping* mengesampingkan alternatif pemecahan masalah. Analisis sistem informasi tidak membuat alternatif lain ketika prototipe pertama yang disajikan mendapat reaksi positif dari *user*.
- Penerapan sistem informasi dengan menggunakan metode *prototyping* biasanya kurang fleksibel dalam menghadapi perubahan-perubahan.
- Tidak selamanya prototipe yang dihasilkan oleh metode *prototyping* mudah untuk diubah. Penggunaan 4GLs seringkali tidak terstruktur, sulit dibaca dan rancangannya terbatas.
- Umumnya prototipe ini terlalu cepat selesai, teknologi yang digunakan tidak mudah dipahami oleh *user* karena hanya sedikit memberikan kesempatan partisipasi *user*.

Metode prototyping pada pelaksanaannya perlu didukung dengan teknik-teknik khusus yang memadai agar penggunaannya bisa sukses.

3.3.3 Metode Rapid Application Development / RAD

Telah kita ketahui bahwa disamping memiliki kelebihan-kelebihan beberapa metode dan teknik pengembangan sistem informasi akuntansi juga memiliki kelemahan-kelemahan. Untuk menutupi kelemahan-kelemahan yang ditimbulkan oleh masing-masing metode dan teknik pengembangan sistem informasi tersebut perlu adanya metode lain yang dianggap memadai. Salah satu metode yang diharapkan dapat mengatasi masalah-masalah yang selama ini dihadapi oleh metode *prototyping* dengan teknik JAD adalah metode *rapid application development* (RAD) atau metode pengembangan aplikasi secara cepat.

Metode *Rapid Application Development* (RAD) adalah penggabungan beberapa metode atau teknik terstruktur (khususnya dalam pengolahan *data* untuk menghasilkan informasi), misalnya dengan mengintegrasikan metode *prototyping*, metode SDLC dan teknik *joint application development* untuk mempercepat pengembangan sistem informasi.

Metode RAD menggunakan metode *prototyping* dan teknik terstruktur lainnya untuk menentukan kebutuhan *user* dan perancangan sistem informasi (akuntansi). Dengan menggunakan teknik yang terstruktur dalam pengembangan sistem informasi, analisis sistem pertama-tama akan menyusun *data* awal dan model pemrosesan *data* yang diinginkan, sedangkan prototipe digunakan untuk membantu analisis sistem informasi dan *user* dalam menentukan kebutuhan informasi yang sebenarnya serta memperbaiki *data* dan model pemrosesan yang digunakan.

Metode Rapid Application Development (RAD) adalah penggabungan beberapa metode atau teknik terstruktur.

Metode RAD menggunakan metode *prototyping* dan teknik terstruktur lainnya untuk menentukan kebutuhan *user* dan perancangan sistem informasi akuntansi

Siklus perancangan model-model yang kemudian dibuat prototipenya berjalan terus hingga dihasilkan satu kombinasi yang mendukung kebutuhan aktivitas bisnis. Pada awal tahun 1994, organisasi pemakai sistem dan penyalur *software* bekerjasama dengan akademisi membentuk Konsorsium Metode Pengembangan Sistem yang Dinamis (*Dynamic System Development Consortium/DSDM*). Konsorsium ini bertujuan untuk mengembangkan dan menindaklanjuti metode RAD. Sasarannya termasuk publikasi kerangka kerja metode RAD, mempromosikannya, menyelenggarakan pelatihan, mengeluarkan sertifikat dan lain-lain.

Versi pertama standar DSDM ini telah dipublikasikan pada awal tahun 1994. Standar tersebut menentukan tiga faktor utama dalam RAD yaitu: kelompok pemakai sistem harus memiliki staf senior yang benar-benar berdedikasi terhadap pengembangan sistem informasi yang memudahkan mereka dalam berhubungan dengan pengembangan sistem; tim pengembang sistem harus stabil dan memiliki kemampuan yang memadai; dan lingkup aplikasi harus komersial dengan penentuan-penentuan permintaan yang jelas dari kelompok pemakai sistem.

Sebagai tambahan bagi ketiga faktor tersebut di atas, DSDM merekomendasikan bahwa metode ini harus:

- Menentukan prioritas dari kebutuhan aktivitas bisnis, sebagai pembanding bagi kualitas karakteristik operasional sistem.
- Membuat sudut pandang sistem yang memenuhi pertanyaan apa tujuan dari prosedur yang dijalankan dan prosedur mana yang paling fleksibel dibandingkan dengan aktivitasnya sehingga dapat menjawab bagaimana pekerjaan tersebut harus dijalankan.
- Gunakan konfigurasi utama dari prosedur akuntansi, karena setiap perubahan harus direvisi.
- Bentuk motivasi kelompok agar lebih memahami aktivitas bisnis, bukan rincian dari tugas-tugas yang dilaksanakan.
- Integrasikan pengujian pada siklus pengembangan sistem.
- Hasil akhir harus difungsikan sesuai dengan waktu dan biaya yang telah dikeluarkan bukan dengan aktivitas yang dilakukan.
- Pusatkan kemungkinan timbulnya risiko pada fungsi sistem bukan pada bentuk sistem; dan
- Jadikan permintaan pemakai sistem sebagai dasar utama agar mudah dirubah selama pengembangan sistem dilaksanakan.
- Dari rekomendasi di atas terlihat bahwa DSDM menekankan pada produk yang dihasilkan bukan bagaimana aktivitas yang menghasilkan produk tersebut. Notasi-notasinya sangat terbuka dan fleksibel.

Proses Pengembangan

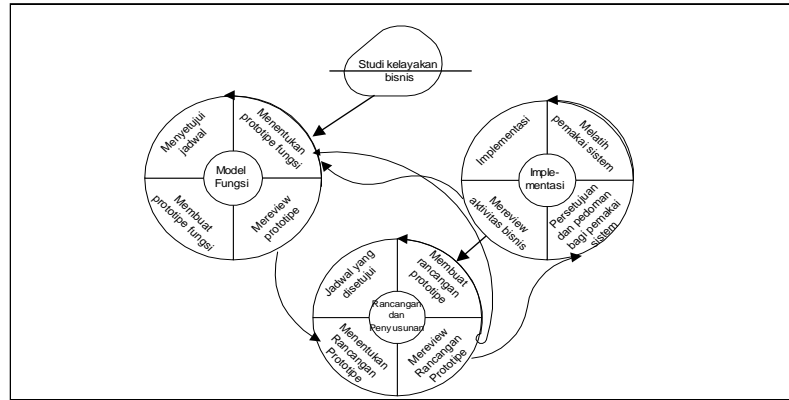
Siklus pengembangan DSDM terbagi dalam beberapa tahap, seperti terlihat pada gambar 3.3.

- **Tahap pertama** - pengembang sistem informasi mempelajari apakah proyek pengembangan sistem yang telah disusun dapat memenuhi kriteria RAD. Hasil dari evaluasi ini adalah berupa laporan studi kelayakan. Jika proyek tersebut layak, maka dibuatlah garis besar rencana pengembangan.
- **Tahap kedua** - pengembang sistem mempelajari aktivitas bisnis organisasi perusahaan, menentukan area bisnis serta fungsi-fungsi yang menjadi prioritas utama dalam rencana pembuatan prototipe.
- **Tahap ketiga** - pengembang sistem (analisis) membuat model dari fungsi-fungsi yang menjadi prioritas untuk menghasilkan prototipe atau model dari fungsi-fungsi tersebut. Selanjutnya analisis mulai merancang dan membangun tahap-tahap yang harus dilakukan untuk menghasilkan suatu prototipe sistem informasi yang dapat diuji untuk menjalankan semua fungsi-fungsi yang telah dibuat.
- **Tahap keempat** - analisis/pengembang sistem informasi memilih prototipe mana yang akan direview dan kemudian mereview prototipe tersebut. Setiap aspek dari tahap-tahap yang harus dievaluasi ini dibatasi oleh waktu yang telah ditentukan dengan tiga jenis pekerjaan yaitu meneliti, meningkatkan dan melakukan konsolidasi prototipe yang direview.
- **Tahap kelima (tahap terakhir)** - sistem diimplementasikan di lingkungan pemakai sistem, diikuti dengan pendokumentasian dan pelatihan. Setelah konsorsium mengembangkan struktur proses DSDM, kelompok kerja teknik melanjutkan dengan menetapkan tugas-tugas yang harus dilakukan serta menentukan aspek-aspeknya, termasuk didalamnya manajemen proyek, personel, perangkat dan tekniknya, jaminan kualitas dan penyediaan *software*. berikut ini adalah uraiannya:
- **Manajemen proyek** - Secara umum, aktivitas manajemen proyek meliputi, perkiraan sumber daya yang dibutuhkan dan penjadwalannya, serta memonitor aktivitas dan sumber daya proyek, termasuk juga penyusunan staf. Misalnya menentukan berapa lama waktu yang dibutuhkan untuk melaksanakan aktivitas X dengan menjawab pertanyaan seberapa jauh sebuah sistem dapat dibentuk dengan menggunakan Y orang.

Metode DSDM :

- Mempelajari apakah proyek pengembangan sistem memenuhi kriteria RAD.
- Mempelajari aktivitas bisnis perusahaan, menentukan area bisnis serta fungsi-fungsi yang menjadi prioritas.
- Membuat model dari fungsi-fungsi yang menjadi prioritas.
- Memilih prototipe mana yang direview
- Implementasi sistem informasi.

Gambar 3.3 Siklus Pengembangan DSDM



- **Personel** - Karena RAD memberi penawaran yang menarik bagi pengembang dan pemakai sistem, DSDM merekomendasikan untuk menggunakan tim kecil saja yang terdiri dari pengembang sistem dan pemakainya. Hal ini lebih memudahkan terciptanya komunikasi dan hubungan antar tim yang tingkat keberhasilannya lebih tinggi.
- **Perangkat dan teknik** - DSDM tidak menyusun satu perangkat atau teknik yang khusus bagi pengembangan sistem informasi. Penekanannya adalah bagaimana produk dihasilkan pada setiap tahapan, yang secara tidak langsung akan menentukan sendiri teknik apa yang sesuai untuk mendukung pembuatan produk termasuk struktur dan metode yang digunakan.
- **Jaminan kualitas** - Pada metode DSDM, jaminan kualitas dipergunakan pada pengenalan sistem yang disajikan sebagai permintaan kebutuhan-kebutuhan aktivitas bisnis. Walaupun penyajiannya tidak dibuat dalam bentuk spesifikasi formal, tetapi pengembang sistem akan dapat menyatukan tujuan ini dengan melakukan serangkaian pengujian sepanjang tahapan dari DSDM yang menitikberatkan pada komponen sistem yang akan menjamin keberhasilan aktivitas bisnis.
- **Penyediaan software** - Perangkat lunak yang digunakan dalam proses pengembangan dengan menggunakan metode DSDM biasanya diperoleh dengan bekerjasama dengan pihak lain yang menyediakan perangkat lunak ini dengan sistem kontrak kerja.

3.3.4 Metode Soft System

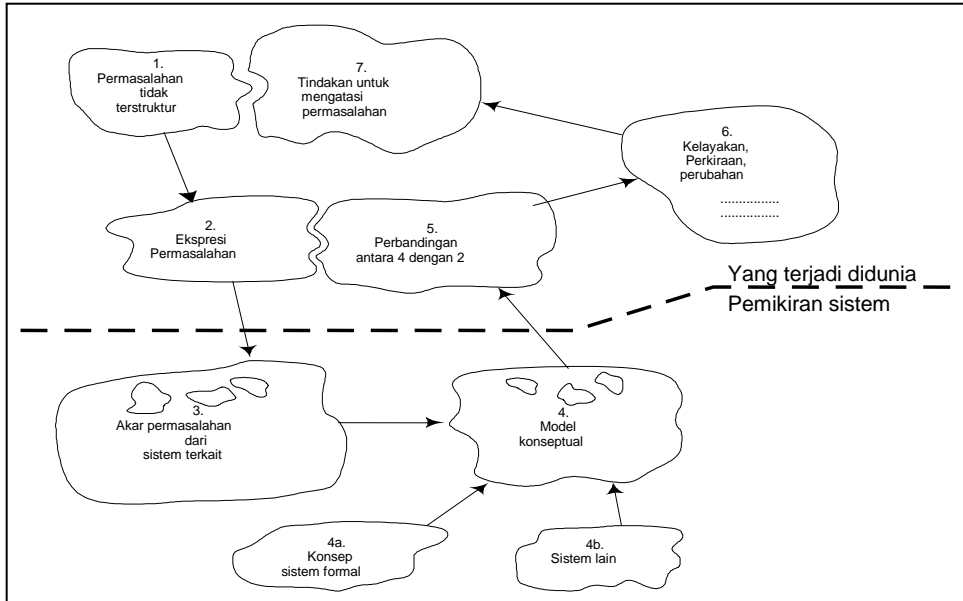
Metode-metode tradisional untuk menganalisis dan mengembangkan sistem informasi menurut beberapa ahli sistem informasi tidak lagi dapat memenuhi kebutuhan bila diterapkan dalam lingkungan yang berbeda budayanya, dimana komponen budaya ini menjadi bagian dari proses pengembangan sistem informasi.

Checkland telah menemukan suatu metode pengembangan sistem informasi yang dapat mengantisipasi adanya perbedaan budaya dimana suatu sistem informasi akan dikembangkan dan diterapkan, metode itu bernama *softsystem*.

Metode *Soft System (Soft System Methodology/SSM)* memiliki tujuh tahapan proses untuk menangani masalah-masalah dalam kehidupan sehari-hari, yang berdampak pada organisasi. Metode *soft system* yang dikembangkan oleh Checkland ini memisahkan masalah sehari-hari dengan pola berfikir suatu sistem informasi. Gambar 15.4 menunjukkan bagaimana tahapan-tahapan dalam metode *soft system* dipisahkan oleh satu garis pemisah. Tahap 1, 2, 5, 6 dan 7 berada di atas garis masalah sehari-hari serta tahap 3 dan 4 berada di bawahnya yang termasuk dalam proses berfikir sebuah sistem informasi akuntansi. Tahap 3 dan 4 dalam metode *soft system* merupakan tahap penting dimana ditentukan akar permasalahan (*root definition/RD*) dan disusunnya konsep dari model sistem informasi (*conceptual model/CM*), berikut ini penjelasannya:

- **Tahap 1** - Masalah relatif bagi setiap orang, karena itu metode ini tidak melihat masalah secara individu, tetapi dalam konteks situasi permasalahan yang merupakan gabungan dari beberapa masalah yang saling berkaitan dan berdasarkan pada apa yang kita lihat, masalah-masalah itu tidak terstruktur
- **Tahap 2** - Ahli sistem mencoba menstrukturkan permasalahan dengan mengekspresikan keterkaitan antara masalah-masalah yang muncul dengan menyusun apa yang disebut sebagai *problematique diagram* dan *rich picture* dan berdasarkan inilah ahli sistem akan memperoleh akar permasalahan dari sistem terkait.
- **Tahap 4** - disusun konsep model yang terdiri dari sistem informasi yang mungkin diterapkan dan strategi yang mungkin digunakan untuk menindaklanjuti masalah yang dihadapi.
- **Tahap 5** - adalah tahap membandingkan antara masalah yang ditentukan pada tahap 2 dengan konsep yang disusun pada tahap 4 untuk menyusun perubahan yang mungkin dilakukan.
- **Tahap 6** - dilakukan diskusi yang bertujuan untuk menghasilkan satu harapan dari sistem informasi dan serangkaian strategi yang sesuai dengan kultur yang ada untuk disesuaikan dengan masalah yang ada.
- **Tahap 7** - adalah realisasi dari tahap 6 dimana serangkaian proposal, strategi dan taktik disusun untuk membuat perubahan yang diharapkan dapat menyelesaikan masalah yang dihadapi. Perubahan yang disusun tidak hanya yang berhubungan dengan struktur dan proses masalah tetapi juga melibatkan nilai dan perilaku manusia yang terlibat didalamnya.

Gambar 3.4 Tahap-tahap metode soft system



Setelah seluruh tahapan dari metode ini lengkap dijalankan dan perubahan telah dilakukan, maka seperti halnya siklus sebuah sistem informasi, maka metode ini pun berjalan sebagai suatu siklus. Gambar 15.4 memperlihatkan bagaimana metode ini diterapkan dalam memecahkan masalah, yang terdiri dari sistem informasi yang digunakan untuk memecahkan masalah dan situasi dimana masalah tersebut muncul.

3.4 Teknik

Teknik adalah pendekatan bagaimana menggunakan alat dan peraturan yang melengkapi satu atau lebih tahapan-tahapan dalam siklus pengembangan sistem informasi

Teknik ada yang hanya diterapkan pada satu tahapan saja pada siklus pengembangan sistem informasi, tetapi mungkin juga diterapkan pada seluruh siklus pengembangan sistem informasi.

Teknik (*technique*) adalah pendekatan bagaimana menggunakan alat-alat dan peraturan-peraturan yang melengkapi satu atau lebih tahapan-tahapan dalam siklus pengembangan sistem informasi. Persamaan dari teknik ini adalah paradigma (*paradigm*).

Dari definisi-definisi tentang siklus, metodologi dan teknik, jelaslah bahwa siklus dalam pengembangan sistem informasi masih bisa diterima. Ada dua hal penting yang perlu diperhatikan dari uraian diatas, pertama, metodologi harus dapat mencakup semua tahapan dari siklus pengembangan sistem informasi, termasuk pada tahapan pemeliharaan. Kedua, kebanyakan metodologi modern menggunakan beberapa teknik dan perangkat yang dapat mendukungnya dalam pengembangan sistem informasi.

Beberapa teknik ada yang hanya diterapkan pada satu tahapan saja pada siklus pengembangan sistem informasi, tetapi mungkin juga diterapkan pada seluruh siklus pengembangan sistem informasi. Salah satu teknik pengembangan sistem informasi yang

paling dikenal adalah pemrograman terstruktur. Pemrograman terstruktur ini diterapkan pada tahap penerapan dan pemeliharaan sistem informasi, tetapi tidak dapat diterapkan pada tahapan perencanaan atau analisis sistem informasi, oleh karenanya perlu digabungkan dengan teknik pengembangan sistem informasi lainnya yang mendukung.

Sebelum menginjak pada metode-metode pengembangan sistem informasi yang sering dipergunakan berikut ini akan diuraikan teknik-teknik yang sering digunakan dalam proses pengembangan sistem informasi akuntansi.

3.4.1 Teknik Terstruktur

Teknik terstruktur (*Structured techniques*) merupakan pendekatan formal untuk memecahkan masalah-masalah dalam aktivitas bisnis menjadi bagian-bagian kecil yang dapat diatur dan berhubungan. Bagian-bagian tersebut kemudian harus dapat disusun kembali menjadi suatu yang memberikan manfaat bagi pemecahan masalah-masalah bisnis. Teknik terstruktur ini dikenal juga dengan nama metode terstruktur. Dalam hubungannya dengan pengembangan sistem informasi dan *software* aplikasi sistem informasi, teknik terstruktur ini terbagi menjadi: pemrograman terstruktur (*structured programming*), desain terstruktur (*structured design*), analisis terstruktur modern (*modern structured analysis*), pemodelan *data* (*data modeling*) dan rekayasa informasi (*information engineering*).

Penekanan dalam teknik terstruktur ini pada umumnya difokuskan kepada aktivitas dan *data* sebagai dua hal penting dalam sistem informasi. Sudut pandang aktivitas atau proses bekerja berdasarkan konsep *input*, proses dan *output* (IPO), teknik terstruktur ini disebut juga sebagai teknik berorientasi kepada proses (*process oriented techniques*). Dalam membangun sistem informasi, teknik ini bekerja berdasarkan kepada pemahaman yang diperoleh dari hasil mempelajari proses dan/atau *input* serta *output* dari proses yang sedang berjalan. Contoh dari teknik ini adalah pemrograman terstruktur, desain terstruktur dan analisis terstruktur modern. Pemrograman terstruktur adalah teknik yang dipergunakan untuk merancang dan menyusun program secara jelas dan konsisten. Sedangkan desain (rancangan) terstruktur adalah teknik dan serangkaian pedoman yang digunakan untuk merancang hirarki dari modul-modul secara logis yang mewakili bahasa program komputer agar mudah diterapkan dan dipelihara. Analisis terstruktur modern merupakan teknik yang dapat menterjemahkan permintaan *user* dari sebuah sistem informasi dalam bentuk gambar-gambar yang mewakili fungsi, aktivitas, *input*, *output* dan penyimpanan *data* pada sebuah sistem informasi. Analisis terstruktur modern ini merupakan pengembangan dari analisis terstruktur yang sudah ada sebelumnya.

Dari sudut pandang *data*, teknik ini membangun model sistem informasi berdasarkan kepada organisasi yang ideal serta akses

Teknik terstruktur adalah pendekatan formal untuk memecahkan masalah-masalah dalam aktivitas bisnis menjadi bagian-bagian kecil yang dapat diatur dan berhubungan untuk kemudian dapat disatukan kembali menjadi satu kesatuan yang dapat dipergunakan untuk memecahkan masalah bisnis berbasis komputer.

IPO merupakan teknik terstruktur yang berorientasi kepada proses

yang dilakukan oleh organisasi tersebut terhadap *data* didalam sistem informasi yang digunakan untuk memenuhi kebutuhan informasi para pemakai (manajemen). Karena itu teknik ini disebut juga sebagai teknik berorientasi *data* (*data oriented techniques*). Contoh dari teknik ini adalah pemodelan *data* dan rekayasa informasi. Model *data* mewakili permintaan-permintaan *user* atas informasi dari sebuah sistem informasi, secara independen untuk mengetahui bagaimana *data* itu diproses atau bagaimana *data* itu menghasilkan informasi, sedangkan teknik rekayasa informasi menerapkan teknik terstruktur (baik yang berorientasi *data* maupun proses) untuk organisasi secara keseluruhan. Dari kedua contoh teknik terstruktur (proses dan *data*) keduanya mengakui bahwa proses memerlukan *data* dan *data* memerlukan proses.

Pemrograman Terstruktur

Pemrograman terstruktur adalah proses yang berorientasi kepada teknik yang digunakan untuk merancang dan menulis program secara jelas dan konsisten.

Secara tidak tertulis pemrograman terstruktur ini telah dijadikan standar dalam industri komputer. Pemrograman terstruktur (*structured programming*) ini merupakan proses yang berorientasi kepada teknik yang digunakan untuk merancang dan menulis program secara jelas dan konsisten. Yang terpenting dalam pemrograman terstruktur adalah bahwa logika dari setiap program dapat dan harus ditulis dengan satu perangkat struktur yang terkendali.

Teknik pemrograman terstruktur ini berhubungan dengan logika dan kode-kode program. Dengan teknik ini program harus dirancang sedemikian rupa sehingga dapat dibaca dari atas ke bawah dengan sedikit pencabangan. Pemrograman terstruktur yang baik biasanya ditulis dengan kombinasi dari tiga struktur pengendalian, kombinasi yang sering ditemui adalah :

- Urutan instruksi atau group instruksi
- Pemilihan instruksi-instruksi atau group instruksi berdasarkan kepada persyaratan tertentu (*if-then-else*)
- Pengulangan instruksi atau group instruksi berdasarkan beberapa persyaratan tertentu (*repeat-until dan do-while*)

Karakteristik utama dari struktur ini adalah bahwa setiap bentuk program harus mewakili *single entry* dan *single exit*. Artinya bahwa setiap program hanya memiliki satu pintu masuk dan satu pintu keluar. Program terstruktur dibaca dari atas ke bawah tanpa referensi ke langkah sebelumnya. Prosedur ini membuat kode mudah dibaca, diuji, dipahami dan dipelihara. Logika rancangan program terstruktur menggunakan model seperti *flowchart*, *box chart*, *pseudocode* atau diagram-diagram yang menggambarkan tindakan yang dilakukan.

Seperti juga teknik terstruktur, pemrograman terstruktur juga digunakan untuk mendukung tahap perancangan, implementasi dan pemeliharaan dalam siklus sebuah sistem informasi.

Rancangan (Desain) Terstruktur

Teknik rancangan terstruktur membantu pengembang sistem informasi dalam menentukan ukuran dan kompleksitas dari suatu program. Rancangan terstruktur (*structured design*) adalah salah satu proses yang berorientasi teknik yang digunakan untuk memilah-milah program besar ke dalam hirarki modul-modul yang menghasilkan program komputer yang lebih kecil agar mudah untuk diimplementasikan dan dipelihara (atau dirubah).

Konsep rancangan terstruktur sangat sederhana. Program dirancang berdasarkan model hirarki dari atas ke bawah. Model ini merupakan kumpulan instruksi seperti paragraf, subprogram atau bagian kegiatan rutin. Model terstruktur dari atas ke bawah ini (*top-down modules*) dikembangkan sesuai dengan aturan dan petunjuk yang ada.

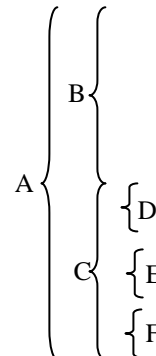
Idealnya setiap logika program dalam suatu modul ditulis dengan menggunakan teknik pemrograman secara terstruktur, sehingga kita melihat bahwa teknik-teknik tersebut dapat digunakan juga untuk memperbaiki cara pemecahan masalah. Pandangan lain telah membangun suatu teknik yang cukup memadai untuk melaksanakan perancangan terstruktur dengan baik. Teknik-teknik ini meliputi:

- **Yourdon-Constantine** - Teknik ini mengembangkan struktur *software* ideal dengan mempelajari arus *data* yang sesuai dengan fungsi program.
- **Warnier-orr** - Teknik ini mengembangkan struktur *software* dengan mempelajari isi dari *input* dan *output*.
- **Jackson** - Teknik ini juga mengembangkan struktur *software* dengan mempelajari isi dari *input* dan *output*.

Semua teknik ini berorientasi ke proses, sesuai dengan tujuannya yaitu untuk merancang proses, terutama proses didalam *software*. Rancangan struktur *Yourdon* merupakan teknik yang paling dikenal luas, rancangan ini memecah sebuah program kedalam modul-modul yang tersusun secara bertingkat dari atas ke bawah. Modul ini memiliki susunan sebagai berikut:

- **Modul harus benar-benar menyatu**; ini berarti bahwa setiap modul harus dapat melaksanakan satu dan hanya satu fungsi dengan demikian modul ini dapat digunakan lagi untuk program masa yang akan datang.
- **Modul harus bebas**, dengan kata lain modul ini tidak boleh tergantung satu sama lain. Kemandirian modul ini akan mengakibatkan setiap perubahan yang terjadi pada satu modul dimasa yang akan datang hanya sedikit pengaruhnya pada modul yang lain.

Rancangan terstruktur membantu pengembang sistem informasi dalam menentukan ukuran dan kompleksitas dari program.



Model *software* yang dikembangkan oleh *Yourdon* ini dikenal dengan nama diagram terstruktur (*structured chart*) yang dihasilkan dengan mempelajari arus *data* dari suatu program. Perancangan terstruktur ini digunakan pada tahap perancangan pada siklus pengembangan sistem. Ada beberapa kelebihan dari rancangan terstruktur ini:

Keuntungan model terstruktur:

- Program yang disusun untuk rancangan terstruktur dapat dengan mudah ditulis dan diuji oleh tim programmer.
- Mudah dipelihara
- Model program dapat digunakan secara berulang.

- Program yang disusun untuk rancangan terstruktur dapat dengan mudah ditulis dan diuji oleh tim programmer. Hal ini dimungkinkan karena pengaruh antar model telah ditentukan dengan baik dan dibatasi oleh aturan-aturan tertentu. Model yang telah diuji dengan baik akan dapat diuji dengan baik pula bila digabungkan dengan model lainnya dalam sebuah sistem informasi. Struktur pemrograman dari atas ke bawah (*top-down*) juga menyederhanakan upaya pemrograman melalui pengkodean dari atas ke bawah dan melakukan pengujian program.
- Sistem dan program yang dirancang serta dikembangkan secara terstruktur akan lebih mudah untuk dipelihara.
- Keuntungan dari perancangan secara terstruktur adalah model program yang dikembangkan dapat digunakan secara berulang-ulang.

Analisis Terstruktur

Analisis terstruktur adalah teknik yang berorientasi kepada proses yang berorientasi kepada proses yang terpusat dan digunakan untuk membentuk model permintaan *user* terhadap sistem informasi.

Analisis terstruktur merupakan teknik yang berorientasi kepada proses yang paling populer dan banyak digunakan dewasa ini. Analisis terstruktur adalah teknik yang berorientasi dan terpusat pada proses. Teknik ini digunakan untuk membuat suatu model berdasarkan permintaan *user* terhadap sistem informasi. Analisis terstruktur memilah-milah sistem kedalam berbagai proses, *input*, *output* dan *file-file*. Teknik ini menyusun arus *input-proses-output* dari berbagai masalah bisnis serta solusinya.

Teknik analisis secara terstruktur konsepnya sangat sederhana. Model sistem informasi baru dikembangkan dari serangkaian diagram arus yang disebut *data flow diagram (DFD)* atau diagram arus *data*. DFD menunjukkan arus *data*, penyimpanan *data*, dan proses yang merespon *data* yang masuk dan merubahnya. Dalam analisis secara terstruktur, analisis sistem mungkin menghasilkan sejumlah DFD model SIM. Setiap DFD dibedakan berdasarkan tujuannya:

- Apakah model sistem informasi manajemen yang dikembangkan untuk menggambarkan sistem informasi berjalan atau yang seharusnya.
- Apakah model sistem informasi yang dikembangkan menggambarkan sistem informasi secara detail atau pokok-pokoknya saja.

Oleh karena itu model DFD bisanya dibuat berdasarkan: (1) sistem informasi berjalan atau yang diimplementasikan saat ini ; (2) pokok-pokok penting secara konseptual dari sistem informasi yang sedang berjalan; (3) pokok-pokok penting secara konseptual dari sistem informasi yang diusulkan dan (4) implementasi sistem informasi yang diusulkan.

Model konsep sistem secara logis sering juga disebut sebagai model pokok atau model inti dari sistem (*essential system*) yang dibuat ketika melakukan dan merupakan hal penting dalam analisis terstruktur. Model tersebut diatas kalau dibuat sekaligus secara bersamaan memiliki beberapa masalah yang diantaranya adalah:

- Membatasi kreativitas karena sejak awal sudah memikirkan sistem baru dalam bentuk bagaimana sistem tersebut dide-sain dan diimplementasikan.
- Mendorong analisis sistem untuk menentukan terlebih dahulu apa yang harus dilakukan (Konseptual sistem) oleh suatu sistem sebelum sistem informasi tersebut dirancang dan diterapkan.

Oleh karenanya teknik ini memaksa analisis sistem untuk lebih dahulu menentukan bagaimana memecahkan masalah bisnis sebelum menentukan solusi tekniknya. Karena itu, para pakar menyatakan harus ada pemisahan antara sistem yang seharusnya dan sistem berjalan agar memberikan beberapa keuntungan antara lain :

- Analisis sistem akan lebih akurat dalam mengidentifikasi permintaan/kebutuhan bisnis dan pemakai sistem tanpa mengkhawatirkan teknologi yang digunakan.
- Analisis sistem akan lebih memiliki kesempatan mengembangkan kreativitasnya dalam menentukan alternatif pemecahan masalah berdasarkan pada sistem yang sedang berjalan.

Pada pendekatan Gane-Sarson dan pendekatan DeMarco, analisis sistem menggambarkan empat set diagram arus *data* yaitu:

1. Diagram arus *data* sistem informasi yang sedang berjalan (menggambarkan bagaimana sistem informasi tersebut saat ini bekerja).
2. Diagram arus *data* secara konseptual dari sistem yang sedang berjalan (merupakan pengembangan dari diagram arus *data* pada nomor 1, tetapi menggambarkan apa yang dilakukan oleh sistem yang sedang berjalan).
3. Diagram arus *data* secara konseptual untuk sistem baru (penambahan, penghapusan, dan modifikasi dari arus *data* pada nomor 2, memperlihatkan apa yang harus dilakukan sistem baru).

4. Diagram arus *data* untuk sistem yang akan disusun, (memperlihatkan bagaimana sistem baru mengimplementasikan permintaan/kebutuhan).

Selanjutnya *Ed Yourdon* mengembangkan versi baru dari analisis terstruktur yang disebut dengan analisis terstruktur modern. Pendekatan ini meringkas rincian model dari sistem yang sedang berjalan baik secara logika maupun fisik, kecuali untuk pembuatan:

- Diagram arus *data* yang sangat sederhana dari sistem yang diusulkan
- Model *data* secara konseptual (logis)
- Diagram arus *data* secara konseptual dari atas ke bawah untuk sistem baru (yang menunjukkan bagaimana sistem seharusnya bekerja)
- Diagram arus *data* yang menunjukkan bagaimana konseptual sistem diimplementasikan.

Setiap langkah ditujukan untuk kepentingan sistem baru, sedikit sekali perhatian diberikan untuk sistem yang sedang berjalan. Variasi analisis terstruktur merupakan gabungan dari pendekatan analisis terstruktur awal dan pendekatan modern. Para ahli kini menyarankan untuk menggunakan analisis terstruktur bagi sistem yang sedang berjalan. Alasannya adalah (1) seringkali waktu yang dibutuhkan tidak disesuaikan dengan biaya yang tersedia; (2) sistem yang sedang berjalan biasanya bias dan kurang bisa mengadaptasi perubahan dari sistem lama ke sistem baru dan (3) pemakai sistem dan manajer sering tidak sabar dengan model sistem yang membutuhkan banyak perubahan dan unsur-unsur yang diganti.

Analisis terstruktur merupakan teknik terstruktur pertama yang digunakan untuk menganalisis siklus pengembangan sistem informasi, sedangkan untuk tahap-tahap lainnya dapat menggunakan teknik yang lain lagi. Analisis terstruktur dan rancangan terstruktur merupakan teknik yang terintegrasi. Rancangan terstruktur yang disusun oleh *Yourdon* menyediakan strategi untuk mengembangkan struktur program, dari diagram arus *data* pada analisis terstruktur, yang biasa disebut juga sebagai teknik rekayasa *software* (*software engineering technique*).

Pemodelan Data terstruktur

Saat ini proses yang berorientasi kepada teknik dilengkapi dengan teknik yang berorientasi kepada *data*. Pemodelan *data* adalah suatu teknik yang berorientasi kepada *data* dengan menunjukkan sistem hanya *datanya* saja terlepas dari bagaimana *data* tersebut akan diproses atau digunakan untuk menghasilkan informasi.

Seperti teknik yang lainnya, pemodelan *data* pun konsepnya sangat sederhana. Jika *data* dikumpulkan dan disimpan dalam *file* dengan struktur *database* yang fleksibel, semua informasi yang dibutuhkan baik saat ini atau pun dimasa mendatang dapat dipenuhi melalui pemodelan *data* ini sepanjang informasi yang diperlukan dimasa mendatang diketahui. Jadi pemodelan *data* digunakan untuk merancang *database*. Teknik pemodelan *data* secara ringkas menjelaskan hal-hal sebagai berikut: Pertama, menentukan hal-hal yang berkaitan dengan bisnis (entitas), tentang bagaimana bisnis atau aplikasinya mengumpulkan *data*. Entitas mungkin meliputi hal-hal sebagai berikut:

- Barang-barang nyata seperti, bahan-bahan, peralatan, mesin, kendaraan dan produk.
- Pelaksana bisnis, seperti pelanggan, pemasok, pegawai dan pemegang kredit.
- Peristiwa-peristiwa, seperti pesanan, permintaan, kontrak, perjalanan, kecelakaan atau pembayaran.
- Tempat, seperti kantor penjualan dan gudang.

Selanjutnya menentukan atribut yang menerangkan masing-masing entitas. Kemudian, menentukan aktivitas (relasi) bisnis di antara entitas. Pada beberapa teknik atau pendekatan pemodelan *data*, analisis harus menentukan aktivitas bisnis yang menyebabkan harus dibuatnya, dihapusnya dan dimodifikasinya suatu *data* dalam entitas. Pada pemodelan *data* yang lain, analisis juga biasanya menentukan aktivitas bisnis yang secara langsung akan mempengaruhi suatu entitas. Manfaat yang terpenting dari pemodelan *data* adalah apabila *file-file* atau *database* untuk sistem baru disusun sesuai dengan pemodelan *data* maka akan memenuhi hal-hal sebagai berikut:

- *Database* akan terdiri dari *data* yang akurat dan terbaru
- *Database* akan memenuhi semua kebutuhan informasi saat ini
- *Database* akan mampu memenuhi kebutuhan informasi masa yang akan datang tanpa ada perubahan yang drastis pada sistem informasi yang ada, karena *data* yang dibutuhkan sudah tersedia atau dapat dengan mudah ditambahkan pada entitas yang sudah ada.

Dengan kata lain, proses dan aplikasi yang diterapkan dapat dipergunakan untuk mengembangkan model *data* dengan sedikit perubahan pada model *data* itu sendiri. Teknik pemodelan *data* ini cocok dipergunakan untuk *file-file* model lama dan *data* yang didistribusikan, juga untuk menyusun aplikasi *database* kecil yang mudah diintegrasikan.

Walaupun model *data* ini sangat berguna, tetapi proses dari sistem informasi tetap saja harus dirancang. Teknik dan perangkat yang berorientasi proses seperti analisis terstruktur modern

Pemodelan *data* adalah teknik yang berorientasi kepada *data* yang menunjukkan sistem hanya *datanya* saja dari sistem tersebut terlepas dari bagaimana *data* tersebut diproses atau digunakan untuk menghasilkan informasi.

merupakan salah satu teknik yang dapat digunakan untuk perancangan proses tersebut di atas dan melengkapi teknik pemodelan *data*. Gabungan dari teknik model *data* dan proses ini dapat dipergunakan untuk mengembangkan teknik lain yang disebut rekayasa informasi (*information engineering*).

Rekayasa Informasi

Rekayasa informasi merupakan perpaduan dari pemodelan *data* dan proses.

Rekayasa informasi disamping merupakan perpaduan dari pemodelan *data* dan proses, rekayasa informasi juga memberikan penekanan baru terhadap pentingnya perencanaan sistem informasi. Rekayasa informasi telah menjadikan analisis dan perancangan terstruktur sebagai teknis paling populer pada praktek saat ini. (Sebagai catatan, rekayasa informasi sebenarnya meliputi juga konsep, alat, dan teknik dalam analisis terstruktur).

Rekayasa informasi merupakan rekayasa yang lebih berorientasi kepada *data*, dan proses yang dilakukan sangat dipengaruhi oleh teknik yang digunakan oleh organisasi perusahaan secara keseluruhan. Walaupun teknik ini menyarankan adanya keseimbangan antara metode yang berorientasi kepada *data* dan proses, akan tetapi ini sangat jelas bahwa teknik ini sangat dikenal oleh *data*, sehingga dalam pengembangan sistem informasi, model *data* dibuat terlebih dahulu baru kemudian model proses. Rekayasa informasi merupakan teknik terstruktur pertama yang dapat digunakan diseluruh siklus pengembangan sistem informasi, kecuali pada fase pemeliharaan. Teknik ini digunakan dalam perancangan sistem informasi dan menempatkannya sebagai alat untuk meningkatkan kualitas sistem informasi dalam organisasi. Langkah-langkah dalam rekayasa informasi adalah:

Langkah-langkah dalam rekayasa informasi :

- Strategi perencanaan sistem informasi
- Menentukan subsistem (area bisnis)
- Menentukan sub sistem aplikasi prioritas
- Implementasi aplikasi hasil perancangan

1. Analisis terlebih dahulu akan menggunakan strategi perencanaan sistem informasi untuk organisasi,
2. Berdasarkan strategi tersebut analisis menentukan subsistem yang di dalam rekayasa informasi disebut sebagai area bisnis. Analisis sistem informasi terstruktur ini kemudian diterapkan di seluruh area bisnis.
3. Analisis setelah itu kemudian menentukan subsistem (area bisnis) yang lain yang paling membutuhkan aplikasi untuk kemudian dianalisis dan dirancang aplikasinya.
4. Analisis kemudian mengimplementasikan aplikasi hasil perancangan tersebut.

Selanjutnya proyek ini ditujukan kepada aplikasi yang lain dalam area bisnis yang sama, sampai semua area bisnis yang ada pada suatu saat membentuk satu aplikasi yang menyatu. Selanjutnya aplikasi ini diintegrasikan dengan *database* yang ada di dalam area bisnis. Area bisnis yang lainnya akan mengikuti proses ini secara bergantian mulai dari analisis, perancangan dan implementasi.

Pada rekayasa informasi, pusat dari semuanya adalah penyimpanan *data*. Analis sistem, programer dan ahli komputer yang lainnya bertanggungjawab dalam merancang seluruh pengumpulan dan penyimpanan *data* serta menjamin bahwa *data* tersebut benar-benar telah dimasukkan, disimpan dan terpelihara dengan baik. Para profesional rekayasa informasi juga merancang dan mengimplementasikan *output* berupa laporan-laporan penting dari sistem informasi yang dibuat. Pemakai sistem informasi (akuntansi) akan menambahkan informasi yang dibutuhkan setelah mempelajari dan menggunakan laporan-laporan yang dihasilkan yang berasal dari berbagai macam kebutuhan pemakai sistem. Rekayasa informasi pada dasarnya merupakan revisi dari teknik analisis terstruktur klasik, tetapi walaupun begitu teknik ini bukan merupakan teknik yang mutlak menggantikan teknik lama, karena teknik ini dapat berjalan seimbang dan berintegrasi dengan teknik-teknik yang berorientasi kepada *data* dan proses.

Dengan rekayasa informasi, maka sekaligus dapat dipelajari pendekatan-pendekatan analisis terstruktur, perancangan terstruktur, pemodelan *data* dan pendekatan penting lainnya digunakan.

Teknik Joint Application Development JAD

Teknik lain yang berkembang dikalangan para ahli sistem informasi akuntansi adalah *Joint Application Design* atau yang bila dihubungkan dengan pengembangan sistem informasi lebih dikenal dengan nama *Joint Application Development (JAD)*. Teknik pada umumnya berhubungan dengan *data* dan prosesnya, tetapi dalam kaitannya dengan pengembangan sistem informasi informasi, teknik JAD adalah suatu teknik baru yang berhubungan dengan manusia.

Joint Application Development (JAD) adalah suatu kerja sama yang terstruktur antara pemakai sistem informasi (*users*), manajer dan ahli sistem informasi untuk menentukan dan menjabarkan permintaan pemakai, teknik-teknik yang dibutuhkan dan unsur rancangan eksternal (*input, output* dan tampilan).

Tujuan dari JAD adalah memberi kesempatan kepada *user* dan manajemen untuk berpartisipasi secara luas dalam siklus pengembangan sistem informasi. Luasnya partisipasi yang diberikan kepada *user* dan manajemen ini memberikan beberapa manfaat yaitu:

- Meningkatkan hubungan antara *user*, manajemen dan ahli sistem informasi informasi,
- Memperluas wawasan *user* dan manajemen dalam bidang komputer, disisi lain memperluas wawasan bisnis dan aplikasinya bagi ahli sistem informasi,
- Meringankan beban tanggung jawab *user* dan manajemen bila terjadi konflik,

Tujuan dari JAD adalah memberi kesempatan kepada *user* dan manajemen untuk berpartisipasi secara luas dalam siklus pengembangan sistem informasi.

- JAD umumnya juga mempersingkat waktu pengembangan sistem informasi yang biasanya diperlukan untuk melakukan berbagai wawancara, melalui satu pola kerja yang lebih terstruktur,
- Melalui penentuan keinginan *user* yang lebih tepat dan penentuan prioritas utama, maka penggunaan JAD ini akan lebih menghemat biaya,
- JAD seringkali menghasilkan sistem informasi yang lebih bernilai dan memberikan kepuasan yang lebih baik bagi *user* maupun pihak manajemen, sehingga meningkatkan kepercayaan dan dukungan *user* dan manajemen terhadap proyek pengembangan sistem informasi yang dilakukan.
- Mengurangi biaya pemeliharaan, karena sejak versi pertama dihasilkan, telah mampu memenuhi kebutuhan organisasi umumnya.

Hampir semua teknik JAD dijadwalkan untuk bekerja cepat. Dengan bimbingan analis sistem informasi yang profesional dalam bidang JAD, kelompok kerja yang dibentuk akan dengan cepat menentukan kunci masalah, kebutuhan, prioritas dan alternatif pemecahan masalah dan memilih pemecahan masalah yang tepat. Analis JAD yang berpengalaman dapat juga menggunakan teknik yang terstruktur bersama-sama dengan *user*, dan secara hati-hati menghindari teknik dan peraturan yang berbelit-belit. Dalam waktu singkat kelompok kerja JAD dapat menggantikan satu sampai enam bulan jadwal wawancara dan mengurangi jadwal pertemuan-pertemuan yang umum dipergunakan dalam suatu siklus pengembangan sistem informasi. Dalam teknik ini kecil sekali kemungkinan munculnya konflik atau perbedaan, karena setiap anggota kelompok kerja sejak awal telah menyetujui hal apa yang paling penting dibentuk dalam penyusunan sebuah sistem informasi akuntansi.

Keberhasilan teknik JAD akan mudah dicapai apabila pihak manajemen dapat memenuhi beberapa ketentuan berikut: Pertama, pihak manajemen harus memberi ijin penuh kepada para pekerjanya agar dapat terlibat secara penuh dalam setiap sesi atau tahapan pengembangan sistem informasi. Kedua, manajemen harus juga turut terlibat secara langsung melalui penciptaan kerjasama dan bersedia turut mendengarkan ketika bekerja dengan bawahannya selama sesi pengembangan sistem informasi akuntansi ini berjalan. Ketiga, pembuat dokumentasi dan pimpinan harus betul-betul terlatih untuk memberikan perhatian penuh dan mengarahkan diskusi serta menjadi penengah bila terjadi konflik dan perdebatan diantara anggota kelompok kerja.

3.5 Keterlibatan User dalam Membangun Sistem informasi

Setiap metode dan teknik pengembangan sistem informasi manajemen yang diuraikan di muka selalu menuntut adanya peranan *user* dalam setiap tahap, perancangan dan pengembangan sistem informasi. Seberapa besar pengaruh keterlibatan *user* pada perancangan dan pengembangan sistem informasi akuntansi terhadap sistem informasi yang akan diterapkan kelak akan diuraikan berikut ini.

Efektivitas dari setiap aplikasi komputer dipengaruhi oleh keterlibatan *user* dalam proses perancangan dan pengembangan sistem informasi akuntansi dan oleh kualitas dukungan yang diberikan *user*. Keterlibatan *user* (*user involment*) dalam perancangan dan pengembangan sistem informasi lebih ditekankan pada bagaimana peranan *user* dalam proses perancangan sistem informasi dan langkah-langkah apa yang dilakukan dalam mendukung dan mengarahkan kontribusinya, sedangkan yang dimaksud dengan dukungan *user* (*user support*) terhadap perancangan dan pengembangan sistem informasi akuntansi berhubungan dengan pengarahan yang diberikan oleh *user* pada saat sistem informasi dioperasikan, salah satunya adalah dengan menggunakan komputer secara efektif.

Beberapa alasan pentingnya keterlibatan *user* dalam perancangan dan pengembangan sistem informasi menurut Leela Damodaran (1983) adalah:

- **Kebutuhan *user*** - *user* adalah orang dalam perusahaan. Analis sistem atau ahli sistem adalah orang diluar perusahaan. Sistem informasi dikembangkan bukan untuk pembuat sistem tapi untuk *user* agar sistem bisa diterapkan, sistem tersebut harus bisa menyerap kebutuhan *user* dan yang tahu kebutuhan *user* adalah *user* sendiri, sehingga keterlibatan *user* dalam pengembangan sistem akan meningkatkan tingkat keberhasilan walaupun tidak memberikan jaminan berhasil.
- **Pengetahuan akan kondisi lokal** - Pemahaman terhadap lingkungan dimana sistem informasi akuntansi akan diterapkan perlu dimiliki oleh perancang sistem informasi, dan untuk memperoleh pengetahuan tersebut perancang sistem harus meminta bantuan *user* yang sangat memahami lingkungan tempatnya bekerja.
- **Keengganan untuk berubah** - Seringkali *user* merasa bahwa sistem informasi yang disusun tidak dapat dipergunakan dan tidak sesuai dengan kebutuhan. Untuk mengurangi keengganan untuk berubah itu dapat dikurangi bila *user* terlibat dalam proses perancangan dan pengembangan sistem informasi.

- **User merasa terancam** - Banyak *user* menyadari bahwa penerapan sistem informasi komputer dalam organisasi mungkin saja mengancam pekerjaannya, atau menjadikan kemampuan yang dimilikinya tidak lagi relevan dengan kebutuhan organisasi. Keterlibatan *user* dalam proses perancangan dan pengembangan sistem informasi merupakan salah satu cara menghindari kondisi yang tidak diharapkan dari dampak penerapan sistem informasi akuntansi dengan komputer.
- **Meningkatkan alam demokrasi** - Makna dari demokrasi di sini adalah bahwa *user* dapat terlibat secara langsung dalam mengambil keputusan yang akan berdampak terhadap mereka. Penerapan sistem informasi berbasis komputer tentunya akan berdampak kepada para pegawai, oleh karenanya diperlukan keterlibatan *user* secara langsung dalam proses perancangan sistem informasi akuntansi ini.

Tidak semua keterlibatan *user* ini membawa keberhasilan, ada beberapa alasan yang menyebabkan terjadinya kegagalan diantaranya:

- Tidak tepatnya pengetahuan yang dimiliki *user* sehingga tidak bersedia membuat keputusan atau memberikan pandangannya, karena *user* kurang memahami dampak dari keputusan yang diambilnya.
- Kurangnya pengalaman dalam menentukan keputusan karena kultur lingkungan yang tidak mendukung dan kurangnya dukungan dari organisasi dalam berpartisipasi untuk mengambil keputusan.
- Pengambilan keputusan terbatas pada tahapan-tahapan yang memungkinkan *user* atau karyawan terlibat dalam pengambilan keputusan.
- Kurangnya kesempatan untuk melakukan uji coba dan kurangnya kesempatan untuk belajar, hal ini muncul karena ketakutan akan tingginya biaya yang perlu dikeluarkan untuk kegiatan tersebut.

Agar keterlibatan *user* dalam perancangan dan pengembangan sistem informasi menjadi efektif perlu persiapan dan perencanaan dalam penyusunan struktur organisasi dan satu prosedur yang mendukung proses pengembangan sistem informasi akuntansi. Dukungan *user* harus dimulai dari awal proses. Keterlibatannya dalam perancangan dan pengembangan sistem informasi akan terus berlanjut pada setiap tahap siklus pengembangan sistem informasi. Langkah-langkah dukungan *user* ini biasanya disusun dalam satu perencanaan yang terintegrasi dengan sistem informasi. Agar dukungan *user* ini menjadi efektif maka perencanaan dan perancangan kerangka kerja dari dukungan *user* harus disusun secara hati-hati. Kriteria-kriteria yang harus diperhatikan adalah:

- Mempromosikan komunikasi dua arah
- Menyediakan jaringan kerja yang terintegrasi dalam mekanisme dukungan.
- Mengenali kemajemukan *user*
- Memiliki kapabilitas yang dinamis
- Mudah menangani keinginan *user*
- Mudah mengenali kebutuhan *user*
- Tersedianya sumber daya yang memadai seperti keuangan, waktu, usaha dan tenaga ahli.

Dukungan terhadap *user* pada dasarnya bisa dibagi menjadi dua bagian yaitu dukungan terhadap sistem informasi seperti pelatihan instruktur dan dukungan kepada tenaga lokal atau *user* lainnya dalam organisasi.

Rangkuman

Seringkali terjadi kesalahpahaman dalam memahami pengertian teknik dan metodologi pengembangan sistem informasi dengan siklus sistem informasi. Metodologi adalah gabungan dan rincian dari langkah demi langkah dan tugas masing-masing langkah tersebut, aturan individu dan kelompok yang harus menjalankan setiap tugas tersebut; standar kualitas dan alur dari setiap tugas; dan teknik-teknik pengembangan yang digunakan untuk setiap tugas yang dilakukan. Teknik adalah pendekatan, alat yang digunakan dan peraturan-peraturan yang melengkapi satu atau lebih tahap-tahapan dalam siklus pengembangan sistem informasi.

Sedangkan yang dimaksud dengan siklus (*life cycle*) adalah tahap-tahapan dan tugas-tugas yang sangat penting untuk mengembangkan sistem informasi, tanpa memperhatikan apa jenis sistem informasi yang akan dibuat dan seberapa luas sistem informasi itu nantinya. Dari definisi-definisi tersebut terlihat bahwa metodologi dan teknik ada untuk melengkapi siklus pengembangan sistem informasi.

Teknik pengembangan sistem informasi akuntansi yang sering digunakan adalah teknik terstruktur, yang memiliki karakteristik berorientasi pada proses dan *data*. Teknik yang berorientasi pada proses menyusun model sistem informasi berdasarkan pada pemahaman tentang arus *input* yang diproses menjadi *output*. Teknik yang sering digunakan dari teknik yang berorientasi pada proses ini adalah pemrograman terstruktur, desain terstruktur dan analisis terstruktur modern.

System Development Life Cycle (SDLC) adalah salah satu metode yang digunakan untuk proses pengembangan sistem informasi. Metode ini merupakan metode pertama yang berkembang dan sering digunakan.

Ketika metode SDLC, dianggap tidak lagi memenuhi kebutuhan analisis sistem dalam proses pengembangan sistem informasi maka dikembangkan metode baru yang bernama metode *prototyping* yang merupakan teknik yang membuat model kerja (prototipe) dari sistem informasi atau subsistem informasi yang akan diterapkan dalam skala kecil (dan melakukan simulasi).

Ada beberapa metode dan teknik lain yang berkembang dalam pengembangan sistem informasi akuntansi yaitu *Joint Application Development (JAD)*, *Rapid Application Development* dan *Softsystem*. *Joint Application Development (JAD)*, adalah suatu kerja sama yang terstruktur antara pemakai sistem informasi (*users*), manajer dan ahli sistem informasi untuk menentukan dan menjabarkan permintaan pemakai, teknik-teknik yang dibutuhkan dan unsur rancangan eksternal (*input, output* dan tampilan). *Rapid Application Development (RAD)* adalah penggabungan beberapa metode dan teknik terstruktur (khususnya dalam perancangan *data* untuk menghasilkan informasi). Metode lainnya adalah metode *softsystem* yang memiliki tujuh tahapan proses untuk menangani masalah-masalah dalam kehidupan (aktivitas) sehari-hari yang berdampak pada organisasi.

Soal

1. Apa beda siklus, metodologi dan teknik ?
2. Apa yang dimaksud dengan teknik terstruktur ? Beri contoh.
3. Apa yang dimaksud dengan rekayasa informasi ?
4. Sebutkan empat metode pengembangan sistem informasi akuntansi?
5. Mengapa Metode SDLC disebut metode tradisional ?

Tugas

1. Sebutkan tahap-tahap dalam metode SDLC ?
2. Apakah yang menjadi masalah dalam metode SDLC sehingga metode tersebut dianggap gagal ?
3. Apa yang menjadi dasar pemikiran munculnya metode *prototyping*? dan sebutkan tahap-tahap dalam metode *prototyping*?
4. Kapan pengembangan sistem dengan metode *prototyping* berakhir ?
5. Mengapa teknik *join application* membantu meningkatkan keberhasilan pengembangan sistem informasi akuntansi?

4

Praktika Pembuatan Model Aplikasi sistem informasi akuntansi sederhana

Pendahuluan

Sistem Informasi dibangun untuk membantu manajemen dalam menyediakan informasi yang berkualitas untuk membantunya dalam proses pengambilan keputusan.

Menyajikan Informasi berkualitas kepada manajemen dapat dilakukan dengan pengelolaan data (Manajemen Data) yang berasal dari berbagai sumber untuk diolah dengan berbagai cara guna memenuhi berbagai kepentingan manajemen yang berada pada tingkatan dan fungsi organisasi yang berbeda.

Manajemen data atau manajemen informasi dalam arti luas dilakukan dengan cara mengelola sumber daya penghasil informasi (*Hardware, Software, Brainware, Prosedur, database, dan komunikasi*). Sedangkan manajemen data dalam arti sempit dilakukan dengan cara mengelola datanya saja dengan menggunakan sistem manajemen *database*, banyak *software* yang dapat digunakan untuk mengelola data baik untuk membuat aplikasi berbasis windows seperti Delfi dan visual foxpro atau aplikasi berbasis web seperti My SQL, Dalam skop aplikasi yang lebih besar penggunaan Oracle dan SQL Server dapat menjadi pilihan untuk mengelola data perusahaan yang besar (BIG DATA).

Visual FoxPro dipilih disini selain karena kemampuannya sudah teruji cukup lama, *Visual FoxPro* juga punya keunggulan dalam hal *Table* dan fasilitas jaringan (*Multi user*) yang sangat diperlukan untuk pembuatan program aplikasi bisnis, pertimbangan lainnya *Visual FoxPro* mudah migrasi ke *Internet* dan didukung oleh perusahaan *software* terbesar didunia, Walaupun visual Foxpro sudah tidak di produksi lagi oleh microsoft tapi *software* ini masih banyak digunakan di masyarakat untuk belajar sistem manajemen *database* dan aplikasi berbasis windows.

Kearena penggunaan komputer sebagian besar untuk bisnis, dan dalam dunia bisnis bagian yang paling banyak menggunakan komputer atau teknologi informasi ini bagian akuntansi yang merupakan bagian yang bertanggung jawab terhadap konten aplikasi akuntansi atau sistem informasi akuntansi dan bagian teknologi informasi yang bertanggung jawab dalam memberi jaminan kepada pengguna bahwa sistem informasi akuntansi atau sistem informasi manajemen yang digunakan akan selalu berfungsi setiap saat diperlukan.

Bagian ini akan menjelaskan bagaimana membuat aplikasi akuntansi yang sederhana dengan menggunakan visual foxpro, penguasaan bidang ini akan sangat melengkapi pengetahuan yang dimiliki oleh baik dosen akuntansi/sistem informasi akuntansi atau para mahasiswa jurusan akuntansi, manajemen dan teknologi akuntansi agar mendapatkan ilmu yang sedang mereka pelajari.

BS 4.1 Mengenal Microsoft Visual FoxPro

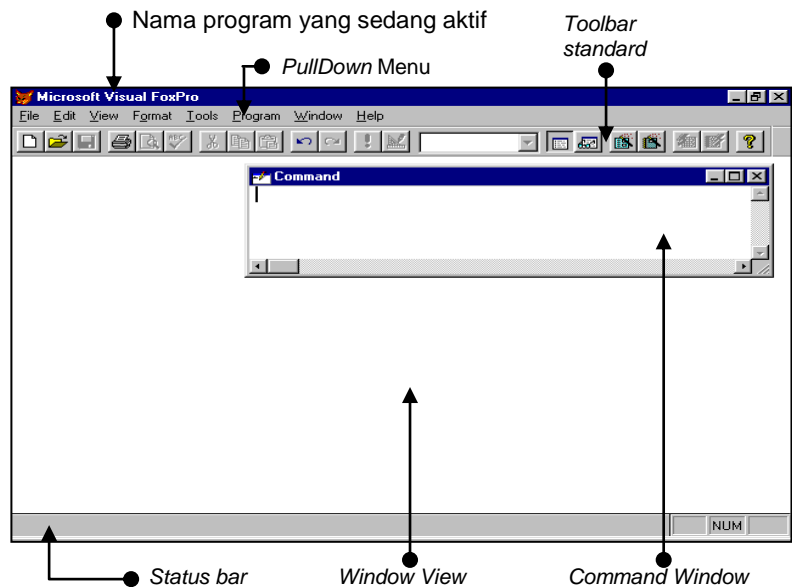
Microsoft Visual FoxPro adalah salah satu bahasa pemrograman pengembangan dari *FoxPro-FoxPro* generasi sebelumnya. *Microsoft Visual FoxPro* akan bekerja lebih optimal bila di *install* pada *Windows 95, 98* atau pada *Windows NT*.

Pada bagian ini akan dibahas beberapa komponen dasar yang perlu diketahui saat menggunakan *Microsoft Visual FoxPro* yang selanjutnya disebut sebagai *Visual FoxPro*.

P 4.1.1 Mengaktifkan Visual FoxPro

Jika anda sudah memiliki program aplikasi *Visual FoxPro* maka untuk mengaktifkannya anda tinggal mengklik menu bar *Visual FoxPro* atau *shortcutnya*. Setelah anda mengaktifkannya maka akan tampil *desktop Visual FoxPro* pada layar komputer anda seperti tampak pada gambar di bawah ini.

Gambar 4.1 Tampilan Visual FoxPro Desktop



4.1.2 Mengenal Menu Visual FoxPro

Menu utama

BS Menu yang muncul pada *desktop Visual FoxPro* diatas terdiri dari beberapa jenis, yaitu *PullDown* menu, *PopUp (dropdown)* menu dan *Icon-icon* yang tersimpan pada *Toolbar*.

Pada main menu terdapat beberapa submenu *pulldown* seperti :

- ❑ **File**, sub menu ini digunakan untuk membuat (*New*), membuka (*Open*), merekam (*Save*), dan menutup file (*Close*). Sub-menu *file* juga memberikan fasilitas untuk mencetak (*Print*) dan keluar dari *Visual FoxPro* (*Exit*).
- ❑ **Edit**, submenu ini digunakan untuk mengedit. Beberapa fasilitas yang ada diantaranya adalah fasilitas untuk mengcopy (*Copy*), memotong bagian objek (Teks atau gambar) yang telah ditandai (*Cut*), menempelkan atau meletakkan objek yang dicopy atau dipotong (*Paste*), dan lain-lain.
- ❑ **View**, submenu ini berisi berbagai fasilitas untuk menampilkan macam-macam *Toolbar* yang akan anda gunakan saat pembuatan program.
- ❑ **Format**, submenu ini berfungsi untuk mengatur format teks seperti huruf (*Font*), Spasi (*space*) dan lain-lain.
- ❑ **Tool**, submenu ini memberikan banyak fasilitas yang diperlukan oleh programmer mulai dari penuntun (*wizards*) sampai dengan pilihan (*options*)
- ❑ **Program**, submenu ini digunakan untuk menjalankan, menghentikan dan mengkompile program.
- ❑ **Windows**, submenu ini membantu programmer mulai dari menyusun semua judul (*Arrange all*) dari *windows* yang dibuka sampai dengan menayangkan (*Windows list*) sembilan *windows* pertama.
- ❑ **Help**, submenu ini digunakan untuk membuka sistem bantuan yang telah disediakan *Visual FoxPro*

BS

TIP

Bacalah fasilitas ini bila diperlukan atau sesuai dengan kebutuhan.

Toolbar

Toolbar adalah kumpulan tombol-tombol *icon*, dimana setiap *icon* mewakili satu fungsi operasi tertentu yang diperlukan saat pembuatan program. Pada saat anda menjalankan *Visual FoxPro* biasanya anda akan melihat *toolbar standard*.

Gambar 4.2 *Toolbar standard*



Toolbar standar bukanlah satu-satunya *toolbar* yang ada pada *Visual FoxPro*, masih banyak *toolbar* lainnya. Untuk mendapatkan *toolbar* lainnya anda tinggal mengklik submenu *view* kemudian pilih *toolbar*, selanjutnya *Visual FoxPro* akan menampilkan daftar *toolbar*. Dari daftar *toolbar* itulah anda tinggal memilih mana *toolbar* yang diperlukan oleh anda saat ini. Menampilkan semua *Toolbar* kurang baik karena akan mengurangi pandangan anda saat anda membuat program.

BS Window Command

Windows Command ini berfungsi untuk menuliskan perintah-perintah kedalam *Visual FoxPro* secara langsung.

4.1.3 Sistem Database pada Visual FoxPro

Sistem *Database* pada *Visual FoxPro* memiliki beberapa komponen dengan fungsi yang berbeda, Komponen-komponen tersebut adalah:

Tabel

Tabel (*Table*) merupakan kumpulan dari *record-record*, sebuah *table* dalam *Visual FoxPro* bisa menampung sampai dengan 1 (satu) milyar *record*, setiap *record* bisa menampung sampai dengan 255 *field*.

Field

Field merupakan elemen *data* yang membentuk suatu *record*. Ada beberapa tipe *field* dalam *Visual FoxPro*, yaitu :

- | | | |
|---|--|--|
| <input type="checkbox"/> <i>Character</i> | <input type="checkbox"/> <i>General</i> | <input type="checkbox"/> <i>Datetime</i> |
| <input type="checkbox"/> <i>Numeric</i> | <input type="checkbox"/> <i>Integer</i> | <input type="checkbox"/> <i>Character</i> |
| <input type="checkbox"/> <i>Memo</i> | <input type="checkbox"/> <i>Currency</i> | <input type="checkbox"/> <i>Character (binary)</i> |
| <input type="checkbox"/> <i>Date</i> | <input type="checkbox"/> <i>Float</i> | <input type="checkbox"/> <i>Memo Binary</i> |
| <input type="checkbox"/> <i>Logical</i> | <input type="checkbox"/> <i>Double</i> | |

Index

Index adalah suatu *file* yang berisi *pointer* yang menunjuk ke *file* tabel. *File index* akan dibaca sangat cepat karena isinya hanya kunci-kunci *record*. Sebagian besar tipe *field* di atas dapat di *index* kecuali untuk tipe *memo*, *general* dan *memo binary*. Apabila satu *field* tertentu dalam sebuah *record* di *index* maka *field* yang di *index* tersebut akan membentuk sebuah *file* baru dengan ekstension *CDX*. Dengan adanya satu *field* yang di *index* maka sebuah *table* akan terdiri dari dua *file* yang berkaitan dengan ekstension yang berbeda, *CDX* dan *DBF*.

Memo

Didalam *record* ada satu jenis *field* yang bisa memiliki lebar tidak tetap (*Fleksible*). Jenis *field* tersebut adalah *field memo*. *Field memo* lebarnya tergantung kepada memori yang tersedia di komputer tetapi disimpan hanya 4 *bytes*. Karena lebarnya yang bisa tidak tetap *field memo* akan membentuk satu *file* tersendiri yang terkait dan memiliki nama sama dengan tabel induk tetapi dengan ekstension *FPT*.

Dengan adanya *field-field* yang membentuk *file* baru tersebut, maka satu tabel akan dapat terdiri dari tiga jenis *file* yang saling berkaitan, yang secara ringkas dapat dijelaskan sebagai berikut yaitu:

BS

- ❑ **Table**, memiliki lebar *data* tetap dengan ekstension DBF
- ❑ **Memo**, memiliki nama yang sama dengan *file table*, memiliki ekstension FPT dan memiliki lebar *field* yang fleksibel.
- ❑ **File indeks**, juga memiliki nama yang sama dengan *file table*, memiliki ekstension CDX. *File* ini berisi *pointer* untuk tabel.

Null

Null merupakan salah satu ciri dari suatu *field* yang menunjukkan apakah *field* tertentu bisa kosong atau tidak.

4.1.4 Mengakhiri/menutup Visual FoxPro

Untuk menutup *Visual FoxPro* anda tinggal mengklik tombol *close* yang berada di sudut kanan atas *window Visual FoxPro* atau dengan menekan tombol ALT+F4 pada *keyboard* anda.

P

4.2 Membuat Program dengan Microsoft Visual FoxPro

Dalam bagian ini anda akan langsung diajak membuat program aplikasi kecil untuk suatu perusahaan dagang katakanlah PT. ABC. Untuk sebuah perusahaan dagang ada beberapa program yang diperlukan. Mengenai program apa yang harus di buat harus diperhitungkan juga masalah manfaat dan biayanya. Karena itu besar kecilnya perusahaan berpengaruh juga terhadap macam dan jumlah program yang harus dibuat.

Pada contoh ini akan dibuat tiga macam program yaitu persediaan, pembelian dan penjualan. Setiap program yang dibuat akan terdiri dari tiga macam program yaitu program untuk memasukkan *data*, memproses *data* dan program untuk menyajikan informasi (Laporan).

Contoh *form-form* modul yang dibutuhkan oleh perusahaan tersebut adalah sebagai berikut:

Program Persediaan,

Form persediaan berfungsi untuk memasukkan *data Master* persediaan. *Form* ini memberikan fasilitas untuk memelihara *data Master* persediaan seperti manambah, menghapus dan mencetak *data*, *form* tersebut terlihat seperti pada gambar dibawah ini.

J

TIP

Pemrograman pada dasarnya memerintahkan kepada komputer untuk melakukan sesuatu yang berkaitan dengan manajemen data untuk menghasilkan informasi yang berkualitas

J Gambar 4.3 Form untuk memelihara data master persediaan

TIP
 Aktivitas manajemen data dalam arti sempit meliputi: mengumpulkan data, memeriksa konsistensi dan akurasi, menyimpan, memelihara, mengamankan, menyusun dan membaca data untuk diproses dan ditayangkan dalam bentuk informasi.

Program Penjualan,

Gambar 4.4 Form faktur penjualan

Form penjualan di atas berfungsi untuk memasukan data penjualan. Form ini memberikan fasilitas untuk memelihara data transaksi penjualan yang termuat dalam faktur penjualan seperti fasilitas menambah, menghapus dan mencetak. Jumlah penjualan dalam faktur penjualan akan memotong saldo persediaan setelah checkbox potong saldo persediaan di klik.

Program Pembelian.

Form pembelian berfungsi untuk memasukan data pembelian. Form ini memberikan fasilitas untuk memelihara data transaksi pembelian yang termuat dalam faktur pembelian seperti fasilitas menambah, menghapus dan mencetak.

Jumlah pembelian dalam faktur pembelian akan menambah jumlah persediaan setelah checkbox Tambah saldo persediaan pada form pembelian ini di klik. Perhatikan gambar dibawah ini.

Gambar 4.5 Form faktur pembelian

Setelah anda membaca dan mengamati pokok permasalahan dan hal apa saja yang dibutuhkan oleh perusahaan tersebut, sekarang anda dapat memulai membuat program-program tersebut di atas dengan menggunakan *Visual FoxPro*.

4.2.1 Membuat Program Persediaan Barang

Membuat Tampilan Form

1. Buka *Interpreter Visual FoxPro* hingga muncul tampilan seperti pada Gambar 4.1 diatas.
2. Pilih menu *pulldown File, New* atau klik *icon New* pada *toolbar standard* selanjutnya akan tampil *Window New File* seperti pada gambar dibawah ini.

PJ

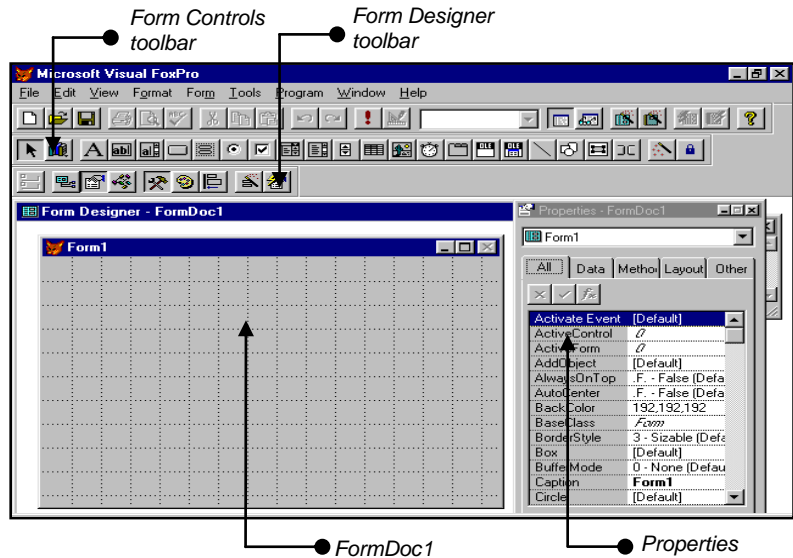
Gambar 4.6 Kotak dialog new file

TIP
Untuk membuat sebuah form anda juga dapat mengetik perintah *Create Form* pada *window Command*

3. Kemudian pilihlah *File type, Form*, dan klik tombol *New File* Selanjutnya *Visual FoxPro* akan menampilkan *Form* kosong dengan nama *Form1* pada *window form designer* seperti pada Gambar 4.7. dibawah ini.

PJ

Gambar 4.7 Form designer dengan nama FormDoc1

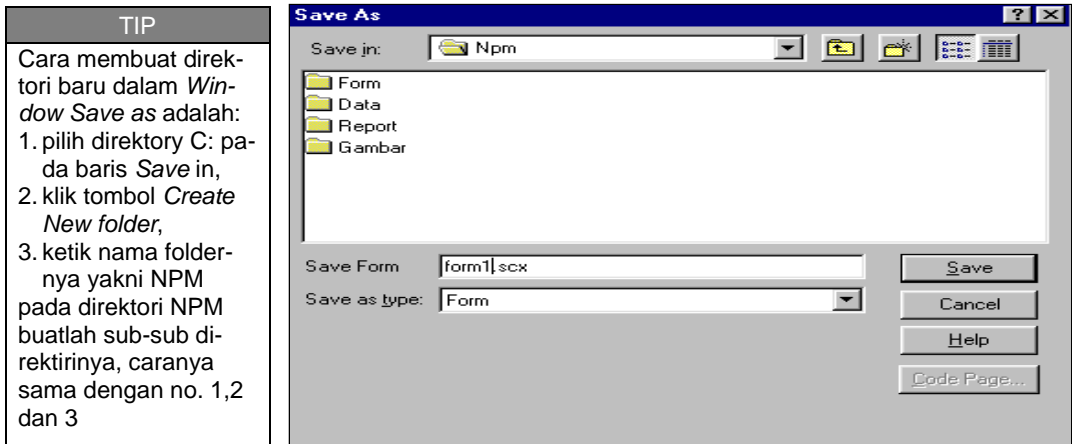


Dalam *properties window* isilah *Height=227* dan *Width=414*. Jika *Properties window* belum ditampilkan maka anda dapat mengklik menu utama *View*, dan sub menu *Properties*. Properti *Height* (tinggi) dan *Width* (lebar) adalah untuk menentukan luas daripada *form* tersebut.

Merekam Form

Setelah anda membuat *form* seperti di atas rekamlah pekerjaan anda dengan mengklik menu utama **File, Save**. Maka akan muncul *Window Save as* seperti gambar dibawah ini.

Gambar 4.8 Window save as



Sebelum merekamnya, terlebih dahulu anda buat direktori baru pada komputer anda dengan nama *C:WPM* (Nomor Pokok Mahasiswa) dan pada direktori NPM buatlah beberapa *folder* seperti

Form, Report, Data, dan jika perlu bisa ditambahkan *folder* lain misalnya *Gambar*. Karena yang akan direkam itu *form* maka simpanlah pekerjaan anda pada *folder Form* dengan nama *Form_brg.scx* kemudian klik tombol *_save* maka *form* yang anda buat sudah terekam.

Lakukanlah langkah-langkah di atas supaya ada keseragaman dalam pembuatan program selanjutnya dan dengan pengelompokan seperti itu anda tidak akan kesulitan dalam mencari mana *form*, mana *data/table* atau mana *report*.

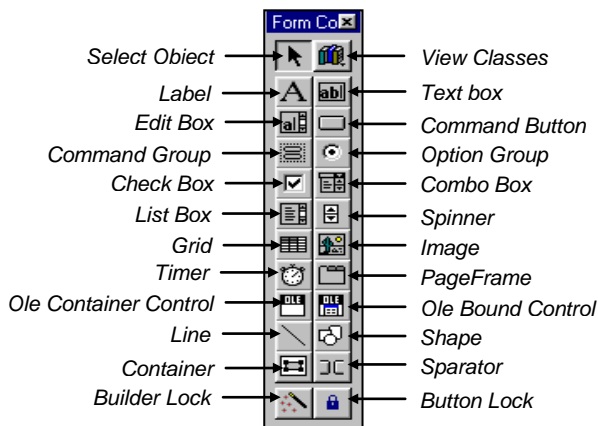
Facilities Pendukung dalam Pembuatan Form

Form Controls Toolbar

Form Controls toolbar berisi berbagai fasilitas untuk membuat suatu objek yang dapat dipasang ke dalam *form*. Setiap objek yang akan dihasilkan ditunjukkan dalam bentuk *icon*.

BS

Gambar 4.9 *Form controls toolbar*



TIP
 Untuk menampilkan *form controls toolbar* kliklah menu utama *View* pada *window Form designer* kemudian pilihlah sub menu *Form Controls toolbar*.

Berikut ini uraian dari *icon-icon* pada *Form controls toolbar* di atas:

- Select Object** adalah fasilitas untuk mengubah atau memin-dahkan sebuah objek, biasanya tombol ini aktif secara oto-matis tanpa harus di klik,
- View Classes** tombol untuk menampilkan atau memilih *class library* yang digunakan,
- Label**, digunakan untuk menampilkan teks baik sebagai judul *form*, keterangan dan yang lainnya
- Text Box** digunakan untuk memasukan atau menampilkan satu baris teks *data* oleh user.
- Edit Box** fungsinya sama dengan *Text Box* tetapi *data* yang diisikan dapat lebih dari satu baris.

TIP

Bacalah fasilitas ini bila diperlukan atau sesuai dengan kebutuhan.

- ❑ **Command Button** (Tombol perintah) berfungsi sebagai sebuah tombol yang dapat mengerjakan perintah tertentu dimana prosesnya anda buat sendiri melalui program.
- ❑ **Command Group** digunakan untuk membuat sejumlah *Command Button*.
- ❑ **Option Group** digunakan untuk membuat pilihan bagi *user* dimana *user* hanya dapat memilih satu pilihan.
- ❑ **Check Box** digunakan untuk membuat dua pilihan atau dua kondisi misalnya *true* (.T.) atau *false* (.F.) pada sebuah *CheckBox*.
- ❑ **Combo Box** digunakan untuk menampilkan sebuah daftar *drop down* dimana *user* dapat memilih satu item dari daftar tersebut.
- ❑ **List Box** fungsinya hampir sama dengan *Combo Box* tetapi disini *user* dapat mengatur beberapa item yang ada di dalamnya.
- ❑ **Spinner** digunakan untuk menampilkan nilai numerik dengan sebuah *range* yang telah ditentukan.
- ❑ **Grid** digunakan untuk membuat *browser* yang dapat menampilkan *data* yang terdapat pada tabel.
- ❑ **Image** berfungsi untuk menampilkan sebuah gambar atau grafik di dalam sebuah *form*.
- ❑ **Timer** adalah sebuah fasilitas untuk menampilkan waktu.
- ❑ **PageFrame** digunakan untuk membuat sebuah agenda yang terdiri dari beberapa halaman dimana halamannya tersebut dapat ditentukan atau dikontrol.
- ❑ **Ole Container Control** digunakan untuk menampung objek *OLE* didalam aplikasi anda.
- ❑ **Ole Bound Control** fungsinya mirip dengan *Ole Container Control* tetapi biasanya *Ole Bound Control* digunakan dalam *field* atau *query*.
- ❑ **Line** digunakan untuk membuat sebuah garis.
- ❑ **Shape** digunakan untuk membuat kotak, oval, atau lingkaran di dalam aplikasi anda.
- ❑ **Container** digunakan untuk memasukan suatu objek kedalam *form* yang aktif.
- ❑ **Sparator** digunakan untuk membuat jarak antara *control toolbar*.
- ❑ **Button Lock** digunakan untuk mengunci kontrol yang anda gunakan sehingga anda dapat menambahkan kontrol yang sama tanpa harus mengklik ulang kontrol tersebut.
- ❑ **Builder Lock** digunakan untuk membuka sebuah *builder* kontrol kontrol baru yang ditambahkan kedalam *form*.

Properties Window

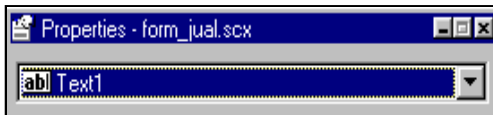
BS

Properties window berisi daftar *property*, *events* dan *methods* untuk *form*, *data environment*, *cursor*, *relation* atau *control*. Komponen-komponen *properties* yang perlu anda ketahui adalah sebagai berikut:

1. List Object

List Object adalah daftar dari *control/object* yang terdapat di dalam *Form designer* yang sedang anda buka pada *window Visual FoxPro*.

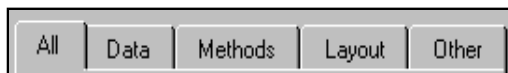
Gambar 4.10 List object



Di sini anda dapat memilih objek mana yang akan anda aktifkan atau akan anda modifikasi, dengan mengklik panah *drop down* yang berada di paling kanan *list object*.

2. Tabs

Gambar 4.11 Tabs



Tabs digunakan untuk menampilkan *properties*, *events*, dan *methods* berdasarkan kategori-kategori di bawah ini:

- All** menampilkan seluruh *properties*, *events*, dan *methods*.
- Data** menampilkan *properties* yang berhubungan dengan *data* atau objek yang akan menampilkan *data*.
- Methods** menampilkan *methods* dan *events*.
- Layout** menampilkan semua *layout properties*.
- Other** menampilkan properti lain dan *properties* yang didefinisikan oleh user.

TIP
Untuk menampilkan *Properties window*, kliklah menu utama *View* pada *window Form designer* kemudian pilihlah sub menu *Properties*.

3. Properties Settings Box

Gambar 4.12 Settings box



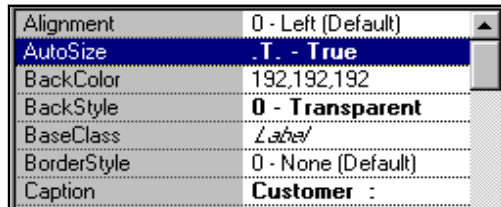
Pada *settings Box* ini anda dapat merubah nilai *properties* dari setiap properti yang anda pilih pada daftar *properties*.

4. Properties List

Properties List adalah daftar dari nama properti-properti yang memiliki dua kolom tampilan, kolom sebelah kiri merupakan nama *properties* dan kolom sebelah kanan merupakan nilai

dari properti yang bersangkutan. Untuk *properties* yang nilainya merupakan pilihan, anda dapat memilih nilainya dengan cara mengklik dua kali *properties* tersebut sampai anda menemukan nilai yang anda maksud atau dapat memilihnya pada *Properties setting box* dengan mengklik *combobox* yang menampilkan nilai-nilai *properties* tersebut, untuk nilai *properties* yang bukan merupakan pilihan anda dapat mengetiknya pada *Properties setting box* dan untuk *method* dan *event* anda dapat mengisinya pada *Code window*.

Gambar 4.13 Properties list



Membuat Table

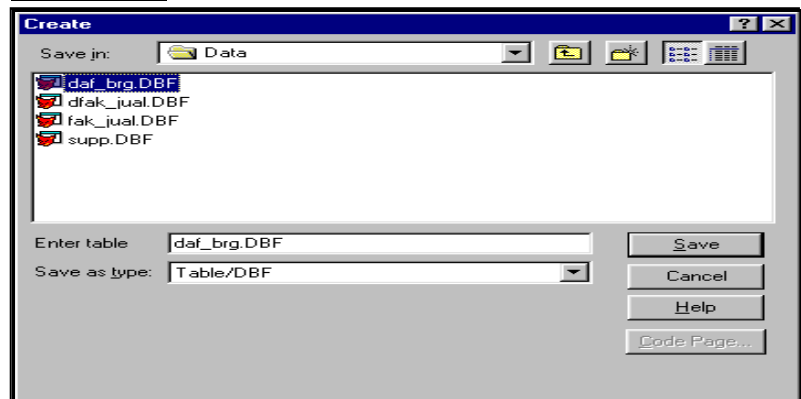
Langkah-langkah pembuatan *table*, yaitu:

PJ

1. Buka menu *pull-down File, New* maka akan muncul *window New file*. Seperti pada gambar 4.6
2. Dalam *Window New file* pada kolom *file type* pilihlah *Table* kemudian klik tombol *New file*.
3. Maka akan muncul *window Create*, ketiklah nama *filenya* dengan nama *file Daf_brg* (kependekan dari daftar barang) dalam direktori *C:\Wpm\Data* kemudian klik tombol *Save*, lihat gambar di bawah ini.

TIP
Selain dengan langkah-langkah di samping, Untuk membuat sebuah *table* anda juga dapat megetik perintah *Create* pada *command window*

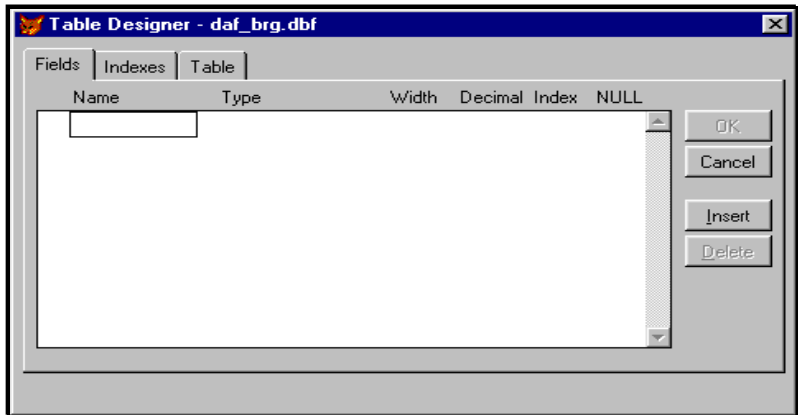
Gambar 4.14 Window create



Setelah anda merekam nama *file* yang akan anda buat maka akan muncul lagi *Window Table Designer – daf_brg* seperti gambar di bawah ini.

Gambar 4.15 Table designer – daf_brg masih kosong

PJ

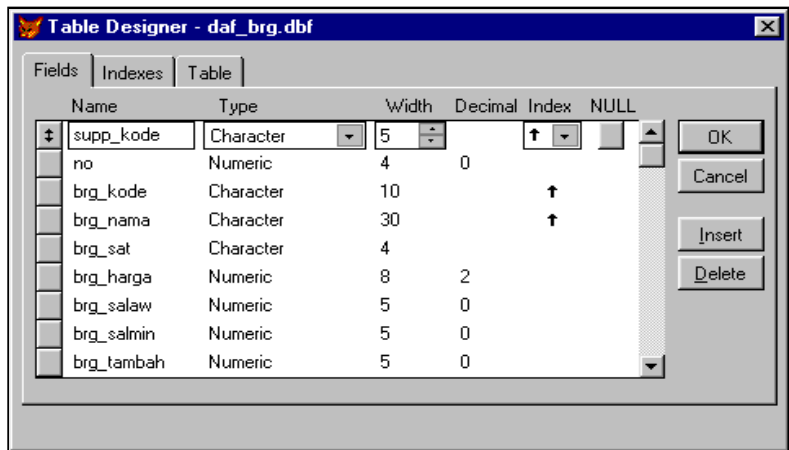


Selanjutnya anda isi nama *field*, *Type*, *width* dan *index* kalau memang akan di *index*, sebagai acuan anda dapat melihat tabel di bawah ini dan supaya seragam anda harus mengisi Tabel Daf_brg sesuai dengan daftar yang ada di bawah ini.

Tabel 4.1 Daftar field untuk tabel Daf_brg

Name	Type	Width	Decimal	Index
Supp_kode	Character	5		Ya
No	Numeric	4		
Brg_kode	Character	10		Ya
Brg_nama	Character	30		Ya
Brg_sat	Character	4		
Brg_harga	Numeric	8	2	
Brg_salaw	Numeric	5		
Brg_salmin	Numeric	5		
Brg_tambah	Numeric	5		
Brg_kurang	Numeric	5		
Brg_Stock	Numeric	5		

Gambar 4.16 Tabel daf_brg setelah diisi



TIP
 Jika anda akan menghapus atau mengganti nama *field* yang telah anda buat dan *field* tersebut diindex maka sebelum menghapus atau mengganti nama *field* tersebut, *index*nya harus di-tiadakan terlebih dahulu atau tidak di *index* .

PJ Gambar panah ke atas pada kolom *index* menunjukkan bahwa *field* tersebut di atas diindex untuk mengindex sebuah *field* anda tinggal memilih gambar panah tersebut caranya dengan mengklik kolom tersebut. Setelah selesai pilihlah tombol OK kemudian akan muncul kotak dialog yang menyatakan bahwa tabel tersebut akan diinput sekarang atau tidak maka anda pilih *No* untuk menyatakan tidak.

Gambar 4.17 Message box

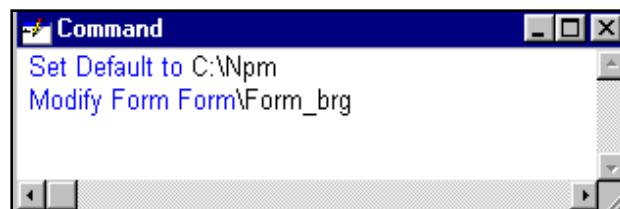


Memanggil dan Memodifikasi Form

Perintah untuk memanggil *Form* yang telah anda buat adalah perintah *Modify Form* dengan langkah-langkah sebagai berikut:

1. Ketiklah *Set default to C:\Npm\Form* (direktori beserta nama *folder* dimana form tersebut disimpan), tekan *Enter*
2. Panggil formnya yaitu *Form_brg* dengan perintah *modi form form_brg* kemudian tekan *Enter* seperti pada gambar dibawah ini, maka akan muncul *form* yang telah anda buat tadi

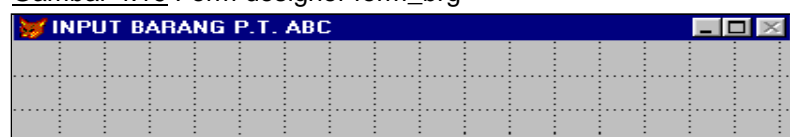
Gambar 4.18 Command window



Setelah *form* yang dipanggil muncul berilah judul form tersebut dengan cara:


1. Jika *Properties Window* belum tampil, maka pilihlah menu *View* dan sub menu *Properties*,
2. Pada *properties list* pilihlah *caption* dan ketiklah judul untuk form tersebut contohnya INPUT BARANG PT. ABC, dan hasilnya tampak seperti pada gambar dibawah ini.

Gambar 4.19 Form designer form_brg

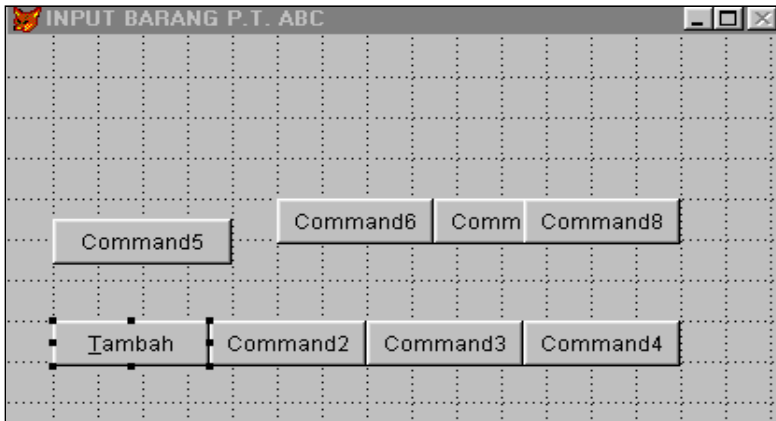




Selanjutnya masukanlah 8 buah tombol (*Command Button*) kedalamnya kemudian susun dengan rapi tombol-tombol tersebut di bagian bawah *form*. Cara memasukan tombol tersebut adalah:

1. Pilihlah tombol *control command button*  pada *form controls toolbar*
2. Tempatkan *mouse pointer* pada lokasi *form* yang anda kehendaki dan klik maka akan muncul sebuah tombol yang siap untuk diedit.
3. Untuk memindahkan posisi *Command button* tersebut anda harus *mendrag* (klik tahan dan geser) *command button* tersebut sesuai keinginan.

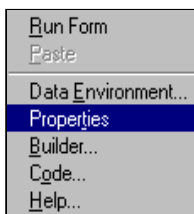
Gambar 4.20 *Form designer Form1*



Untuk memasukan *caption* atau properti-properti lainnya pada sebuah objek ini adalah:

1. Jika *Properties window* belum ditampilkan, klik mouse kanan pada salah satu objek *Command button* di atas maka akan muncul menu *pulldown*.

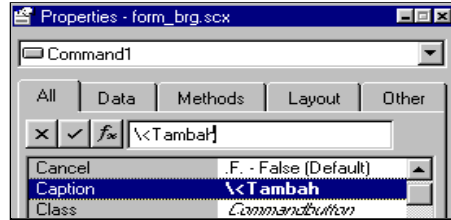
Gambar 4.21 *Menu pulldown*



2. Pilihlah sub menu *properties*, selanjutnya *Properties window* akan tampil.
3. dan pada *properties list* isilah *captionnya* sesuai dengan yang anda inginkan, sebagai contoh lihat gambar *properties* dibawah ini.

PJ

Gambar 4.22 Properties window

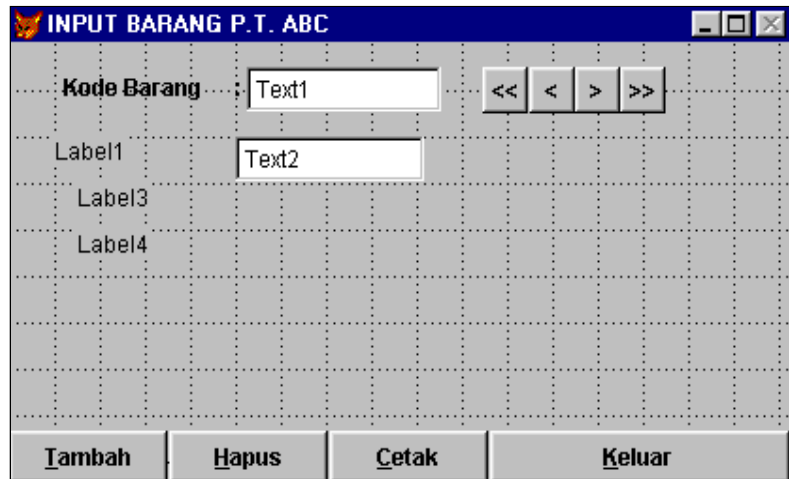


Selanjutnya masukan kedalam *form* tersebut 11 buah *Label* (untuk title), 10 *Text Box* (untuk data) dan 1 buah *Container* (untuk menampung), cara-cara memasukan sebuah objek semuanya sama, baik itu *CommandButton*, *Label*, *Textbox* dan lain-lain yaitu dengan cara:

1. Klik tombol sebuah objek (*control*) pada *form controls toolbar*
2. Tempatkan *mouse pointer* anda pada lokasi *form* yang anda kehendaki dan klik,
3. Kemudian setiap objek dimodifikasi melalui *properties window* baik itu *captionnya*, *posisinya* dan lain-lain sesuai dengan daftar *properties* yang ada.

TIP
Sebagai acuan anda lihat kembali Gambar 4.3 Form persediaan & input barang.

Gambar 4.23 Form1 dengan objek-objek didalamnya



Sebagai panduan, berikut ini daftar objek yang digunakan dalam *Form Form_brg* lengkap dengan *properties* yang digunakan dalam setiap objek tersebut, dan untuk tahap awal anda diminta untuk mengikuti dulu pengisian *properties* untuk tiap objek berdasarkan daftar pada tabel 4.2 berikut ini.

Tabel 4.2 Daftar objek pada form Form_brg




Nama Objek	Properties yang digunakan
Form1	Caption='INPUT BARANG PT ABC', Height=227, Left=150, ShowWindow=1-In Top Level Form, Top=10, Width=414
Label1	AutoSize=.T., BackStyle=0-Transparant, Caption='Kode Barang :', FontBold=.T., FontSize=9, FontStyle=Arial, Left=29, dan Top=15 *Catatan: Untuk seluruh label gunakan <i>properties</i> yang sama dengan label1 kecuali untuk properti <i>Caption</i> , <i>Left</i> dan <i>Top</i> .
Label2	Caption='Nama Barang :', Left=25, Top=45
Label3	Caption='Satuan :', Left=62, Top=71
Label4	Caption='Harga beli / satuan :', Left=7, Top=93
Label5	Caption='Quantitas/Stock :', Left=219, Top=72
Label6	Caption='Harga Jual :', Left=236, Top=93
Container1	BackColor=128,128,128, Height=78, Left=4, Top=120, Width=409
Label7	Caption='Keterangan', Left=13, Top=123
Label8	Caption='Saldo awal :', Left=45, Top=128
Label9	Caption='Saldo Minimal :', Left=31, Top=173
Label10	Caption='Jml Brg Masuk :', Left=220, Top=151
Label11	Caption='Jml Brg Keluar :', Left=221, Top=174
Text1	Height=23, Left=126, Top=12, Width=103
Text2	Height=23, Left=126, Top=40, Width=281
Text3	Height=23, Left=126, Top=67, Width=91
Text4	Height=23, Left=126, Top=91, Width=91
Text5	Height=23, Left=126, Top=143, Width=92
Text6	Height=23, Left=126, Top=169, Width=92
Text7	Height=23, Left=315, Top=143, Width=92
Text8	Height=23, Left=315, Top=169, Width=92
Text9	Height=23, Left=315, Top=89, Width=92
Text10	Height=23, Left=315, Top=65, Width=92
Command1	Caption='\<Tambah', Height=27, Left=0, Top=200, Width=84
Command2	Caption='\<Hapus', Height=27, Left=85, Top=200, Width=84
Command3	Caption='\<Cetak', Height=27, Left=170, Top=200, Width=84
Command4	Caption='\<Keluar', Height=27, Left=255, Top=200, Width=159
Command5	Caption='<<', Height=24, Left=251, Name='Atas', Top=12, Width=24
Command6	Caption='<', Height=24, Left=276, Name='Mundur', Top=12, Width=24
Command7	Caption='>', Height=24, Left=300, Name='Maju', Top=12, Width=24
Command8	Caption='>>', Height=24, Left=324, Name='Bawah', Top=12, Width=24

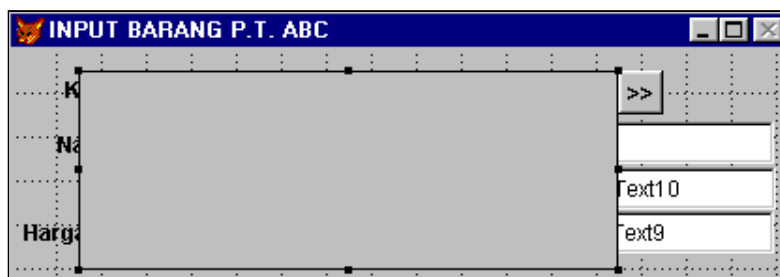
TIP
Untuk menggaris bawah salah satu huruf *caption* sebuah tombol misalnya kata Tambah maka tulislah \<Tambah pada properti *caption* sehingga hasilnya akan seperti ini Tambah.

TIP
Caption tombol yang memiliki garis bawah pada salah satu hurufnya selain diklik tombol tersebut nantinya dapat juga dijalankan dengan menekan ALT+(huruf yang digarisbawahi) sebagai pengganti klik, misal Tambah dapat dengan menekan ALT+T

Khusus untuk *Container* cara memasukan objek ini adalah sebagai berikut:

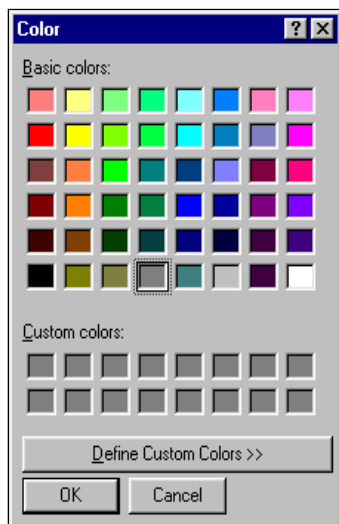
- PJ** 1. Klik tombol *control Container*  pada *Form controls toolbar*. Tempatkan dan klik di atas *form* yang anda kehendaki, atau dapat pula dengan klik dan tahan kemudian geser (*drag*) ke kiri atau ke kanan untuk menentukan panjang dan ke atas atau ke bawah untuk menentukan tinggi, setelah luasnya sesuai lepas.

Gambar 4.24 Sebuah container di dalam form form_brg



2. Untuk menentukan panjang dan lebar dapat pula diedit di dalam *properties* yang nilainya tertera pada tabel 4.2 daftar *properties* di atas (*Container1*).
3. Berilah warna *background* dengan memilih *BackColor* pada *properties list*.
4. Untuk memilih warna yang anda sukai klik tombol yang berada di sisi kanan *setting box properties*, maka akan muncul *window color* seperti pada gambar di bawah ini,

Gambar 4.25 Window color (kotak pilihan warna)



5. Agar *container* ini berada paling bawah pada *form* form_brg diantara objek-objek lainnya, adalah dengan memilih menu utama *Format* dan sub menu *Send to Back*, dan kalau ingin berada paling atas pilihlah *Bring to Front*.

Dan hasil akhir dari form designer Form_brg tampak seperti pada gambar dibawah ini.

Gambar 4.26 Tampilan akhir form designer Form_brg



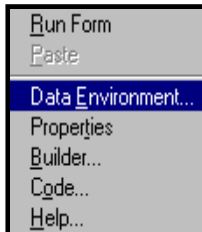
Memasukan Tabel ke dalam Data Environment

Cara memasukan tabel ke dalam *form* adalah dengan tahap-tahap sebagai berikut:

PJ

- 1 Klik tombol mouse kanan pada *form* dalam hal ini form_brg kemudian akan muncul menu seperti dibawah ini.

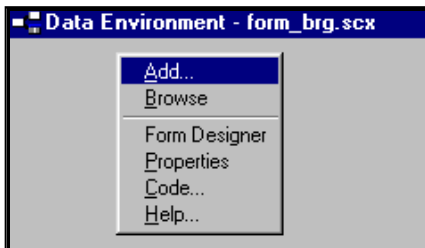
Gambar 4.27 Menu pilihan



TIP
Selain dengan cara di samping, menampilkan *data environment* dapat juga dengan cara mengklik menu utama *View* dan klik sub menu *Data Environment*

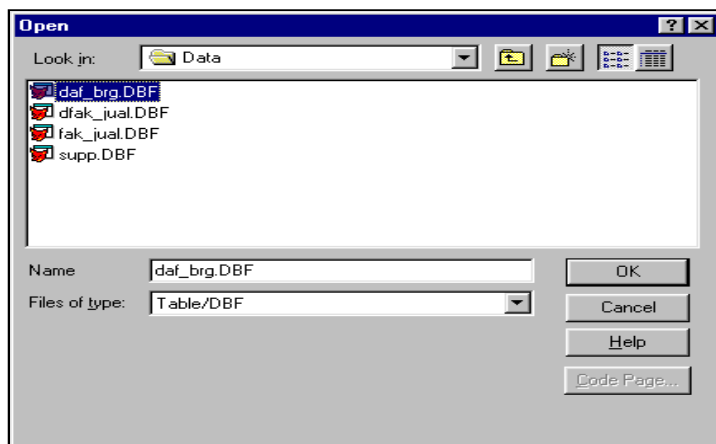
- 2 Pilih dan klik sub menu *Data Environment* maka akan muncul tampilan *Data Environment – form_brg*, setelah itu klik tombol *mouse* kanan pada area *window* tersebut dan akan muncul menu seperti pada gambar di bawah ini.

Gambar 4.28 Window data environment



- PJ** 3 Pilih dan klik Add pada menu di atas maka akan muncul tampilan berikutnya adalah *window Open*,

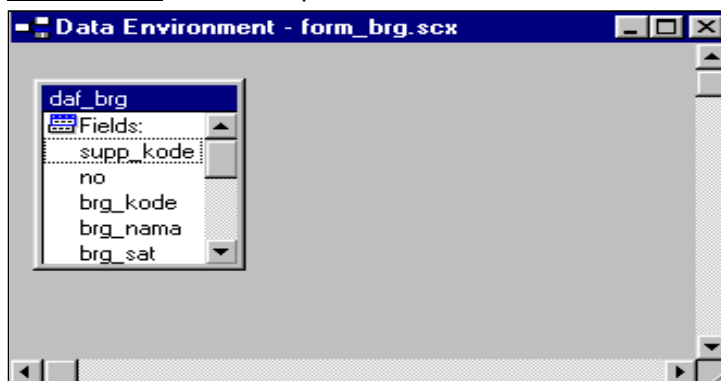
Gambar 4.29 *Window open*



Pilihlah nama tabel yang akan anda masukan kedalam *form Form_brg*, di sini anda tinggal memakai tabel yang anda buat tadi yaitu tabel *Daf_brg* kemudian klik tombol OK maka tabel tersebut akan masuk ke dalam *form* anda dan telah siap untuk anda olah sesuai dengan yang diharapkan.

- 4 Tahap terakhir adalah menutup *window Data Environment*, dengan cara mengklik tombol *close* yang berada di sudut kanan atas *window* tersebut.

Gambar 4.30 Hasil akhir proses data environment



Memasukan Field ke dalam Text Box

Form yang telah selesai anda buat bila belum diisi program belum bisa memproses apa-apa, untuk itu anda harus memberi identitas terhadap *Textbox-Textbox* yang anda tampilkan pada *form* tersebut, yaitu dengan memasukan *field* dari *data* atau tabel yang

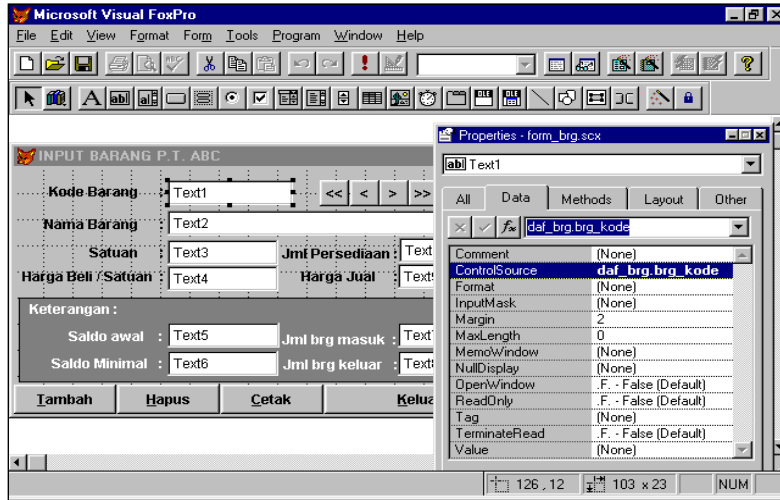
telah anda masukan tadi kedalam *Text Box* yang ada dalam *form*, agar lebih jelasnya perhatikan uraian di bawah ini.



Cara memasukan *field* ke dalam *Text Box* adalah dengan tahap-tahap sebagai berikut:

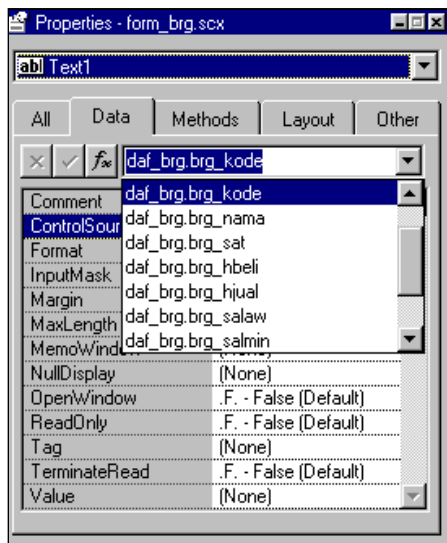
1. Pilih atau klik terlebih dahulu *Text Box* yang akan diinisialisasi.

Gambar 4.31 Window form designer Microsoft Visual FoxPro



2. Pada *properties* klik *tab Data* kemudian pilih properti *Control source*
3. Klik daftar *field* pada *Combo Setting Box*, maka akan muncul daftar field dari tabel yang sudah dimasukan ke dalam *data environment* seperti pada gambar dibawah.

Gambar 4.32 Properties control source



TIP
Selain *properties*, *Text box* juga memiliki *window* lain untuk memodifikasi atau mengisi *value* dari *text box* tersebut adalah *window Text box builder*, cara menampilkan *window* tersebut yaitu dengan cara mengklik kanan *text box* tersebut maka akan muncul menu *pull-down* pilihlah sub menu *Builder*.



4. Pilihlah nama *field* untuk tiap *Text Box* sesuai dengan tabel di bawah ini:

Tabel 4.3 Control source untuk tiap Text Box

Nama Text Box	Control Source diisi dengan
Text1	Daf_brg.brg_Kode
Text2	Daf_brg.brg_Nama
Text3	Daf_brg.brg_Sat
Text4	Daf_brg.brg_Hbeli
Text5	Daf_brg.brg_Salaw
Text6	Daf_brg.brg_Salmin
Text7	Daf_brg.brg_Tambah
Text8	Daf_brg.brg_Kurang
Text9	Daf_brg.brg_Hjual
Text10	Daf_brg.brg_stock

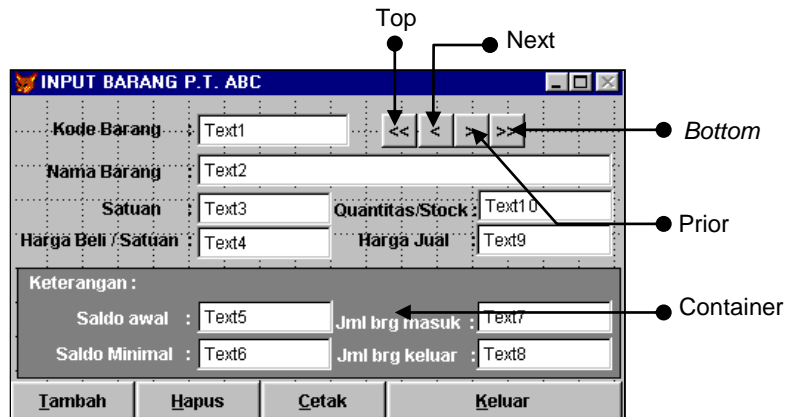
Mengisi Program pada setiap Objek

Setelah anda membuat fasilitas untuk memasukan *data (Form input)*, selanjutnya anda buat suatu rangkaian perintah (*program*) yang bekerja setelah kondisi tertentu dipenuhi, kondisi tersebut misalkan:

- Akhir memasukan *data*
- Setelah mengklik tombol tertentu
- Setelah menekan tombol *keyboard* tertentu, dan persyaratan lainnya misalkan tombol aktif kalau yang mengerjakan user tertentu, sebagai *control*.

Pada praktika PAK saat ini anda tidak belajar *control*, *control* diberikan pada modul praktika lanjutan yaitu Praktika Sistem Informasi Manajemen atau Sistem Informasi Akuntansi.

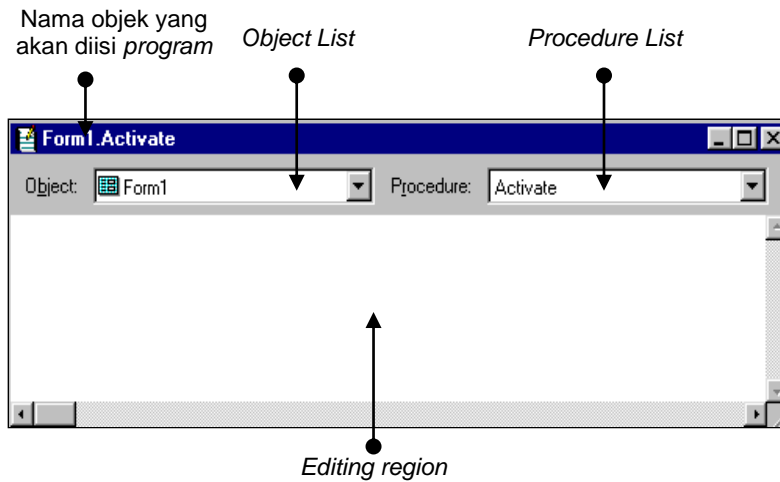
Gambar 4.33 Form designer Form_brg





Setelah anda memasukan *field* kedalam *Text Box* sekarang anda akan mulai mengisi program pada setiap objek. Cara mengisi perintah (*program*) pada sebuah objek/*control* yaitu dengan mengklik dua kali (*double click*) objek yang bersangkutan, maka secara otomatis akan masuk ke dalam *code window program*, seperti pada gambar dibawah ini.

Gambar 4.34 *Code window program*



Berikut adalah penjelasan dari gambar diatas:

- ❑ **Object List** adalah daftar seluruh objek yang anda masukan kedalam form tersebut termasuk formnya itu sendiri.
- ❑ **Procedure List** adalah daftar seluruh *event* yang dapat digunakan berdasarkan objek yang dipilih didalam *Object list*. Ketika anda memilih *event* atau *method* pada *procedure list*, parameter-parameter inti untuk *event* atau *method* tertentu (jika ada) akan ditampilkan dalam *editing region* pada *code window* tersebut. Nama *event* yang tercetak tebal pada *procedur list*, tandanya bahwa *event* tersebut berisi perintah.

Berikut ini adalah daftar *event* yang terdapat pada *procedure list* beserta keterangannya.

Tabel 4.4 Daftar event dan bilamana terjadinya

Nama Event	Menjalankan Event pada saat...
<i>Activate</i>	Objek menerima <i>focus</i>
<i>Click</i>	Menekan dan melepaskannya kembali (klik) tombol <i>mouse</i> kiri
<i>DbClick</i>	Ketika anda mengetuk/mengklik tombol <i>mouse</i> kiri sebanyak dua kali dalam waktu yang cepat
<i>Deactivate</i>	Menonaktifkan <i>container object</i> seperti <i>form</i> atau dengan kata lain menutup <i>form</i>

J

Lanjutan tabel 4.4

TIP
Masuk ke <i>code window</i> untuk mengisi perintah pada sebuah objek dapat pula dengan mengdouble click nama <i>procedure</i> atau <i>event</i> pada <i>properties</i> , sehingga anda tidak perlu lagi memilih nama <i>procedurnya</i> pada <i>code window</i> tersebut.

Nama Event	Menjalankan Event pada saat...
<i>Deleted</i>	Menandai sebuah <i>record</i> pada sebuah tabel untuk dihapus
<i>Destroy</i>	Ketika anda keluar dari sebuah objek
<i>DragDrop</i>	Ketika anda menjalankan <i>drag and drop operation</i> (menarik sambil ditekan kemudian dijatuhkan/dilepaskan)
<i>DragDrop</i>	Ketika anda menjalankan <i>drag and drop operation</i> (menarik sambil ditekan kemudian dijatuhkan/dilepaskan)
<i>DragOver</i>	Mendrag sebuah objek diatas target potensial
<i>DropDown</i>	Ketika anda mengklik panah pada sebuah <i>Combo box</i> atau <i>List box</i> , dan <i>Drop down</i> terjadi pada saat sebelum jatuhnya list tersebut. Gunakan <i>event</i> ini untuk menyeleksi ulang sebelum list ditampilkan.
<i>Error</i>	Ketika <i>event</i> menemukan error pada saat menjalankan aplikasi
<i>ErrorMessage</i>	Meninggalkan <i>event</i> yang seharusnya dikerjakan, <i>event</i> ini digunakan untuk mengembalikan pesan <i>error</i>
<i>GotFocus</i>	Ketika anda memindahkan <i>focus</i> atau <i>cursor</i> ke dalam sebuah <i>field</i> atau <i>object</i>
<i>Init</i>	Sebuah objek dijalankan dalam <i>memory</i> sebelum menampilkannya
<i>InteractiveChange</i>	Ketika anda merubah nilai pada sebuah <i>object</i>
<i>KeyPress</i>	Menekan sebuah tombol pada <i>keyboard</i> yang mana tombol tersebut telah didefinisikan
<i>Load</i>	Ketika anda menjalankan sebuah objek
<i>LostFocus</i>	Ketika <i>focus</i> keluar atau meninggalkan sebuah objek
<i>Message</i>	Ketika <i>object</i> mendapatkan <i>focus</i> , <i>event</i> ini digunakan untuk menampilkan pesan pada status bar
<i>MouseDown</i>	Ketika anda menekan tombol <i>mouse</i>
<i>MouseMove</i>	Ketika anda menggerakkan <i>mouse</i>
<i>MouseUp</i>	Ketika anda melepaskan tekanan pada tombol <i>mouse</i>
<i>Moved</i>	Ketika anda memindahkan posisi kolom pada sebuah <i>Grid</i>

Lanjutan tabel 4.4

J

Nama Event	Menjalankan Event pada saat...
<i>ProgrammaticChange</i>	Ketika nilai dari sebuah <i>object</i> berubah, biasanya digunakan pada <i>object</i> yang menampilkan waktu (menampilkan Jam)
<i>Resize</i>	Ketika anda merubah lebar kolom pada sebuah Grid
<i>RightClick</i>	Ketika anda menekan dan melepaskannya kembali (mengklik) tombol <i>mouse</i> kanan
<i>Scrolled</i>	Ketika anda menggulung <i>scroll grid</i> (<i>Scrolling</i>) hingga selesai pada sebuah <i>Grid</i>
<i>Timer</i>	Ketika penetapan nomor untuk milli detik terlewatkan
<i>Unload</i>	Ketika keluar dari objek atau keluar dari aplikasi (menutup form)
<i>Valid</i>	Ketika anda memindahkan <i>focus</i> dari sebuah <i>object</i> dan sebelum sebuah <i>object</i> ditinggalkan focus tersebut. Jika isi <i>event</i> tidak sesuai dengan <i>event</i> ini maka <i>focus</i> tidak akan dapat meninggalkan <i>object</i> ini
<i>When</i>	Sebelum sebuah <i>object</i> mendapatkan focus. Jika isi <i>event</i> tidak sesuai dengan <i>event</i> ini maka <i>Visual FoxPro</i> tidak akan memberikan <i>focus</i> untuk <i>object</i> ini

- ❑ **Editing region** adalah lokasi untuk mengetik perintah-perintah pemrograman.

Setelah anda membaca dan memahami *code window* tersebut sekarang anda mulai mengetik perintah untuk setiap objek dalam form form_brg. Perlu diketahui bahwa untuk menyatakan sebuah objek misal *Command1* dalam pemrograman *Visual FoxPro* harus dimulai dengan kalimat *ThisForm* (menyatakan form bersangkutan) untuk melanjutkan perintah yaitu dengan tanda . (titik) kemudian objek yang dimaksud yaitu *Command1* dan terakhir perintahnya misalnya *Setfocus* (perintah untuk menempatkan (memfokuskan) *cursor* atau *pointer* pada sebuah objek). Sehingga dalam pengetikan perintah tersebut adalah sebagai berikut:

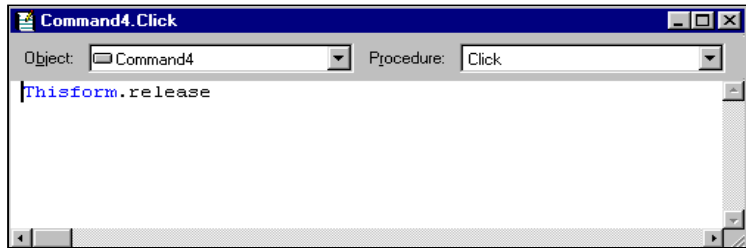
Thisform.Command1.Setfocus

Sebagai panduan untuk anda, anda ikuti perintah-perintah untuk setiap objek seperti pada uraian di bawah ini.

PJ

❑ Tombol Keluar (Command4-Click)

Gambar 4.35 Code window command4



Cara untuk membuat perintah diatas adalah sebagai berikut:

1. Klik dua kali (*double click*) objek *command4* pada form Form_brg, maka akan masuk ke *code window*.
2. Pilihlah *Click* pada *procedure list*.
3. Dan ketiklah perintah di bawah ini pada *editing region*,

Thisform.Release

Perintah *release* adalah satu perintah untuk mengclose atau menutup *form*.

Penulisan *Command4-Click* pada judul diatas menyatakan objeknya *Command4* dan *event* yang digunakan adalah *Click*. Untuk mengisi objek-objek lainnya lakukanlah hal yang sama dengan langkah-langkah diatas.

❑ Tombol Tambah (Command1-Click)

TIP
Menambah record dilakukan setelah record paling bawah.

Perintah untuk menambah *Record* pada sebuah tabel yaitu *Append Blank*, sebelum melakukan editing pada sebuah tabel terlebih dahulu nama tabelnya harus di sebutkan. Lihat perintah untuk menambah *record* di bawah ini.

Sele daf_brg
GO BOTT
Append Blank
Thisform.refresh
Thisform.text1.setfocus

Berikut uraian dari perintah di atas:

Select daf_brg yaitu perintah untuk mengaktifkan sebuah tabel yang mempunyai nama *file* *daf_brg*

GO BOTT adalah kependekan dari *GO BOTTOM*, pada *Visual FoxPro* suatu perintah dapat disingkat menjadi empat huruf, dan perintah *Go Bottom* ini adalah satu perintah untuk menempatkan *pointer* di *record* paling bawah pada tabel tersebut.

Append blank adalah perintah untuk membuat sebuah record baru yang kosong. *Thisform.refresh* adalah untuk me-*refresh Form*

Thisform.text1.setfocus adalah untuk menempatkan *pointer* pada *Text1*. Setelah tombol ini dijalankan (diklik) user dapat langsung mengisi *text-text* tersebut dengan *data* yang dikehendaki sesuai dengan judul *text* tersebut.

□ Tombol *Hapus (Command2-Click)*

Tombol Hapus pada *form Form_brg* berfungsi sebagai tombol untuk menghapus *data Daf_brg* perintah untuk menghapus satu *record* yang aktif adalah *Delete* perintah *delete* ini juga bisa diberi kondisi, ada juga perintah *Delete all* perintah ini perintah untuk menghapus seluruh *data* yang ada di dalam *file* yang bersangkutan, *delete all* juga bisa diberi kondisi seperti halnya perintah *delete*.

Misalnya pada tabel A ada *field* Nama yang berisi nama-nama dari huruf A sampai Z, kemudian anda ingin menghapus nama yang berhuruf awal B, maka perintah yang harus anda buat adalah seperti di bawah ini:

```
Sele A
Delete all for left>Nama,1)="B"
```

Maka seluruh nama yang dimulai dengan huruf B semuanya akan dihapus, untuk membatalkan semua yang dihapus tadi adalah dengan perintah *Recall* caranya sama dengan perintah diatas tinggal mengganti *delete* dengan *recall*.

Perintah perintah yang digunakan untuk tombol menghapus (*Command2*) pada *form form_brg* adalah sebagai berikut:

```
Sele daf_brg
Delete
skip
if eof()
    go bott
endif
Thisform.refresh()
```

Berikut uraian dari perintah di atas:

Skip adalah satu perintah untuk memindahkan *pointer* satu *record* ke bawah misalkan posisi *record* ada di nomor 1 maka akan berpindah ke posisi nomor 2.

Perintah IF...ENDIF adalah suatu perintah dimana perintah ini sudah mengacu pada suatu kondisi yang memerlukan logika, perintah ini penting dikuasai untuk membuat program *Expert System* dalam sistem informasi manajemen. Uraian dari perintah IF..ENDIF dapat anda baca pada bagian akhir buku ini. Perhatikan perintah dibawah.

```
Skip
If eof()
    Go bottom
Endif
```

Perintah diatas mempunyai arti sebagai berikut, jika posisi *pointer* berada di akhir file (*EOF()* (*End Of File*)) maka pindahkan posisinya ke record paling bawah, karena bagian ELSE bersifat *opsional* maka diabaikan saja artinya bagian ELSE sudah diwakilli oleh perintah sebelum IF...ENDIF yaitu *Skip*.

❑ Tombol *Cetak* (*Command3-Click*)

Tombol ini adalah untuk menampilkan *Report* yang telah anda buat, pembuatan *reportnya* akan dibahas nanti, sekarang anda perhatikan bagaimana menampilkan *report* yang telah anda buat, di dalam sebuah *form*. Cara menampilkan *report* adalah dengan perintah berikut:

```
Report Form cNamaReport Preview noconsole
```

Lihat perintah yang disimpan pada Tombol *Cetak* pada *procedure Click* di bawah ini.

```
Sele Daf_brg
Set filt to
_Screen.show
Report form C:\Npm\Report\Daf_brg preview noconsole
_Screen.hide
Acti wind form1
Sele daf_brg
Go top
```

Pertama menulis nama tabel yang yang anda ambil datanya, kemudian tampilkan seluruh data yaitu dengan perintah *Set filter to* setelah seluruh data ditampilkan anda harus menampilkan *Screen/layar* yaitu dengan perintah *_Screen.Show*, setelah itu tampilkan *reportnya* yang disimpan di direktori C:\Npm dan *folder Report*.

Waktu menutup *report* layar harus disembunyikan kembali yaitu dengan perintah *Screen.Hide* dan *form form_brg* harus diaktifkan kembali yaitu dengan perintah *Active Window Form1* berhubung properti *Name* dari *Form_brg* adalah *Form1* maka diperintahnya juga harus *Form1* kalau properti namanya dirubah misalnya menjadi *FormBarang* maka perintahnya juga menjadi *Active Window FormBarang*.

❑ Tombol Top (*Command5-Click*)

Tombol ini berfungsi untuk memindahkan *pointer* pada *record* paling atas pada tabel *Daf_brg*. Perintah yang ada dalam *procedure Click* pada tombol ini yaitu:

```
Sele daf_brg
Go top
Thisform.refresh
```

❑ Tombol Prior (*Command6-Click*)

Tombol ini berfungsi untuk memindahkan *pointer* satu *record* ke atas pada tabel *Daf_brg*. Perintah yang ada dalam *procedure click* pada tombol ini yaitu:

```
Sele daf_brg
Skip-1
if bof()
    Go top
endif
thisform.refresh
```

Skip-1 adalah perintah untuk memindahkan *pointer* satu *record* ke atasnya.

BOF() adalah singkatan dari *Before Of File* maka perintah di atas adalah memindahkan *pointer* ke atas sebanyak satu *record* dan jika sudah berada di *record* sebelum *file* maka *go top* atau tempatkan *pointer* di *record* paling atas.

❑ Tombol Next (*Command7-Click*)

Tombol ini berfungsi untuk memindahkan *pointer* satu *record* ke bawah pada tabel *Daf_brg*. Perintah yang ada dalam *procedure Click* pada tombol ini yaitu:

```
Sele daf_brg
Skip
if eof()
    Go bott
endif
thisform.refresh
```

Skip adalah perintah untuk memindahkan posisi *pointer* satu *record* ke bawah.


EOF() sudah dijelaskan pada bahasan sebelumnya maka perintah di atas adalah memindahkan *pointer* ke bawah sebanyak satu *record* dan jika sudah berada di akhir *record* maka *go bott* atau tempatkan *pointer* di *record* paling bawah.

❑ Tombol Bottom (*Command8-Click*)

Tombol ini berfungsi untuk memindahkan *pointer* ke *record* paling bawah pada tabel *Daf_brg*. Perintah yang ada dalam *procedure Click* pada tombol ini yaitu:

```
Sele daf_brg
Go Bottom
thisform.refresh
```

Setelah anda selesai mengisi seluruh program pada setiap objek, coba anda jalankan *Form* tersebut dan isilah datanya pada setiap *textbox* sesuai dengan judul masing-masing *textbox*. Cara untuk menjalankan *form* adalah:

1. Klik *icon Run*  yang bergambar tanda seru pada *toolbar standard* atau dapat pula dengan memilih menu utama *Form*, dan sub menu *Run Form*.

TIP

Skip adalah memindahkan *pointer* satu *record*, jika anda ingin memindahkan dua atau lebih maka anda tinggal menambahkan angka nominal sesuai dengan keinginan anda misalnya *Skip2* memindahkan *pointer* sebanyak 2 *record*.

PJ

2. Jika form anda belum direkam maka *Visual FoxPro* akan menampilkan kotak dialog yang menanyakan apakah *form* tersebut akan direkam ?, maka anda pilih *Yes*.

Gambar 4.36 Kotak dialog Save



TIP
 Untuk menjalankan form dapat juga dengan mengklik kanan form *designer* yang bersangkutan, maka akan muncul menu *pull-down* pilihlah sub menu *Run Form*, atau dengan cara mengetik *Do Form Daf_brg* pada *Command window*

3. Setelah *Form* tersebut jalan, maka anda isi data daftar barang dengan cara mengklik Tombol tambah dan mulailah mengisi kode, nama barang, satuan dan lain sebagainya.

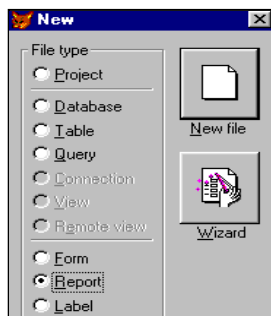
Gambar 4.37 Hasil akhir form persediaan barang



Membuat Report (Informasi) Daftar Barang

Langkah-langkah awal dalam pembuatan sebuah *Report* sama dengan pembuatan *Form* yaitu dengan menampilkan terlebih dahulu *Window New file* kemudian pilih option *Report* pada *File type* dan klik tombol *New file*.

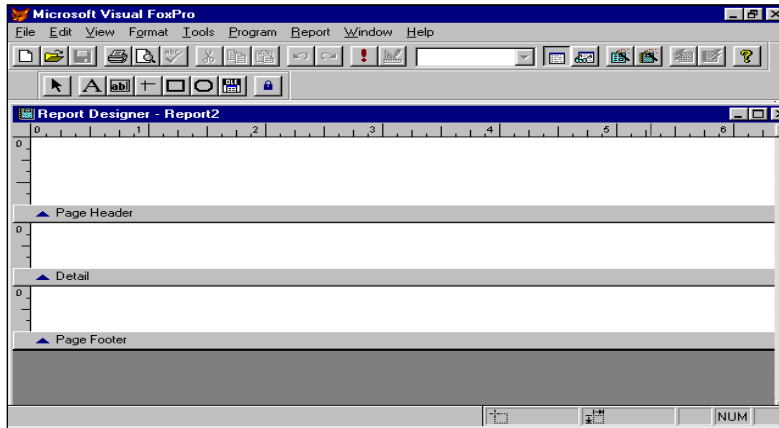
Gambar 4.38 Window new file



TIP
 Selain dengan langkah-langkah disamping anda juga dapat mengetik perintah *Create Report* pada *command window*

Setelah melakukan langkah-langkah di atas selanjutnya *Visual FoxPro* akan langsung menampilkan *window Report designer* seperti tampak pada gambar dibawah ini.

Gambar 4.39 Window report designer



Didalam mengerjakan sebuah *report* anda wajib mengetahui fasilitas-fasilitas pendukungnya, seperti halnya pembuatan *Form* dalam pembuatan *Report* juga ada *control-control* yang tergabung dalam sebuah *toolbar* yang bernama *Report controls toolbar*, *toolbar* ini berfungsi untuk membuat label, *field*, garis dan lain-lain.

BS

Gambar 4.40 Report controls toolbar




Sekilas uraian tentang *Report controls toolbar*:

- Select object** adalah tombol untuk memperbesar atau memperkecil juga untuk memilih dan memindahkan sebuah *control* yang telah anda masukan ke dalam sebuah *report*.
- Label** digunakan untuk membuat teks judul atau keterangan pelengkap *report designer*.
- Field** digunakan untuk menampilkan isi *field* dalam sebuah tabel, *memory variable*, atau menampilkan ekspresi-ekspresi lainnya. Khusus untuk *control* ini jika anda mengklik *control* ini dan menempatkannya pada area kerja *report*, maka akan langsung muncul *Window Report Expression* yang berfungsi untuk membuat ekspresi atau menyimpan nama *field* dari sebuah tabel.
- Line** adalah untuk menggambar sebuah garis vertikal atau horizontal.
- Rectangle** berfungsi untuk menggambar sebuah kotak segi empat.
- Rounded rectangle** berfungsi untuk menggambar sebuah lingkaran atau ellips.
- Picture/OLE Bound control** berfungsi untuk menampilkan sebuah gambar atau untuk menampilkan *field data general* dalam *report* anda.

TIP

Jika *report controls toolbar* belum ada maka untuk menampilkan *toolbar* tersebut adalah dengan cara mengklik menu utama *View* dan sub menu *Report control toolbar*.

BS

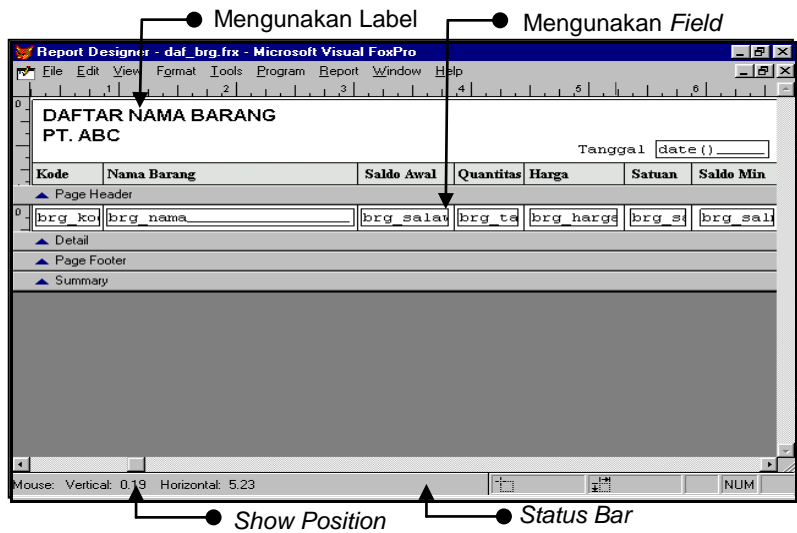
- ❑ **Button lock**  adalah tombol untuk mengunci sebuah tombol *control* yang sedang anda gunakan secara *continue*.

Untuk menempatkan sebuah objek misal *Label* pada *report designer* anda adalah sebagai berikut:

1. Klik *Icon Label* pada *toolbar* diatas.
2. Tempatkan *mouse pointer* anda pada tempat yang anda kehendaki dan klik. Maka sebuah objek akan siap untuk diedit.

Report designer di bawah ini akan memberi sedikit bayangan kepada anda bagaimana cara membuat sebuah *report* dan diharapkan anda dapat membuat *report* seperti di bawah ini tanpa harus menggunakan *Wizard*.

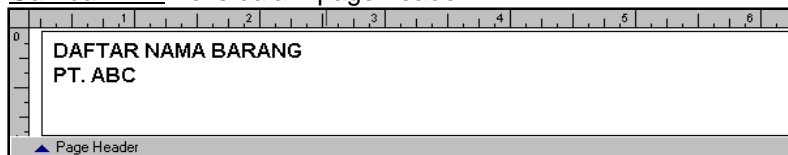
Gambar 4.41 Sebuah report jadi (report Daf_brg)



Membuat Teks pada Page Header

Tahap awal dalam pembuatan *report* Daf_brg dimulai dari *Page header* perhatikan gambar dibawah ini.

Gambar 4.42 Teks dalam page header



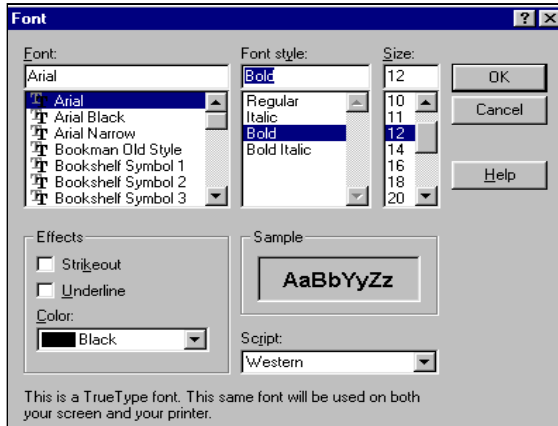
Page header adalah bagian paling atas dari setiap halaman *report* jadi daerah ini dijadikan tempat untuk menyimpan judul awal untuk setiap halaman *report* seperti yang anda lihat pada gambar diatas.

Untuk membuat tampilan teks seperti pada gambar diatas adalah dengan langkah-langkah sebagai berikut:

1. Klik icon Label **A** pada report control toolbar.
2. Tempatkan mouse pointer anda dilokasi report designer page header dan klik.
3. Ketiklah teks 'DAFTAR NAMA BARANG PT ABC' persis seperti gambar diatas.
4. Klik area kosong untuk keluar dari label tersebut.
5. Klik label 'DAFTAR NAMA BARANG PT ABC' tersebut.
6. Pilih menu utama *Format* dan sub menu *Font* maka akan muncul window font seperti pada gambar dibawah ini.

PJ

Gambar 4.43 Window font

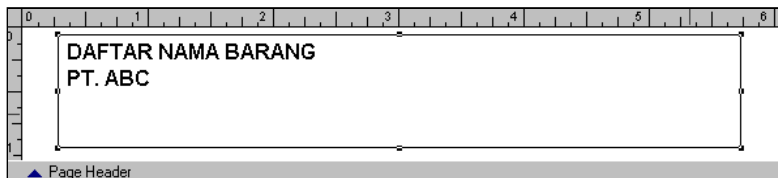


TIP
 Untuk mengedit atau memperbaiki teks di dalam sebuah *Label* adalah:
 1. Klik icon label pada report control toolbar
 2. Klik, tepat diatas label yang akan diperbaiki.


7. Isilah *Fonttype = Arial*, *fontsize = 12*, dan *fontstyle = Bold*. Klik tombol OK. Sehingga hasilnya akan terlihat seperti pada gambar 4.42 diatas.

Membuat Kotak pada Page Header

Gambar 4.44 Sebuah kotak pada page header



Buatlah kotak dengan posisi $Top=0.02$, $Left=0.26$, $Height=0.89$, dan $Width=6.35$ pada halaman *Page Header* dengan cara sebagai berikut:

1. Klik icon *Rectangle*  pada report controls toolbar.
2. Tempatkan pointer mouse diatas report designer page header
3. *Drag* (klik tahan geser) mouse anda ke kiri atau ke kanan untuk menentukan lebar (*Width*) dan ke bawah atau ke atas untuk menentukan tinggi (*Height*) jika sudah cukup lepas.
4. Untuk menentukan posisi *Top*, anda dapat menekan panah atas atau bawah dan panah kiri atau kanan untuk posisi *Left*

TIP

Untuk menampilkan *Show position bar* anda pilih menu utama *View* dan sub menu *Show position*.

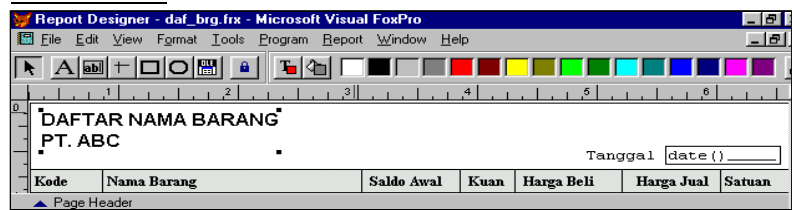
Untuk melihat posisinya anda lihat pada *Show position bar* pada bagian bawah *report designer*.

Object: Top: 0.02 Left: 0.26 Bottom: 0.91 Right: 6.60 Height: 0.89 Width: 6.35

5. Setelah *Top* dan *Left* sesuai, anda lihat *Height* dan *Width* jika belum sesuai maka sesuaikan.
6. Selain dengan menggunakan *mouse* untuk menentukan luas dari kotak tersebut adalah dengan menekan **SHIFT**+(panah kiri atau kanan) untuk *Width* dan **SHIFT**+(panah atas atau bawah) untuk *Height* sambil menyesuaikan ukurannya pada *show position bar*.

Membuat Judul Data

Gambar 4.45 Judul untuk data





Untuk membuat tampilan seperti pada gambar di atas adalah sebagai berikut:

1. Masih dalam *Page header*, buatlah teks-teks untuk judul dengan menggunakan *Label control* sesuai dengan gambar diatas, berdasarkan tabel dibawah ini.

Tabel 4.5 Daftar teks untuk judul data

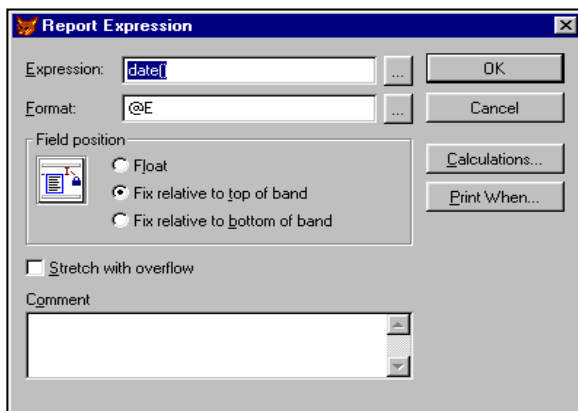
Nama teks	Posisi
Kode	Top=0.72, Left=0.30
Nama Barang	Top=0.72, Left=0.91
Saldo Awal	Top=0.72, Left=3.09
Kuantitas	Top=0.72, Left=3.89
Harga beli	Top=0.72, Left=4.39
Harga Jual	Top=0.72, Left=5.33
Satuan	Top=0.72, Left=6.06

2. Untuk semua teks di atas anda gunakan *font=Times New Roman, font style=Bold* dan *font size=9*.
3. Jika telah selesai membuat teks-teks tersebut di atas buatlah satu buah kotak dengan posisi *Top=0.28, Left=0.26, Height=0.23*, dan *Width=6.35*.
4. Berilah warna kotak tersebut dengan menampilkan *palette* warna terlebih dahulu caranya adalah pilih menu utama *View* dan sub menu *Color Palette toolbar*.
5. Karena yang akan diberi warna adalah objek kotak maka terlebih dahulu anda klik *icon Background color* pada *toolbar* tersebut, dan pilihlah warna yang anda kehendaki.

6. Batasi jarak antar teks dengan sebuah garis vertikal (*Height=0.21*) dengan cara, Klik *icon Line*  pada *report control toolbar*.
7. Tempatkan *mouse pointer* anda pada lokasi yang tepat. Dan *drag mouse* dari atas ke bawah.
8. Setelah semuanya diberi gari, Buatlah label 'Tanggal :' dengan posisi *Top=0.46, Left=4.82*.
9. Dan disebelah label 'Tanggal' tersebut tempatkan sebuah *Field* untuk menampilkan tanggalnya dengan cara:
 - Klik *icon Field*  pada *report control toolbar*.
 - Tempatkan *mouse pointer* anda disebelah label Tanggal dan Klik, maka akan muncul *window report expression* seperti dibawah ini.

PJ

Gambar 4.46 Window report expression



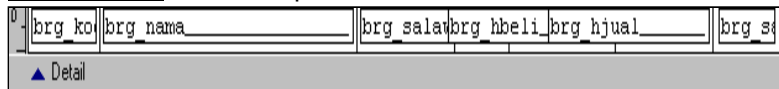
TIP
Control field pada *report designer* tidak hanya untuk menampilkan data dari sebuah *table* tetapi bisa juga untuk menampilkan sebuah ekspresi atau logika atau juga tampilan teks seperti halnya sebuah *label*.

- Untuk menampilkan tanggal hari ini dengan menggunakan *field*, adalah mengisi *Expression* dengan *Date()*. Lihat gambar di atas.

Anda telah selesai membuat *Header* untuk sebuah *report* Sampai sini anda sudah dapat menjalankan *Report* ini, yaitu dengan mengklik menu utama *View*, dan sub menu *Preview*.

Mengisi Kolom Details

Gambar 4.47 Field-field pada halaman detail



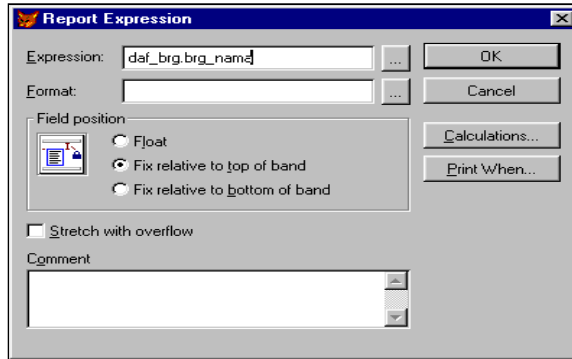
Selanjutnya anda harus mengisi halaman detail dengan *control-control field* yang diisi dengan nama-nama *field* dari tabel *Daf_brg*. Penggunaan *control field* ini adalah sebagai berikut:

1. Klik tombol *control field* pada *report control toolbar*,
2. Klik daerah yang akan menjadi lokasi dari *control* tersebut dan secara otomatis akan muncul kotak dialog *Report Expression*,

PJ

3. Ketiklah nama tabel dan nama *field* yang akan anda tampilkan pada baris *Expression* misalnya *Daf_brg.brg_Kode* lalu klik OK lihat gambar di bawah ini.

Gambar 4.48 Report expression



Sebagai panduan untuk *report* di atas buatlah *field-field* sesuai dengan judul pada *header* dengan daftar berikut ini.

Tabel 4.6 Daftar field untuk tiap judul

Kolom	Field diisi dengan
Kode	Daf_brg.brg_Kode
Nama	Daf_brg.brg_nama
Saldo awal	Daf_brg.brg_Salaw
Quantitas	Daf_brg.brg_Stock
Harga Beli	Daf_brg.brg_hbeli
Harga Jual	Daf_brg.brg_hjual
Satuan	Daf_brg.brg_sat

Anda juga dapat memberi format dengan mengklik tombol sebelah kanan dari baris format.

Tombol *calculation* pada kotak dialog di atas adalah untuk menghitung, mencari nilai maksimal atau minimal, nilai rata-rata, dan lain-lain yang berhubungan dengan perhitungan matematis untuk sebuah *field* dari sebuah tabel yang bertipe *numeric* dan biasanya *control field*nya harus disimpan pada lokasi *Summary*.

Sedangkan tombol *Print when* yaitu untuk memberikan ekspresi kapan sebuah *field* harus ditampilkan.

Susun rapi *field-field* tersebut dan buatlah kotak seperti kotak pada page *header* dan batasi jarak antar *field* dengan garis vertikal, posisi dari *field-field* dan garis tersebut harus sama dengan judulnya. Lihat gambar.

Gambar 4.49 Susunan field sejajar dengan judul / header

Kode	Nama Barang	Saldo Awal	Kuan	Harga Beli	Harga Jual	Satuan
▲ Page Header						
brg_ko	brg_nama	brg_salaw	brg_hbeli	brg_hjual	brg_sat	
▲ Detail						

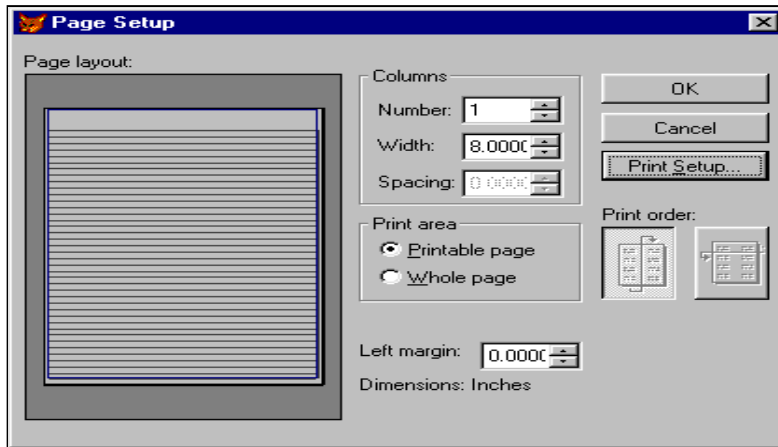
Menyeting Tipe Kertas

PJ

Langkah selanjutnya dalam pembuatan *report* ini adalah menyeting tipe kertas yang akan anda gunakan, caranya adalah sebagai berikut:

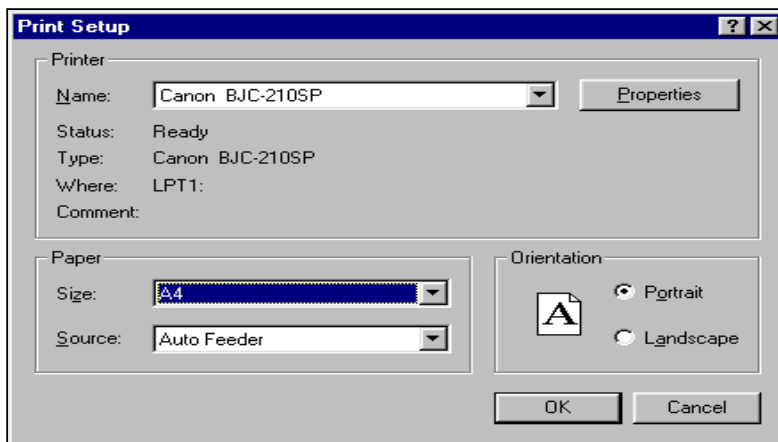
1. Pilih menu utama *File* kemudian klik submenu *Page Setup*.

Gambar 4.50 Kotak dialog page setup



2. Pada kotak dialog *Page setup*, klik tombol *Print Setup* maka akan muncul tampilan kedua yaitu kotak dialog *Print setup*,

Gambar 4.51 Kotak dialog print setup



3. Pada kolom *Paper* pilihlah ukuran kertas yang anda kehendaki misalnya kertas A4 dan *sourcenya* pilih *Auto feeder*, pada kolom *Orientation* pilihlah *portrait*, kalau anda ingin lebih sempurna dalam menyeting *report* ini anda dapat mengklik tombol *Properties* pada kotak dialog *print setup* maka akan muncul kotak dialog *properties* yang berhubungan dengan *properties printer* sesuai dengan nama *printer* yang anda pilih pada kotak dialog *Print setup* setelah selesai klik tombol OK.



Rekamiah *report* anda dengan nama *Daf_brg.frx* pada direktori *C:\Npm\Report* dengan cara mengklik menu utama *File* dan sub menu *Save* atau cukup mengklik *icon Save* pada *toolbar Standard*. Untuk melihat hasil tersebut anda jalankan terlebih dahulu *Form Form_brg* dan klik tombol *Cetak* pada form tersebut maka *Report Daf_brg* akan ditampilkan. Dan jika anda masih ber-ada dalam *Window Report designer* untuk melihat hasil akhir dari *report* ini anda tinggal mengklik menu *View* dan submenu *Preview* atau dengan mengklik mouse kanan pada lokasi kosong pada *report* anda maka akan muncul menu pilihan dan pilihlah *Preview* maka *report* yang anda buat akan ditampilkan seperti pada gambar di bawah ini.

Gambar 4.52 Hasil akhir report *Daf_brg*

DAFTAR NAMA BARANG PT. ABC						
Tanggal 13/02/2001						
Kode	Nama Barang	Saklo Awal	Kuan	Harga Beli	Harga Jual	Satuan
00001	Buku	0	5	12,000.00	12,000.00	Losi
00002	kursi	0	-7	4,500.00	1,235.00	Ff
00003	Kasur	0	-2	25,000.00	30,000.00	Pcs
00004	jfajfaf	0	0	0.00	0.00	

4.2.2 Membuat Program Penjualan

Di dalam suatu perusahaan biasanya ada transaksi penjualan dan pembelian di sini anda akan membuat suatu program untuk memudahkan perusahaan tersebut dalam melakukan kegiatan usahanya, dalam penyimpanan *data*, mencari *data*, atau penghapusan *data* dan lain sebagainya.

Membuat form penjualan

Seperti yang telah diuraikan di atas bahwa pembuatan *form* itu dimulai dengan mengetik perintah *Creat form* pada *command window* setelah itu akan muncul *Window form designer Form1*, anda rekam *form* baru tersebut dengan nama *Form_jual* pada direktori *C:\Npm\form*, *properties* yang digunakan untuk *form* baru ini antara lain sebagai berikut:

1. *BorderStyle=1-FixedSingle*,
2. *Caption = Faktur Jual*,
3. *Height = 256*,
4. *ShowWindow = 1-In Top-Level Form*,
5. *Width = 530*.

Selanjutnya anda isi *form* kosong tersebut dengan *control Label*, *Command button*, *Text box*, dan *Grid* yang ditata rapi seperti pada gambar di bawah ini.

Gambar 4.53 Form faktur jual (Form_jual)

Untuk *control-control* Label, *TextBox*, dan *Command Button* anda telah mempelajarinya pada form persediaan, dengan cara yang sama anda dapat menerapkannya disini. Tambahan materi untuk anda disini adalah bagaimana cara untuk memasukan dan memodifikasi *control Grid*, untuk itu lakukanlah tahap-tahap berikut ini.

1. Klik tombol *icon Grid* yang terdapat pada *Form control toolbar*
2. Tempatkan *pointer mouse* anda di lokasi yang anda kehendaki di atas *form* tersebut. Dan untuk mengolah *Grid* tersebut diatur di dalam *Properties*,
3. Untuk *Grid* di atas akan dibahas bagaimana cara memodifikasi *Grid* tersebut. Agar lebih mudah anda lihat saja daftar *properties* yang digunakan untuk *Grid* di atas pada tabel di bawah ini.

Tabel 4.7 Properties list kategori all untuk control Grid1

ColumnCount	7
Height	145
Left	0
Name	Grid1
Panel	1-Right(default)
RecordSource	BFak_jual
Top	55
Width	530

Keterangan :

- ColumnCount** adalah properti untuk menentukan jumlah kolom pada *Grid* yang anda buat.
- Height** untuk menentukan lebar dari *Grid* tersebut.
- Left** adalah untuk menentukan posisi *Grid* tersebut dihitung mulai dari sisi kiri *form*.
- RecordSource** adalah untuk menentukan nama tabel yang akan ditampilkan dalam *Grid* tersebut, berhubung tabelnya



belum dibuat jadi anda jangan dulu memasukkannya dan anda harus membuat terlebih dahulu tabel dengan nama *Fak_jual*.

- Top** adalah untuk menentukan posisi *Grid* tersebut dihitung mulai dari sisi atas *form*.
- Width** untuk menentukan panjang dari *Grid* tersebut

Dan masih banyak *properties* yang dapat digunakan untuk memodifikasi *control Grid* tersebut, sementara ini penjelasan diatas dianggap cukup.

Membuat Struktur Data Penjualan

Untuk Faktur penjualan akan digunakan dua buah tabel yang saling berhubungan (*Link*) tabel pertama disebut sebagai tabel Faktur (*Parent*) dan tabel kedua disebut baris Faktur (*Child*) untuk menghubungkan antara tabel Faktur dengan tabel baris Faktur anda gunakan perintah *Relation*. Sekarang anda akan membuat tabel yang pertama yaitu tabel Faktur dengan nama file *Fak_jual*. Dan untuk tabel baris Faktur diberi nama file *bFak_jual*.

Cara membuat tabel telah diuraikan pada pembicaraan sebelumnya. Sebagai panduan untuk anda lihat daftar nama *Field* untuk kedua tabel tersebut. Agar ada keselarasan antara contoh program yang ada dalam buku ini dengan contoh program yang anda buat sekarang.

Tabel 4.8 Daftar nama field untuk file *Fak_jual*

Name	Type	Width	Decimal	Index
Fak_kode	Character	5		Ya
Fak_tgl	Date	8		
Cus_nama	Character	20		
Cus_alamat	Character	20		
Tot_harga	Numeric	12	2	
Mark_	Character	1		

Tabel 4.9 Daftar nama field untuk file *bFak_jual*

Name	Type	Width	Decimal	Index
Fak_kode	Character	5		Ya
No	Numerik	4		
Brg_kode	Character	5		Ya
Brg_nama	Character	20		
Brg_satuan	Character	4		
Brg_harga	Numeric	10	2	
Sub_tot	Numeric	10	2	

Buatlah tabel sesuai dengan daftar di atas dan rekamlah tabel-tabel tersebut ke direktori *C:\Wpm\data*.

Untuk relasi diperlukan adanya *field index* dikedua tabel, *field* yang *diindex* adalah yang menjadi *Key/identifies* dari masing-masing tabel/*record*. Dengan adanya *Field index* pencarian data akan lebih cepat karena pencarian data dilakukan melalui *Field index* terlebih dahulu.

Memasukan Table ke Form dan membuat Relasi

PJ

Buka *form* Form_jual yang telah anda rekam tadi dan untuk memanggil *form* adalah dengan perintah *Modify form* pada *Command window* yang tentunya anda wajib menyebutkan nama direktorinya terlebih dahulu dengan perintah *Set default to*, agar lebih jelasnya lihat perintah di bawah ini.

```
Set defa to C:\Npm\Form
Modi form Form_jual
```

Atau dengan perintah,

```
Modi form C:\Npm\Form\Form_jual
```

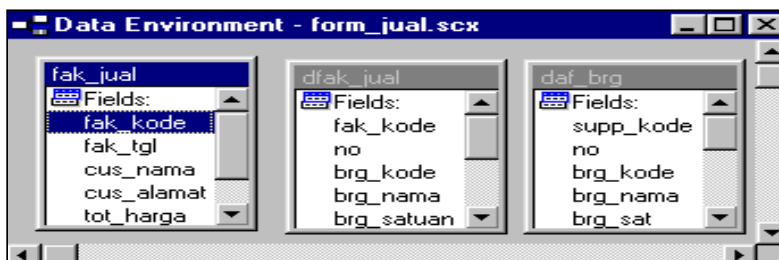
Setelah *form* Form_jual muncul pada *Window Visual FoxPro*, tahap awal untuk memodifikasi *form* tersebut adalah dengan memasukkan tabel yang diperlukan untuk *form* penjualan ke dalam *data environment*, tabel-tabel yang diperlukan dalam program penjualan yaitu:

1. Tabel **Daf_brg** sebagai daftar barang yang akan dijual,
2. Tabel **Fak_jual** yaitu tabel Faktur untuk menyimpan *data* nomor transaksi, tanggal dan pembeli atau Customer serta alamat Customer pada Faktur penjualan tersebut,
3. Tabel **bFak_jual** sebagai *file* baris Faktur yaitu tabel untuk menyimpan data dari item-item yang dijual, atau tabel untuk menyimpan data barang-barang yang dijual baik harga dan kuantitasnya.

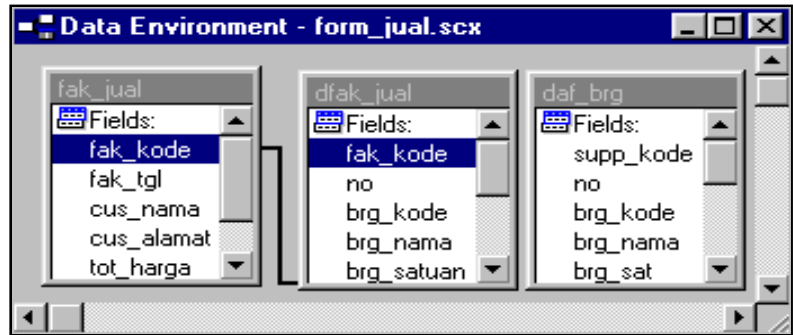
Setelah anda memasukkan ketiga tabel tersebut ke dalam *data environment*, sekarang anda tinggal menghubungkan tabel Fak_jual dengan tabel bFak_jual dengan cara:

1. Klik dan tahan *field index* yaitu *field Fak_kode* pada tabel Fak_jual
2. Kemudian geser dan masukan ke dalam tabel bFak_jual tepat di atas *field Fak_kode*. Kalau telah terhubung maka akan ada garis penghubung yang akan terlihat seperti gambar di bawah ini.

Gambar 4.54 Fak_jual dan bFak_jual belum terhubung



Gambar 4.55 Garis hitam penghubung antara Fak_jual dan bFak_jual (Garis relasi)



J

Setelah garis relasi terlihat berarti kedua tabel tersebut telah saling berhubungan satu sama lainnya, jika anda ingin memutus relasi tersebut anda tinggal mengklik garis tersebut dan tekan tombol *delete* pada *keyboard* anda. Tutuplah *Window data environment* dengan mengklik tombol *close* yang berada di sudut kanan atas *Window* tersebut.

Relation (relasi) akan menghubungkan dua atau lebih tabel sehingga *data* yang memiliki *index* dan nama *field* yang sama akan terhubung secara otomatis. Contoh anda memiliki *data* Faktur sebagai *Parent* dan *data* Baris Faktur sebagai *Child* dan kedua tabel *data* ini memiliki satu *field* penghubung yang sama yaitu *Fak_kode* *field* ini memiliki beberapa *record*, misalnya memiliki 4 *record* yang tiap *record* berbeda dan pada tabel baris Faktur memiliki banyak *record* yang pada sebagian *record* ada yang sama isinya. Agar lebih jelas lihat tabel di bawah ini.

Tabel 4.10 Tabel faktur (Parent)

Fak_Kode	Nama
01	Haris
02	Alex
03	Indra

Tabel 4.11 Tabel baris faktur (Child)

Fak_Kode	Barang
01	Buku
01	Pensil
02	Buku
02	Pensil
02	Kertas
03	Disket

Jika anda memilih 01 pada *field* *Fak_Kode* tabel Faktur maka pada tabel Baris Faktur yang akan ditampilkan hanya *data* yang memiliki *Fak_Kode* 01, jika 02 maka pada tabel Baris Faktur juga hanya ditampilkan yang berkode 02 saja. Jadi prinsip ini akan anda pakai untuk membuat transaksi penjualan dimana *file* Faktur yaitu *Fak_jual* yang mencatat nomor transaksi memiliki *file* *Child* yaitu *bFak_jual* yang mencatat barang yang dibeli customer.

Mengisi Program pada setiap Objek yang ada pada Form Form_jual.

J

Sebuah *form* kosong tentu tidak ada gunanya apa-apa tanpa diisi program dan perintah pada tiap *control* yang saling mendukung antara *control* yang satu dengan *control* lainnya. Program pada *form* Form_jual sebagai program Faktur penjualan tentunya berbeda dengan *form* sebelumnya yaitu *form* Form_brg. Program-program yang diperlukan untuk program penjualan lebih kompleks bila dibandingkan dengan program-program yang digunakan dalam program persediaan, selain itu tabel yang digunakannya pun lebih banyak.

Gambar 4.56 Form designer Form_jual

The screenshot shows a form titled "FAKTUR JUAL" with the following components:

- Fields for "Pelanggan:" (Text1), "Alamat:" (Text2), "Nomor:" (Text3), and "Tanggal:" (Text4).
- A table with columns: Np., Kode Brg, Nama Barang, Satuan, Jumlah, Harga, and Sub total.
- Fields for "Mencari Nomor Faktur:" (Text5) and "TOTAL:" (Text6).
- A checkbox labeled "Potong Saldo Persediaan".
- Buttons: "Tambah", "Hapus", "Cetak", and "Keluar".

Annotations in the image point to a "Check Box" (the checkbox) and a "Grid" (the table).

Berikut ini adalah uraian dari form di atas, dimulai dari objek yang paling besar yaitu Form1 yang menjadi dasar dari seluruh objek yang ada di atasnya.

□ Form1-Init

Pada Form1 tidak banyak disimpan perintah, perintah yang ada di form1 terletak pada *procedure init* (Form1-init) yang perintahnya adalah sebagai berikut:

PJ

```

Sele fak_jual
go top
sele bfak_jual
set filter to fak_kode=fak_jual.fak_kode
Thisform.refresh
    
```

Perintah ini adalah perintah untuk memfilter *data* yang berada pada tabel bFak_jual sebagai *file Child* yang memiliki Fak_kode sebagai *key*, difilter berdasarkan isi dari *field* Fak_kode yang ada pada *file* Fak_jual, pada saat pertama kali *Form* ini dijalankan.



Untuk keluar dari *Code window* yaitu dengan perintah CTRL+J atau dengan mengklik tombol *close* di sudut kanan atas *Window*.

❑ *Text1*

Di *Text1* tidak ada perintah apa-apa hanya dalam *properties kategori data ControlSource* diisi dengan *fak_jual.cus_nama*

❑ *Text2*

Di *Text2* juga tidak ada perintah apa-apa hanya dalam *properties kategori data ControlSource* diisi dengan *fak_jual.cus_alamat*

❑ *Text3*

Text3 dalam *properties kategori data ControlSource* diisi dengan *fak_jual.Fak_kode*

❑ *Text4*

Di *Text4* juga sama tidak ada perintah apa-apa hanya dalam *properties kategori data ControlSource* diisi dengan *fak_jual.Fak_tgl*

❑ *Text5*

Karena berhubungan dengan *Command5* tombol untuk menyimpan perintah mencari nomor transaksi atau untuk mencari *data* dari *field Fak_kode* maka perintah yang ada dalam *Text5-LostFocus* adalah:

Thisform.command5.setfocus

Perintah ini adalah perintah untuk memindahkan *pointer* ke *Command5* pada saat *pointer* meninggalkan *Text5* ini.

❑ *Text6*

Di *Text6* sama dengan *Text1* tidak ada perintah apa-apa hanya dalam *properties kategori data ControlSource* diisi dengan *fak_jual.Sub_tot*.

❑ *Command1-Click (Tombol Tambah)*

Properties yang digunakan untuk *Command1* ini adalah *Caption=⌋Tambah*, *FontBold=.T.*, *FontName=Arial* dan *FontSize=9*. Tombol ini adalah untuk menyimpan perintah untuk menambah atau membuat transaksi baru pada Faktur penjualan yang tentunya berhubungan dengan tabel *Fak_jual* dan *bFak_jual*. Kalau dalam form *Form_brg* (form barang) untuk mengisi kode diisi secara manual maka dalam form ini anda akan membuat program agar kode diisi secara otomatis. Perintahnya adalah sebagai berikut:



```

*Mengisi File Parent
Sele fak_jual
GO BOTT
xKode = val(fak_jual.fak_kode)+1
if xKode = 1
    Thisform.text3.Value = '00000'
endi
appe blank
Repl fak_jual.fak_Kode with Left((Thisform.text3.Value),;
    5-len(allt(str(xKode))))+allt(str(xKode))
Repl fak_tgl with date()
xrec = recno()
go xrec

*Mengisi File Child
sele bFak_jual
go bottom
append blank
repl no with 1,fak_kode with fak_jual.fak_kode
sele fak_jual
go bott
thisform.refresh
Thisform.text1.setfocus

```

Berikut uraian dari perintah diatas:

Tanda bintang yang tertera pada **Mengisi File Parent* adalah tanda agar kalimat tersebut tidak diproses atau bukan merupakan satu perintah, tanda (*) digunakan kalau anda akan memberi judul atau keterangan pada perintah yang anda gunakan.

Select fak_jual yaitu perintah untuk mengaktifkan sebuah tabel yang mempunyai nama file fak_jual

GO BOTT adalah kependekan dari *GO BOTTOM* karena di *Visual FoxPro* suatu perintah bisa disingkat menjadi empat huruf, dan perintah *Go bottom* ini adalah satu perintah untuk menempatkan *pointer* di record paling bawah pada tabel tersebut.

$xKode = val(fak_jual.fak_kode)+1$, *xkode* ini merupakan satu variabel yang mewakili $val(fak_jual.fak_Kode)+1$, *xkode* itu sendiri nilainya adalah numerik atau angka dan field *fak_kode* bertipe character maka agar field *fak_kode* bertipe numeric, ada satu perintah yang dapat merubah *type character* menjadi *numeric* yaitu perintah *Val* misalnya ada satu variabel $A="0001"$ kalau variabel *A* dirubah kedalam *numeric* maka perintahnya adalah $Val(A)$ dan anda simpan ke variabel *B* maka perintahnya menjadi $B=Val(A)$ dan nilai dari *B* adalah 1. anda bisa mempraktekannya di *Command window*.

Kemudian untuk perintah selanjutnya adalah perintah *IF..*
..ENDIF pada perintah,



```
If xKode = 1
    Thisform.Text3.Value = '00000'
Endif
```

Kalau perintah di atas diterjemahkan, maka artinya adalah sebagai berikut, Jika xKode bernilai 1 maka isilah *text3* dengan nilai "00000"

Append Blank adalah perintah untuk menambah satu *record* kebawah dalam sebuah tabel.

```
Repl fak_jual.fak_Kode with Left((Thisform.text3.Value),;
    5-len(allt(str(Xkode))))+allt(str(xKode))
```

Perintah *Repl* dari perintah di atas adalah kependekan dari *Replace* yang berarti mengisi field *fak_kode* dari tabel *fak_jual*, sedangkan *With* sama dengan *oleh* atau *dengan*.

Kemudian tanda titik koma (;) dalam *Visual FoxPro* adalah suatu alat penghubung yang menyatakan bahwa perintah yang di atas masih nyambung dengan perintah yang dibawah dalam artian masih satu baris.

Perintah *Left* adalah satu perintah untuk mengambil beberapa character mulai dari kiri ke kanan yang banyaknya ditentukan, contoh sederhananya sebagai berikut, misal anda ingin mengambil 3 huruf dari kiri ke kanan dari kata "FoxPro" yang disimpan ke variabel A (A="FoxPro"), maka perintahnya adalah *Left(A,3)* maka hasilnya adalah "Fox". anda bisa mencobanya pada *Command window*.

Len adalah suatu perintah untuk mengetahui panjang dari sebuah kata, misal variabel A="FoxPro" maka untuk mengetahui panjang kata atau jumlah huruf dari kata tersebut perintahnya adalah sebagai berikut *Len(A)* maka hasilnya adalah 6.

Allt adalah kependekan dari *Alltrim* adalah satu perintah untuk menghilangkan *space-space* yang tidak dibutuhkan dalam sebuah kata misal ada sebuah kata "Visual " sedangkan yang anda butuhkan hanya "Visual" maka untuk menghilangkan *spaceny*a adalah dengan perintah *Alltrim("Visual ")*.

Str adalah kebalikan dari *Val*, *Str* merupakan perintah untuk merubah type *numeric* menjadi string atau character misal A=1 kalau anda rubah ke string maka perintahnya adalah *Str(A)* dan hasilnya adalah "1".

Maka maksud dari perintah *Left* di atas itu mempunyai arti sebagai berikut:

```
Left((Thisform.Text3.Value),5-len(allt(str(Xkode))))+;
allt(str(Xkode))
```

Perintah ini artinya adalah mengambil kata dari nilai *Text3* mulai dari kiri ke kanan sebanyak 5 *character* dikurangi panjang dari *string* *xKode* kalau misal *Text3* bernilai "00000" dan *xKode* bernilai 1 maka hasil dari perintah tersebut adalah "0000" ditambah "1" maka nilainya adalah "00001"

Perintah *Repl fak_tgl with date()* adalah perintah untuk mengisi *field* *fak_tgl* pada tabel *fak_jual* dengan tanggal sekarang.

```
xrec = recno()
go xrec
```

xrec bukan merupakan perintah tetapi variabel pengganti dari *recno()*, dimana *recno()* merupakan satu perintah yang mencatat posisi record pada saat itu, kemudian *go xrec* merupakan perintah untuk menempatkan *pointer* pada posisi record yang telah di catat tadi pada sebuah tabel,

Perintah yang berada di bawah kalimat **Mengisi File Child* adalah perintah untuk mengisi tabel *bFak_jual* sebagai *child* dari *Fak_jual*, yang tentunya *field* *Fak_kode* harus diisi sama dengan *field* *Fak_kode* dari *file Parent* atau *Fak_jual*. Yang perintahnya tertera pada kalimat *repl no with 1,fak_kode with fak_jual.fak_kode field no* (nomor) diisi dengan 1 karena ini merupakan tahap awal dalam mengisi *file child*, dan *Fak_kode* diisi dengan *Fak_kode* yang ada di *file parent*. Perintah-perintah lain dari perintah di atas rasanya tidak perlu lagi dibahas karena sudah dibahas pada program sebelumnya. Untuk menambah *file child* nanti akan dibahas pada *control Grid1*.

❑ *Command2-Click (Tombol Hapus)*

Properties untuk tombol ini sama dengan *Command1* hanya berbeda dalam *caption* nya saja, Tombol ini diisi program yang nantinya dapat menghapus transaksi penjualan, yang namanya menghapus transaksi penjualan tentunya menghapus *parent* dengan *child*nya. Kalau anda menghapus nomor transaksi 00001 tentunya anda harus menghapus juga barang yang menjadi item dari nomor itu. Perintah yang disimpan di *Command2-Click* adalah sebagai berikut.

```
Sele fak_jual
kode=allt(fak_jual.fak_kode)
```

```
IF messagebox('Anda yakin akan menghapus '"+kode+;
"',36,'Hapus')=6
sele bFak_jual &&Menghapus Child
dele all for allt(fak_kode)=kode
sele fak_jual &&Menghapus parent
Delete
skip-1
if bof()
```



```

                                go top
                            endif
                        ENDIF
                    Thisform.refresh()

```

Kode pada perintah di atas merupakan variable pengganti dari *allt(fak_jual.fak_kode)*, variable ini nantinya bisa anda masukan ke dalam message seperti yang anda lihat pada perintah di atas.

Untuk ketentuan-ketentuan perintah *messagebox* silahkan anda baca bab Daftar Perintah pada bagian akhir buku ini. *Messagebox* ini fungsinya adalah untuk menampilkan kotak pesan pada perintah anda.

Tanda *&&* pada kalimat *sele bFak_jual &&Menghapus Child* adalah agar kalimat setelah tanda *&&* tidak diproses berbeda dengan tanda *** tanda *&&* disimpan satu baris setelah kalimat perintah aktif dan tanda *** disimpan pada awal kalimat agar kalimat tersebut tidak diproses.

Menghapus dua *data* pada tabel yang saling berhubungan tentunya tidak seperti menghapus *data* pada satu tabel, cara menghapusnya adalah dengan menghapus terlebih dahulu *data* yang berada pada *file Child* setelah itu baru menghapus *data* yang berada pada *file Parent*. Perintah untuk menghapus *data* yang berada pada *file Child* yaitu:

```

dele all for allt(fak_kode)=kode

```

Delete all adalah menghapus semua *Fak_kode* yang sama dengan *kode*. Dengan perintah itu maka semua *data* yang menjadi anak atau *child* dari *file Fak_jual* dan *field Fak_kode* akan terhapus, misal *Fak_kode='00001'* maka *field Fak_kode* pada *bFak_jual* yang bernomor '00001' akan dihapus. Dan untuk menghapus *Fak_kode* pada *Parent* cukup dengan perintah *Delete* saja karena hanya satu *record*.

□ Command3-Click (Tombol Cetak)

Perintah yang terdapat dalam *Command3-click* adalah sebagai berikut:

```

Sele Fak_jual
rek=recno()
Sele bFak_jual
Set filter to fak_kode=Fak_jual.fak_Kode
Go top
_screen.show
report form c:\Npm\report\fak_jual preview noconsole
_screen.hide
acti wind form1
sele Fak_jual
go rek

```



Perintah *set filter to* pada perintah di atas adalah untuk memfilter *Fak_kode* yang sama dengan *Fak_kode* yang ada di *File Parent* atau *Fak_jual*. Perintah *Preview pada* perintah *report form c:\Npm\report\fak_jual preview noconsole* adalah untuk menampilkan *report* ke atas layar kalau ingin langsung masuk ke *printer* anda tinggal mengganti *Priview* dengan *To Print*. Untuk perintah-perintah yang lainnya telah dibahas dalam pemrograman *Persediaan Barang*. Untuk pembuatan *report*-nya nanti akan dibahas pada sub judul *Membuat Report*.

❑ *Command4-Click (Tombol Keluar)*

Perintah untuk keluar dari aplikasi yang dijalankan adalah perintah *Thisform.Release*

❑ *Command5-Click (Mencari Nomor Faktur)*

Perintah yang ada di *Command5* ini berhubungan dengan nilai dari *Text5*, *Text5* merupakan tempat untuk mengetikkan nomor transaksi yang akan anda cari kemudian setelah diketik anda harus mengklik tombol disebelahnya (*Command5*), Perintahnya adalah sebagai berikut:

```
public cari
cari=allt(Thisform.text5.value)

If Empty(cari)
Return
Else
Sele fak_jual
rek=recno()
Locate for allt(fak_kode)=cari
If !found()
go rek
Endif

Endif

Thisform.refresh
rele cari
```

Perintah *Public* pada perintah *public cari* adalah perintah untuk mengenalkan variable *cari* yang nantinya dicatat oleh memory. *IF Empty(Cari)* artinya adalah jika variable *cari* kosong maka *Return* atau perintah yang ada di bawahnya jangan diproses, *Else* jika tidak maka masuk ke perintah di bawahnya. Variable *rek* pada perintah *rek=recno()* adalah variable untuk mencatat posisi *record* pada saat itu, kemudian *Locate for allt(fak_kode)=cari*, artinya adalah menempatkan *pointer* pada posisi *record* untuk nilai dari *Fak_kode* yang sama dengan nilai *cari*, tanda ! (seru) pada perintah *If !found()* mengandung arti sebaliknya, jadi perintah itu mempunyai arti jika tidak ditemukan maka *go rek*, artinya kembali pada posisi *record* yang telah dicatat tadi.



□ Grid1

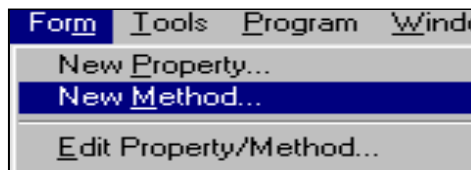
TIP
 Untuk menentukan lebar *Column* pada sebuah *Grid* dapat pula dengan cara mengklik *column* tersebut kemudian geser dengan menekan SHIFT + (Panah kiri/kanan) pada *keyboard* anda

Grid1 ini adalah objek untuk menambah, menghapus, dan menampilkan *data* untuk tabel *bFak_jual* sebagai *file child* dari *Fak_jual*, *Grid1* ini merupakan *parent* dari kolom dan baris. Yang nantinya kolom ini (*Column*) bertindak sebagai *Field* dari *data* *bFak_jual* dan baris bertindak sebagai *record* dari *file* yang bersangkutan. Setiap *Column* ini dapat ditentukan lebarnya di *properties* atau dengan mengklik sisi kanan dari *header column* tersebut tahan dan geser sesuai keinginan, *Column* dari *Grid* ini mempunyai *Header* dengan nama *Header1* yang berfungsi sebagai judul untuk *Column* tersebut cara memberinya judul yaitu dengan memilih *Caption* pada *Properties* dan *Column* harus diisi dengan nama *field* dari *file* *bFak_jual*, di dalam *Column* ini telah tersedia *TextBox* sebagai objek untuk menampilkan *data* dan *Column* ini juga dapat diisi dengan *control* lain selain *TextBox*, misalnya saja dapat diisi dengan *ComboBox*, *CommandButton*, atau yang lainnya.

Method yang digunakan dalam *Grid1* ini adalah *Right Click Event* (*procedure* untuk menyimpan perintah pada mouse tombol kanan) yang diisi dengan perintah untuk menambah dan menghapus *file child* pada *Grid1* ini. Karena nantinya perintah ini disimpan pada *Text* yang ada di setiap *Column* pada *Grid1* maka agar tidak terjadi pengulangan pencatatan program yang panjang, anda buat *Method* baru misalnya dengan nama *dHapus* dengan cara sebagai berikut:

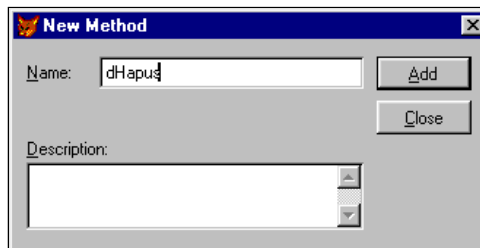
- Pilihlah menu utama *Form* kemudian pilih dan klik submenu *New Method* lihat gambar.

Gambar 4.57 Menu New Method



- Maka akan muncul kotak dialog *New Method*, Ketiklah nama dari *Method* tersebut pada baris *Name*, anda ketik saja *dHapus*. Lalu klik tombol *Add* atau tekan *ENTER*.

Gambar 4.58 Kotak dialog new method



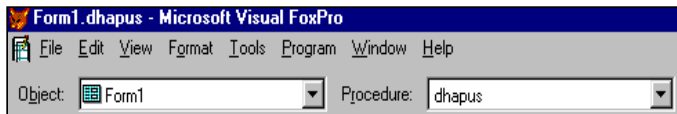
- *New Method* telah selesai dibuat klik tombol *Close* untuk keluar dari kotak dialog tersebut.



Mengisi program dHapus

Untuk mengisi perintah pada *Method* baru dHapus adalah sebagai berikut, Klik dua kali pada *Form Form_jual* maka akan masuk ke *Code Window Form1*, pada baris *Procedure list* di dalam *Code Window* pilihlah dHapus.

Gambar 4.59 Code window-procedure dHapus



Perintahnya adalah:

```

Y=0
DEFINE POPUP Tam FROM MROW(),MCOL();
SHORTCUT IN WINDOW Form1
DEFINE BAR 1 OF Tam PROMPT '\<Tambah ?'
DEFINE BAR 2 OF Tam PROMPT '\<Hapus ?'
ON SELECTION POPUP Tam Y = BAR()
ACTIVATE POPUP Tam
IF Y > 0
    IF Y = 1
        sele bFak_jual
        calculate cnt(no) to a
        go bottom
        append blank
        Repl no with a+1,fak_kode ;
        with fak_jual.fak_kode
        set filt to fak_kode=fak_jual.fak_kode
        This.Refresh
    ELSE
        sele bFak_jual
        delete
        go top
        i=0
        scan
            i=i+1
            repl no with i
        endscan
    ENDIF
ENDIF
    
```

TIP
 Pada *Grid* ada properties yang namanya *DeleteMark* yang berfungsi untuk menghilangkan kolom untuk menandai penghapusan sebuah *record*, yang kolomnya terletak di sisi kiri *grid* tersebut.

Perintah yang menggunakan huruf kapital semua pada perintah di atas adalah perintah untuk membuat menu *Pulldown* pada aplikasi anda. Yang mempunyai dua sub menu yaitu *Tambah* dan *Hapus*, pada menu *Tambah* yang diwakili dengan *Y=1* diisi dengan perintah untuk menambah *data* pada tabel *bFak_jual*.

PJ

Perintah *calculate cnt(no) to a* adalah perintah untuk menghitung jumlah *record* pada tabel *bFak_jual*, perintah lainnya sudah dibahas pada modul sebelumnya. Dan yang berada pada daerah *ELSE* sama dengan *Y=2*, diisi dengan perintah untuk menghapus *data*, setelah perintah menghapus ada perintah *Scan..Endscan*.

```
i=0
scan
    i=i+1
    repl no with i
endscan
```

Perintah tersebut adalah untuk mengisi *field no* pada *file bFak_jual* tetap urut, misalnya jika pada *bFak_jual* tersebut ada 4 nomor berarti urutannya 1,2,3 dan 4 jika anda hapus nomor 2 jadi urutannya menjadi 1,3, dan 4 agar urutannya menjadi 1,2 dan 3 maka gunakan perintah tersebut.

- *Grid1-RightClick*

Jadi perintah yang disimpan di *Grid1-RightClick* adalah bukan lagi perintah di atas yang panjang, tetapi hanya memakai perintah sebagai berikut:

```
Thisform.dHapus
```

Perintah ini jangan hanya disimpan di *Grid1* saja tetapi simpan juga di setiap *Text* dalam kolom pada *Grid1* ini. Karena perintah yang disimpan di *Grid*, tidak bisa digunakan di *Grid* yang kolomnya telah terisi. Lihat gambar.

Gambar 4.60 *Grid1* dengan menu pull-down

TIP	
Kolom paling kiri pada sebuah <i>Grid</i> adalah kolom <i>RecordMark</i> jika anda ingin menghilangkan kolom tersebut maka anda harus memberi nilai <i>.F.</i> (<i>False</i>) pada <i>properties RecordMark</i> tersebut	

No.	Kode Brg	Nama Barang	Satuan	Jumlah	Harga	Sub total
1	00001	Buku	Losi	5	12000.00	60000.00
2	00002	kursi	Ff	7	1235.00	8645.00
3	00003	Kasur	Pcs	2	30000.00	60000.00
4						0.00

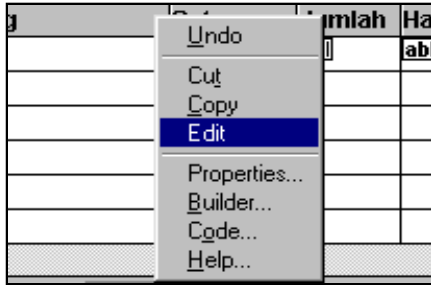
Jika perintah seperti pada perintah di *Grid1* tidak dipasang pada kolom-kolom dalam *Grid* ini maka menu *pull down* seperti yang terlihat pada gambar di atas tidak dapat dikeluarkan pada posisi *Grid* yang telah terisi yaitu pada posisi nomor 1,2,3 dan 4.

Untuk mengisi program di setiap kolom pada *Grid1* anda harus mengaktifkan terlebih dahulu kolom (*Column*) tersebut. Cara mengaktifkan *Column* pada *Grid1* tersebut ada beberapa langkah, langkah-langkahnya adalah sebagai berikut:

1. Klik objek *Grid* tersebut, kemudian klik mouse kanan pada daerah *Grid* tersebut, maka akan muncul menu seperti pada gambar di bawah ini.

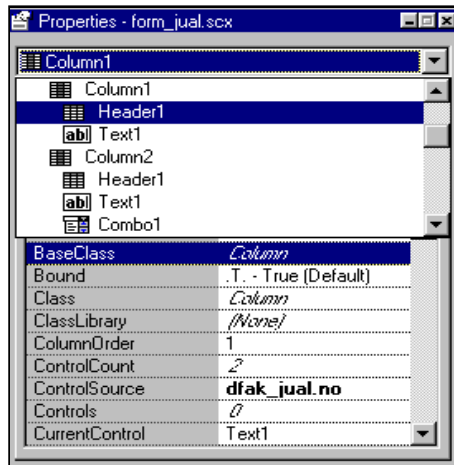


Gambar 4.61 Mengedit Grid1



2. Pilih dan klik *Edit*, maka *Column* pada *Grid* tersebut dapat anda edit sesuai dengan yang anda inginkan, baik itu memberi judul *header* memasukan nama *field* atau memasukan perintah.
3. Ada juga cara yang lebih ringkas tanpa harus mengklik mouse kanan dan memilih edit, tetapi cukup memilih objek mana yang akan anda aktifkan pada *Properties Object*. Lihat gambar di bawah ini.

Gambar 4.62 Properties-Form_jual



Setelah anda mengaktifkan *Grid1* dan dapat memilih setiap *Column*, maka *header* pun dapat anda pilih.

1. Grid1.Column1

Pilihlah *header1* sebagai judul dari *Column1* dan pada *Properties* isilah *Caption* pada *setting box* dengan **No**, agar hurufnya tebal maka isilah *FontBold=.T.-True* dengan cara mengkliknya dua kali atau dapat dipilih di *setting box* dan *Width=26*. Dan untuk

TIP
Header pada sebuah <i>Grid</i> bisa anda hilangkan, cara menghilangkan <i>Header</i> adalah dengan memberi nilai 0 (nol) pada <i>properties HeaderHeight</i> .

PJ

mengisi *Column* dengan nama *field* yaitu dengan cara klik terlebih dahulu *Columnnya (column1)* sampai pada *Properties* object menunjukkan *Column1*, kemudian pilihlah tab *Data* dan pada *Properties list* pilih *ControlSource* pilihlah *bFak_jual.no* pada *setting box*.

Untuk mengisi perintah pada *Column1.Text1-RightClick* ini yaitu dengan cara mengklik dua kali (*Double Click*) *Column1* tersebut maka akan masuk ke dalam *Code Window* pilihlah *Object Text1* dan prosedur *RightClick* kemudian pada *editing region* ketiklah perintah yang sama seperti pada *Grid1-Right Click* yaitu perintah: *Thisform.dHapus*.

Pasanglah perintah ini di semua *Text1* yang ada dalam *Column* pada *Grid1* ini mulai dari *Column1* sampai *Column7*. Kecuali *Column2* karena *Column2 Text1* diganti dengan *Combo1*

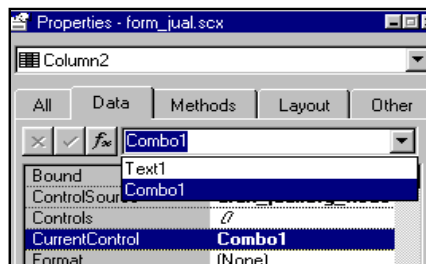
2. Grid1.Column2

Lakukanlah hal yang sama dengan apa yang dilakukan pada *Column1*, tetapi *Caption* untuk *Header* diisi dengan **Kode_brg** dan *Width=59*, dan *ControlSource* untuk *Column2* yaitu *bFak_jual.Brg_kode*.

Karena nantinya *Column2* ini dipakai untuk menampilkan daftar barang, maka di dalam *Column2* ini harus dipasang *ComboBox* dengan cara:

1. Setelah *Column2* ini aktif, pilihlah *control Combo Box* pada *Form controls toolbar*
2. Klik tepat di atas daerah *Column2* maka *Combo Box* tadi telah masuk ke dalam *Column2*.
3. *Column2* ini masih memakai *Text1* untuk mengganti *Text1* dengan *Combo1* caranya adalah dengan mengklik *Properties list CurrentControl* dan pada *setting box* pilihlah *Combo1*, maka yang akan aktif adalah *Combo1*, lihat gambar.

Gambar 4.63 Properties list current control



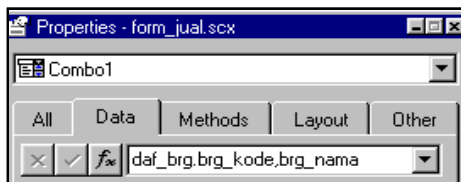
Sekarang anda harus mengisi *Combo1* tersebut dengan *data* barang yang ada pada tabel *Daf_brg* tentunya yang ditampilkan dalam *Combo1* ini hanya *Kode* dan nama barangnya saja. Untuk menampilkan dua *field* tersebut anda harus memodifikasi terlebih

dahulu *Combo1* tersebut, untuk memodifikasi *Combo* tersebut adalah sebagai berikut:

PJ

- Klik *Column2* yang menjadi tempat untuk menyimpan *Combo1* tersebut.
- Di *Properties Object* yang tadinya *Column2* pindahkan ke objek *Combo1*, pilih *tab All*.
- Pada *Properties list* isi *ColumnCount* dengan 2, dan *ColumnWidth* dengan 50,100 agar *Combo1* tersebut menjadi datar atau berbentuk *Flat* anda bisa mengganti *properties SpecialEffect* yang tadinya 0-3D menjadi 1-Plain.
- Setelah semuanya selesai pindahkan tab *properties* anda ke tab *Data*.
- Pilih *properties list RowSource* dan pilih *Daf_brg.brg_Kode* pada *Setting Box*, setelah dipilih anda tambahkan pada *setting box* tersebut ,*brg_nama* lihat gambar.

Gambar 4.64 Properties Grid1.Column2.Combo1



- Pilih *properties list RowSourceType* isi dengan 6-*Field* dengan memilihnya pada *Setting box*.

Mengisi perintah pada objek *Grid1.Column2.Combo1*

- *Combo1-RightClick*
Dengan mengklik dua kali *Column2* tersebut maka akan masuk ke *Code Window*, kemudian setelah masuk di *Code Window* pilihlah *Combo1* pada *Object list*, dan pada *Procedure list* pilihlah *Right Click*, dan pada *editing region* ketiklah perintah berikut ini.

Thisform.dHapus

- *Combo1-InteractiveChange*
Karena berada pada *Object* yang sama maka anda tinggal memindahkan *procedure* ke *InteractiveChange*, dan ketiklah perintah di bawah ini:

```
This.value = allt(daf_brg.brg_kode)
Sele bFak_jual
Repl brg_kode with daf_brg.brg_kode,;
    brg_nama with daf_brg.brg_nama,;
    brg_harga with daf_brg.brg_harga,;
```

TIP

Pada sebuah *Combo-Box* anda dapat menampilkan dua atau lebih *field* cara menambahkan *field* pada sebuah *Combo* adalah dengan tanda koma (,) dan tentunya kolom dalam *Combo* tersebut juga harus disesuaikan dengan banyak *field* yang anda masukan.



```
brg_satuan with daf_brg.brg_sat
Thisform.refresh
```

Procedure InteractiveChange adalah salah satu *method* untuk menyimpan satu perintah dan perintah tersebut akan dikerjakan pada saat nilai dari suatu objek berubah. Dan perintah di atas adalah perintah untuk mengisi *field-field* dari tabel *bFak_jual*.

3. Grid1.Column3

Lakukanlah hal yang sama dengan apa yang dilakukan pada *Column1*, tetapi *Caption* untuk *Header* diisi dengan **Nama Barang**, *Width=150*, dan *ControlSource* untuk *Column3* *bFak_jual.Brg_nama*.

4. Grid1.Column4

Sama dengan apa yang dilakukan pada *Column1*, yang berbeda hanya dalam mengisi *Caption* untuk *Header* yaitu diisi dengan **Satuan**, *Width=60*, dan *ControlSource* untuk *Column4* diisi dengan *bFak_jual.Brg_Satuan*.

5. Grid1.Column5

Untuk *Header* diisi dengan **Jumlah**, *Width=49*, dan *ControlSource* diisi dengan *bFak_jual.Jumlah*, selain di *RightClick*, *procedure* yang digunakan untuk *Text1* pada *Column5* ini yaitu *Procedure: Valid* yang diisi dengan perintah berikut ini:

```
Grid1.Column5-Valid
```

```
Sele bFak_jual
a=brg_harga*This.value
Repl sub_tot with a
rek=recno()
Go top
Calculate sum(sub_tot) to b
Thisform.text6.value=b
Go rek
Thisform.refresh
```

Maksud dari perintah di atas adalah untuk mengalikan harga dengan jumlah dan mengisikannya ke dalam *field* *bFak_jual.Sub_tot*, dan untuk perintah *Calculate sum(sub_tot) to b* adalah menjumlahkan *Sub_tot* dan mengisikannya ke dalam *Text6*.

6. Grid1.Column6

Tidak berbeda dengan *Column1*, *Column* ini juga memiliki perintah yang di simpan seperti pada *Column1*

tetapi *Caption Header* ini adalah **Harga**, *Width*=63, dan untuk *ControlSourcenya* diisi dengan `bFak_jual.Brg_harga`.



7. Grid1.Column7

Untuk *Column7* juga Tidak berbeda dengan *Column1*, *Column* ini juga memiliki perintah yang di simpan seperti pada *Column1* tetapi *Caption Header* ini adalah **Sub total**, *Width*=78, dan untuk *ControlSourcenya* diisi dengan `bFak_jual.Sub_tot`.

□ *Check1 (Potong Saldo Persediaan)*

Cara memasukan *CheckBox* ke dalam *Form* sama dengan memasukan *control-control* lain, *CheckBox* ini dapat diberi *caption* seperti halnya *Command button*, anda beri *caption* untuk *Check box* ini dengan kata Potong Stock, *Check Box* hanya memiliki dua nilai pilihan yaitu 1 atau 0, jika nilai dari *Checkbox* ini 1 maka dia akan menampilkan tanda (✓) pada kotak *Check box* tersebut dan jika nilainya 0 dia tidak akan menampilkan tanda apa-apa. Perintah yang terdapat pada *Check box* ini adalah sebagai berikut:

TIP
<i>Check Box</i> ini juga dapat dimodifikasi menjadi sebuah tombol yang akan terbenam (<i>Sunken</i>) jika bernilai 1, dan akan timbul (<i>Raised</i>) lagi jika bernilai 0, caranya agar sebuah <i>Check box</i> menjadi sebuah tombol yaitu dengan memilih <i>Style</i> pada <i>properties</i> dan pilih <i>1-Graphical</i> pada <i>setting box properties</i>

1. Check1-Click

```
Sele Fak_jual
rek=recno()
```

```
IF This.Value=1
    sele Fak_jual
    Repl Mark_ with '1'
    Sele bFak_jual
    Set filt to allt(Fak_kode)=allt(Fak_jual.Fak_kode)
```

```
Scan
    If Seek(Brg_kode,'Daf_brg','Brg_kode')
        Sele Daf_brg
        Repl Brg_Stock with;
        Brg_Stock-(bFak_jual.Jumlah);
        Brg_Kurang with;
        Brg_Kurang+(bFak_jual.jumlah);
        Brg_Hjual with bFak_jual.Brg_Harga
```

```
Endif
Endscan
```

```
ELSE
    sele Fak_jual
    Repl Mark_ with '0'
    Sele bFak_jual
    Set filt to allt(Fak_kode)=allt(Fak_jual.Fak_kode)
    Scan
        If Seek(Brg_kode,'Daf_brg','Brg_kode')
            sele Daf_brg
            Repl Brg_Stock with;
```

PJ

```

    Brg_Stock+(bFak_jual.Jumlah);
    Brg_Kurang with;
    Brg_Kurang-(bFak_jual.jumlah);
    Brg_Hjual with bFak_jual.Brg_Harga
  Endif
Endscan
ENDIF
Sele Fak_jual
Go rek
Thisform.Refresh

```

Perintah di atas memiliki arti sebagai berikut, jika Check1 bernilai 1 atau bertanda Ceklis maka isilah *field* Mark_ pada tabel Fak_jual dengan '1' (satucharacter), *field* mark ini nantinya berfungsi sebagai tanda bahwa transaksi faktur yang bersangkutan sudah di Cek atau sudah dipotong stocknya.

Perintah Scan...EndScan adalah perintah untuk membaca seluruh *record* dari awal sampai akhir *record* yang di dalamnya terdapat juga perintah Seek yang fungsinya untuk mencari *record* dari sebuah *field* yang mempunyai nilai yang sama. Jadi perintah *If Seek(Brg_kode,'Daf_brg','Brg_kode')* adalah mencari nilai Brg_kode dari tabel bFak_jual yang sama dengan nilai Brg_kode pada tabel Daf_brg, jika di ketemukan maka akan memproses perintah dibawahnya, yaitu perintah *Repl Brg_Stock with Brg_Stock-(bFak_jual.Jumlah)* yaitu perintah untuk mengisi Brg_stock dengan Brg_stock itu sendiri dikurangi dengan Jumlah dari bFak_jual, sehingga Stock dari barang tersebut menjadi berkurang.

Dan perintah yang berada pada kolom *Else* berarti perintah untuk nilai *Check1=0*, perintahnya adalah sebaliknya dari perintah yang telah dibahas tadi atau dikembalikan lagi pada kondisi sebelumnya, dan tentunya anda pun dapat menerjemahkan perintah tersebut.

2. Check1-Refresh

Perintah yang terdapat dalam objek *Check1* dan *procedure Refresh* adalah sebagai berikut:

```

Sele Fak_jual
If mark_='1'
  This.value=1
Else
  This.value=0
Endif

```

Menyimpan perintah pada *procedur* atau *methode Refresh* akan dikerjakan pada saat sebuah *Form* merefresh maka seluruh objek yang ada di dalam *form* tersebut ikut direfresh jadi anda tidak perlu lagi mengerjakan klik atau apa saja pada objek yang bersangkutan.



Seperti yang telah dibahas di atas bahwa *field* *Mark_* pada tabel *Fak_jual* adalah untuk memberi tanda pada sebuah transaksi bahwa transaksi / Faktur tersebut telah dikurangi stocknya sesuai dengan jumlah barang yang dibeli Customer. Jadi perintah di atas memiliki arti jika *field* *Mark_* bernilai '1' maka *Check1* bernilai 1, jika tidak maka *Check1* bernilai 0.

Setelah anda selesai mengisi seluruh objek dengan program maka coba anda jalankan *Form* tersebut dan isilah dengan beberapa data penjualan, seperti pada gambar di bawah ini.

Gambar 4.65 Form *Form_jual*, program faktur penjualan

No.	Kode Brg	Nama Barang	Satuan	Jumlah	Harga	Sub total
1	00001	Buku	Losi	5	12000.00	60000.00
2	00002	kursi	Ff	7	1235.00	8645.00
3	00003	Kasur	Pcs	2	30000.00	60000.00
4						

Mencari Nomor Faktur: TOTAL : 128645.00

Potong Saldo Persediaan

Tambah Hapus Cetak Keluar

Membuat Report

Seperti yang telah dibahas pada *Command3* bahwa nama dari *report* untuk Faktur penjualan adalah *Fak_jual*, cara-cara pembuatan *report* telah anda ketahui pada program sebelumnya, yaitu dengan mengetik perintah *Create Report* pada *Command window* atau karena nama dari *report* ini telah ditentukan maka anda bisa mengetik perintah di bawah ini:

Modify Report C:\Npm\Report\Fak_jual

Maka *report* *Fak_jual* ini akan langsung disimpan pada direktori *C:\Npm\Report*, setelah itu akan muncul *Window report designer* *Fak_jual* yang masih kosong.

Membuat Judul Halaman Report

Gambar 4.66 Page header *Fak_jual*

FAKTUR PENJUALAN
PT. ABC

Page Header

PJ

Tentu sudah terbayang oleh anda bagaimana membuat tampilan seperti di atas karena tampilan seperti itu sudah dibahas dalam pembuatan *report* Persediaan Barang, tetapi untuk mengingatkan kembali, ada baiknya jika cara pembuatan tampilan di atas diuraikan lagi:

1. Klik *icon Label* pada *report control toolbar*
2. Klik pada lokasi *report designer* dan ketiklah teks “FAKTUR PENJUALAN”
3. Tekan *ENTER* dan ketik lagi teks “PT. ABC”
4. Untuk teks ini gunakan *font=Arial*, *font size=12* dan *font style =Bold*.
5. Buatlah kotak dengan posisi *Height=1.34*, *Width=6.13*, *Top =0,26*, *Left=0,26* dan *Bottom=1,60* pada daerah *page header*.

Menampilkan Data Faktur pada Page Header

Sebagaimana layaknya sebuah Faktur, dalam pembuatan *report* ini juga anda harus menampilkan data Pembeli atau *Customer*, Alamat pembeli, Nomor Faktur beserta tanggal Faktur yang *data*-nya mengambil dari *data* Faktur penjualan yaitu *Fak_jual*.

Gambar 4.67 Tampilan field data *Fak_jual* (Parent)

The screenshot shows a report header with the following content:

FAKTUR PENJUALAN
PT. ABC

Nama Pelanggan :

Alamat Pelanggan :

Nomor Faktur :

Tanggal Faktur :

▲ Page Header

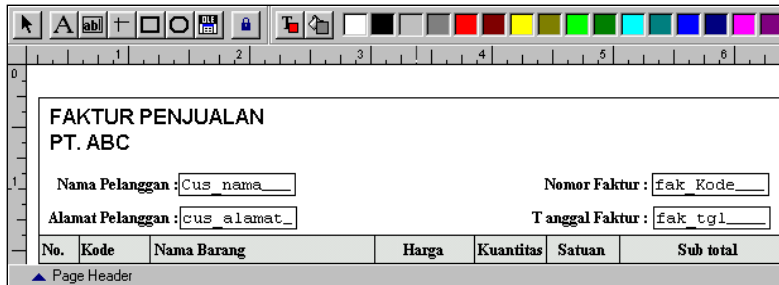
Langkah-langkah untuk membuat tampilan di atas adalah sebagai berikut:

1. Buatlah teks-teks dibawah ini dengan menggunakan *Font= Times New Roman*, *Font style=Bold* dan *Size=9*.
 - Nama Pembeli :
 - Alamat Pembeli :
 - Nomor Faktur :
 - Tanggal Faktur :
2. Susun rapi teks-teks tersebut seperti pada gambar di atas.
3. Buatlah *field-field* untuk menampilkan data-data dari tabel *Fak_jual*, dengan mengisi *Expression* pada window *report expression* dengan teks-teks dibawah ini:
 - *Fak_Jual.Cus_Nama*
 - *Fak_Jual.Cus_Alamat*
 - *Fak_Jual.Fak_Kode*
 - *Fak_Jual.Fak_tgl*
4. Susun rapi *field-field* tersebut seperti pada gambar di atas.

Membuat Judul Data untuk Data Baris Faktur



Gambar 4.68 Judul untuk field-field BFak_jual (Child)



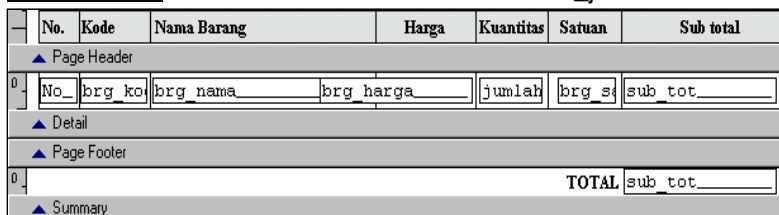
Cara untuk membuat tampilan judul untuk data baris faktur seperti di atas adalah sebagai berikut:

1. Buatlah teks-teks dibawah ini dengan menggunakan *Font=Times New Roman, Font style=Bold dan Size=9.*
 - No.
 - Kode
 - Nama Barang
 - Harga
 - Kuantitas
 - Satuan
 - Sub total
2. Susun rapi teks-teks tersebut seperti pada gambar di atas.
3. Langkah selanjutnya adalah, buatlah sebuah kotak baru seperti pada pembuatan *report* Daf_brg pada bahasan sebelumnya dengan posisi *Height=0,23 Width=6,13 Left=0,26 dan Bottom=160.*
4. Warnailah kotak tersebut dengan warna *gray.*
5. Klik menu utama *Format* dan sub menu *Send to back* untuk memastikan bahwa kotak tersebut berada dipaling bawah *report designer.*
6. Pisahkan jarak antar label dengan memberi satu garis vertikal pada setiap jarak seperti pada gambar di atas.

Menampilkan Data Baris Faktur pada Detail

Selanjutnya adalah menyusun *field-field* untuk menampilkan *data* bFak_jual sebagai *child* dari Fak_jual. Untuk menampilkan *data child* anda harus menempatkan *field-fieldnya* pada lokasi *Detail* dalam *report designer* Fak_jual tersebut. Lihat gambar di bawah.

Gambar 4.69 Susunan field-field dari tabel bFak_jual



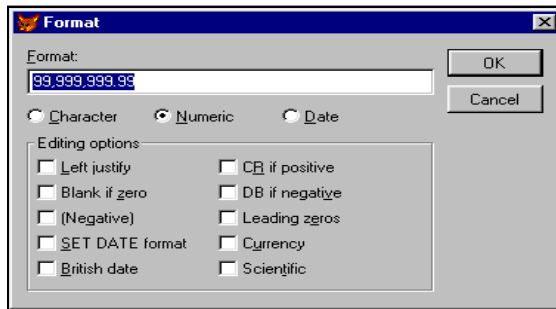


Untuk membuat tampilan pada lokasi *Detail* seperti pada gambar di atas adalah sebagai berikut:

1. Buatlah *field-field* untuk menampilkan data *bFak_jual* dengan mengisi *Expression* untuk *field-field* tersebut dengan uraian di bawah ini:
 - Untuk kolom No. *Expression* diisi dengan *bFak_jual.No* dengan *Font=Courier New, Fontstyle=Regular* dan *Size=10*.
 - Kolom Kode, *Expression* diisi dengan *bFak_jual.Brg_kode*.
 - Kolom Nama Barang, *Expression* diisi dengan *bFak_jual.Brg_Nama*
 - Kolom Harga, *Expression* diisi dengan *bFak_jual.Brg_harga*
 - Kolom Kuantitas *Expression* diisi dengan *bFak_jual.jumlah*
 - Kolom Satuan, *Expression* diisi dengan *bFak_jual.Brg_Sat*
 - Dan Kolom yang terakhir Sub total, *Expression* diisi dengan *bFak_jual.Sub_tot*
2. Berilah kotak untuk *field-field* di atas dengan membatasi jarak antar *field* dengan garis vertikal yang posisinya sama dengan kotak dan garis yang ada pada halaman page header.
3. Buatlah teks 'TOTAL :' pada lokasi *Summary*.
4. Pada lokasi *Summary* tempatkan satu buah *field*
5. Pada *window report expression*, isi *Expression* dengan *bFak_jual.Sub_tot*
6. Klik tombol yang berada pada sisi kanan baris format, maka akan muncul *window format* seperti pada gambar dibawah ini.

TIP
 Jika pada tampilan *report field* yang menampilkan *field* data *numeric*, bernilai bintang, seperti ini ***** maka *field* tersebut kurang lebar.

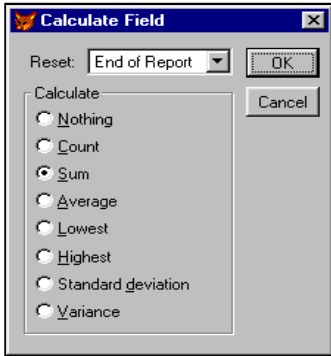
Gambar 4.70 kotak dialog Format



7. Isilah format dengan *Numeric* dan *text* format diisi dengan 99,999,999.99 maksud dari format tersebut adalah untuk membuat agar tampilan angka dari *field* tersebut menjadi terkelompok yaitu kelompok desimal, satuan, puluhan, ratusan dan seterusnya, klik tombol OK.
8. Selanjutnya agar *field* tersebut menampilkan jumlah total dari *field* *Sub_tot* anda harus mengklik tombol *Calculations* pada kotak dialog *Report Expression*, maka akan muncul kotak dialog baru dengan nama *Calculate Field*, pada kotak dialog tersebut pilihlah *End of report* di baris *Reset* dan *Sum* di kolom *Calculate*, seperti pada gambar dibawah ini.



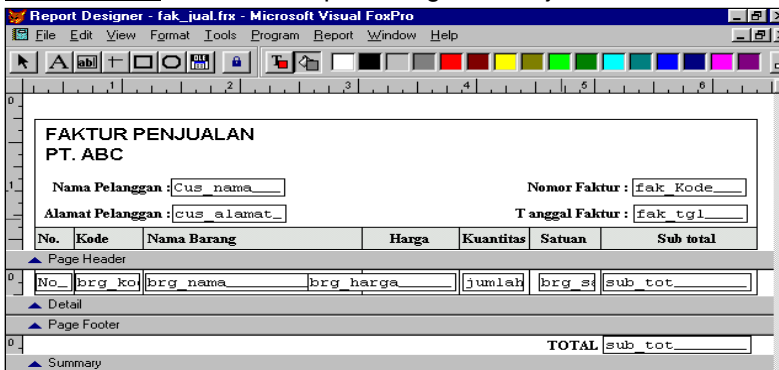
Gambar 4.71 Kotak dialog calculate field



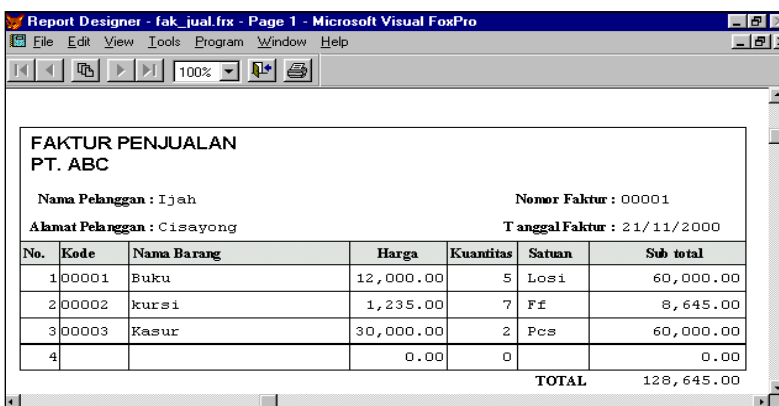
Untuk *Page setup* lakukanlah hal yang sama dengan langkah-langkah yang telah diuraikan pada pembuatan *report* sebelumnya, yaitu *report* *Daf_brg*. Setelah selesai rekamlah *report designer* *Fak_jual* tersebut.

Untuk melihat hasilnya jalankan *Form* *Form_brg* kemudian klik tombol *Cetak* maka hasilnya akan terlihat seperti pada gambar 4.73 dibawah.

Gambar 4.72 Hasil akhir report designer *Fak_jual*



Gambar 4.73 Preview report *Fak_jual*, faktur penjualan





4.2.3 Membuat Program Pembelian

Program pembelian adalah sarana untuk mencatat transaksi pembelian, untuk menyimpan dan mencari *data* transaksi pembelian sehingga anda sebagai programmer dituntut untuk membuat sebuah modul pembelian agar mudah digunakan dan sangat membantu user dalam mengerjakan pekerjaannya.

Membuat Form

Tahap awal dalam membuat program pembelian adalah membuat sebuah *Form* sebagai sarana untuk mengolah *data* transaksi pembelian, cara kerja *form* pembelian ini mirip dengan *form* penjualan, kalau dalam penjualan perusahaan tersebut mengeluarkan atau mengurangi barang dan kalau pembelian si perusahaan menambah stok barang. Karena cara kerja *form* untuk pembelian sama dengan *form* untuk transaksi penjualan maka untuk memudahkan dan mempercepat pekerjaan anda sebagai programmer maka anda tinggal mengcopy atau membuat duplikat *form* penjualan tersebut yang mempunyai nama *file* *Form_jual* yang nantinya anda cukup memodifikasi *form* tersebut menjadi *form* untuk transaksi pembelian baik struktur *datanya* atau pun perintah-perintahnya.

Tahap-tahap dalam membuat duplikat *form* dan memodifikasinya adalah sebagai berikut:

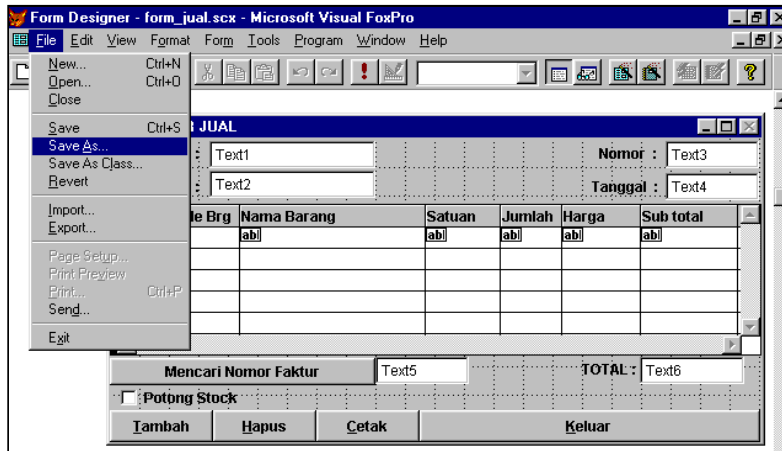
1. Membuka *Form* penjualan atau *Form_jual*

Dalam *Command window* anda buka *form* *Form_jual* dengan mengetik perintah *Modify Form C:\Npm\Form\Form_jual* atau dengan mengklik *pull-down* menu *File*, kemudian klik *Open* pada item menunya maka akan tampil *Open window* atau dapat pula dengan menekan tombol *Open* pada *toolbar standard* maka akan muncul *window Open* dan anda cari nama *folder* *Npm* pada direktori *C:* dan *File_jual* pada sub direktori *Form*.

2. Membuat duplikat *Form_jual*

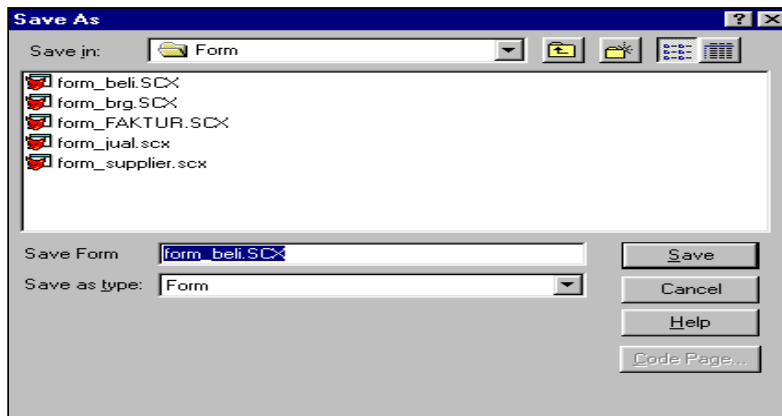
Setelah *Form_jual* ditampilkan maka anda tinggal mencopynya dengan mengklik *pull-down* menu *File* kemudian pilih dan klik item menu *Save as*, maka akan tampil *window Save as*, lihat gambar di bawah ini.

Gambar 4.74 Form designer Form_jual



PJ

Gambar 4.75 Save as Window



Pada direktori C:\Npm\Form ketiklah nama *file* untuk duplikat dari form_jual tersebut pada baris *Save Form*, anda ketik saja namanya dengan Form_beli setelah itu klik tombol *Save*, maka duplikat Form_jual dengan nama Form_beli telah terekam dan ditampilkan langsung pada *window Visual FoxPro*.

3. Memodifikasi tampilan Form_beli

Isi dari Form_beli ini sama dengan Form_jual untuk itu anda harus memodifikasinya mulai dari *caption form*, *caption object*, perintah dari tiap objek sampai struktur *datanya*. Dalam memodifikasi perintah-perintah yang digunakan dalam tiap objek tentu tidak seluruhnya dirombak tetapi hanya mengganti penulisan nama tabel dan nama *field* kalau memang berbeda, karena cara kerja untuk modul pembelian sama dengan cara kerja modul penjualan hanya nama *file* untuk *data* yang berbeda. Mulailah memodifikasi *Form* ini dari merubah tampilan *form*. Objek-objek yang dirubah *Captionnya* adalah sebagai berikut:



- *Form*, rubahlah *caption* dari *Form1* yaitu 'FAKTUR JUAL' menjadi 'MODUL PEMBELIAN' pada *properties Form1*
- *Label2*, *label2* ber*caption* 'Pembeli' anda rubah menjadi 'Penjual'
- *Header1* dari *Grid1.Column6*, yang tadinya 'Harga Jual' rubahlah menjadi 'Harga Beli'
- *Command5*, *caption* dari *Command5* adalah 'Mencari Nomor Faktur' rubah menjadi 'Mencari Nomor Bukti Pembelian'
- *Check1*, mulanya ber*caption* 'Potong Saldo Persediaan' rubahlah menjadi 'Tambah Saldo Persediaan'.

Rekamlah *Form* hasil memodifikasi tadi dengan mengklik menu *File, Save*, dan hasil dari modifikasi tersebut adalah seperti pada gambar di bawah ini.

Gambar 4.76 Form designer Form_beli



Membuat Struktur Data Pembelian

Sama seperti pada program penjualan transaksi pembelian juga menggunakan dua buah tabel yang saling berhubungan ada *Parent* dan ada *Child*, sebagai *file Parent* anda buat tabel dengan nama *file* Fak_beli dan untuk *file Child* anda buat tabel dengan nama *file* bFak_beli yang nama *field-fieldnya* telah disiapkan pada tabel di bawah ini.

Tabel 4.12 Daftar nama field untuk file Fak_Beli

Name	Type	Width	Decimal	Index
Fak_kode	Character	5		Ya
Fak_tgl	Date	8		
Sup_nama	Character	20		
Sup_alamat	Character	20		
Tot_harga	Numeric	12	2	
Mark_	Character	1		

Tabel 4.13 Daftar nama field untuk file bFak_beli

Name	Type	Width	Decimal	Index
Fak_kode	Character	5		Ya
No	Numerik	4		
Brg_kode	Character	5		Ya
Brg_nama	Character	20		
Brg_satuan	Character	4		
Brg_harga	Numeric	10	2	
Sub_tot	Numeric	10	2	



Buatlah tabel sesuai dengan daftar di atas dan rekamlah tabel-tabel tersebut ke direktori C:\Wpm\data.

Memasukan Table ke dalam Form

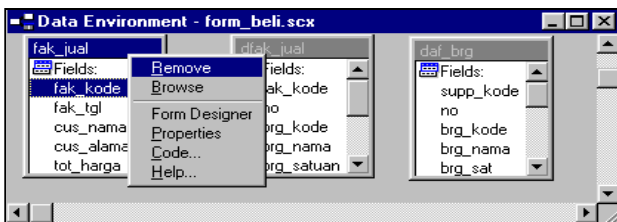
Buka *form* Form_beli yang telah anda rekam tadi dan untuk memanggil *form* tentunya anda telah mengetahuinya. Untuk memasukkan tabel yang anda perlukan pada *form* tersebut, yaitu dengan cara membuka *window data environment*, tabel-tabel yang diperlukan dalam *form* Form_beli yaitu ada tiga tabel:

1. Tabel **Daf_brg** sebagai daftar barang yang akan atau sudah dibeli,
2. Tabel **Fak_beli** yaitu tabel *parent* untuk menyimpan *data* nomor transaksi, tanggal dan penjual atau Supplier beserta alamatnya pada transaksi pembelian tersebut,
3. Tabel **bFak_beli** sebagai *file child* yaitu tabel untuk menyimpan *data* dari item-item yang dibeli, atau tabel untuk menyimpan *data* barang-barang yang dibeli.

Karena Form_beli adalah duplikat dari Form_jual yang di dalam *Data Environment*nya telah tersimpan tiga buah *data* yaitu Daf_brg, Fak_jual dan bFak_jual, sedangkan *data* Fak_jual dan *data* bFak_jual tidak akan diperlukan pada Form_beli ini maka anda tinggal meRemove kedua tabel tersebut. Langkah-langkah meremove *data* yang memiliki garis relasi adalah:

1. Anda harus menghapus garis relasinya terlebih dahulu yaitu dengan mengklik garis relasi tersebut kemudian tekan *Delete* pada *keyboard* anda.
2. Klik mouse kanan pada tabel yang akan anda hapus maka akan muncul menu *pulldown* dan anda klik *Remove*. Lihat gambar di bawah.

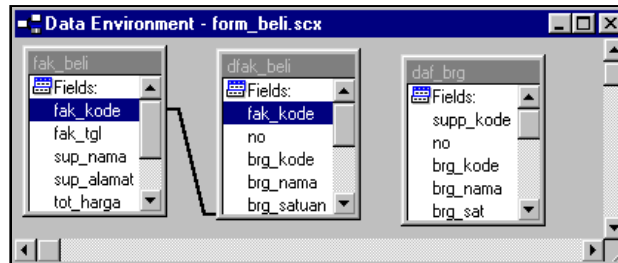
Gambar 4.77 Menghilangkan tabel pada data environment





Setelah anda menghapus kedua tabel tersebut selanjutnya anda masukan tabel Fak_beli dan tabel bFak_beli ke dalam *Data Environment* Form_beli, setelah anda masukan kedua tabel tersebut, sekarang anda tinggal menghubungkan tabel Fak_beli dengan tabel bFak_beli dengan cara mengklik *field index* yaitu *field Fak_kode* pada Fak_beli tahan dan geser kemudian masukan ke dalam tabel bFak_beli tepat di atas *field Fak_kode*. Kalau telah terhubung maka akan ada garis penghubung/garis relasi yang akan terlihat seperti gambar di bawah ini.

Gambar 4.78 Data environment Fom_beli



Mengisi Program pada setiap objek yang ada di Form Form_beli

Seperti yang telah diuraikan di atas bahwa perintah-perintah yang ada pada setiap objek pada Form_jual sepenuhnya akan dipakai juga pada Form_beli, berhubung Form_beli ini hasil duplikat dari Form_jual berarti anda cukup merubah nama tabel yang ada pada perintah *form* tersebut. Tetapi sebagai acuan untuk anda ada baiknya jika seluruh perintah dalam setiap objek pada form_beli ini dilampirkan tetapi tidak dengan penjelasannya karena penjelasannya telah diuraikan pada Form_jual. Berikut perintah-perintah yang terdapat dalam setiap objek pada Form_beli ini, penge-tikan seperti ini *Form1-Init* ini arinya Objek *Form* dan perintahnya di simpan pada *procedure Init*.

❑ *Form1-Init*

```
Sele fak_beli
go top
sele bFak_beli
set filter to fak_kode=fak_beli.fak_kode
Thisform.refresh
```

❑ *Form1-dHapus*

```
Y=0
DEFINE POPUP Tam FROM MROW(),MCOL();
SHORTCUT IN WINDOW Form1
DEFINE BAR 1 OF Tam PROMPT '<Tambah ?'
```

```

DEFINE BAR 2 OF Tam PROMPT '\<Hapus ?'
ON SELECTION POPUP Tam Y = BAR()
ACTIVATE POPUP Tam
IF Y > 0
    IF Y = 1
        sele bFak_beli
        calculate cnt(no) to a
        go bottom
        append blank
        Repl no with a+1,fak_kode ;
        with fak_beli.fak_kode
        set filt to fak_kode=fak_beli.fak_kode
        This.Refresh
    ELSE
        sele bFak_beli
        delete
        go top
        i=0

        scan
            i=i+1
            repl no with i
        endscan
    ENDIF
ENDIF
    
```



TIP
Anda dapat membuat menu <i>Pulldown</i> seperti ini pada sebuah tombol misal anda ingin menyatukan perintah tambah dan hapus untuk file Fak_beli pada satu tombol.

❑ *Text1*

Di *Text1* tidak ada perintah apa-apa hanya dalam *properties kategori data ControlSource* diisi dengan *fak_beli.Sup_nama*

❑ *Text2*

Di *Text2* juga sama *properties kategori data ControlSource* diisi dengan *fak_beli.Sup_alamat*

❑ *Text3*

Text3 dalam *properties kategori data ControlSource* diisi dengan *fak_beli.Fak_kode*

❑ *Text4*

Di *Text4* juga sama tidak ada perintah apa-apa hanya dalam *properties kategori data ControlSource* diisi dengan *fak_beli.Fak_tgl*

❑ *Text5-LostFocus*

Thisform.command5.setfocus

❑ *Text6*

Di *Text6* sama dengan *Text1* tidak ada perintah apa-apa hanya dalam *properties kategori data ControlSource* diisi dengan *fak_beli.Sub_tot*.



□ **Command1-Click (Tambah)**

```

*Mengisi File Parent
Sele Fak_beli
GO BOTT
xKode = val(Fak_beli.fak_kode)+1
if xKode = 1
    Thisform.text3.Value = '00000'
endi
appe blank
Repl Fak_beli.fak_Kode with;
Left((Thisform.text3.Value), ;
5-len(allt(str(Xkode))))+allt(str(xKode))
Repl fak_tgl with date()
xrec = recno()
go xrec

*Mengisi File Child
sele bFak_beli
go bottom
append blank
repl no with 1,fak_kode with Fak_beli.fak_kode

sele Fak_beli
go bott
thisform.refresh
Thisform.text1.setfocus

```

□ **Command2-Click**

```

Sele fak_beli
kode=allt(fak_beli.fak_kode)
IF messagebox('Anda yakin akan menghapus '"+kode+;
    "' ,36,'Hapus')=6
    sele bFak_beli &&Menghapus Child
    dele all for allt(fak_kode)=kode
    sele fak_beli &&Menghapus parent
    Delete
    skip-1
    if bof()
        go top
    endif
ENDIF
Thisform.refresh()

```

□ **Command3-Click**

```

Sele Fak_beli
rek=recno()
Sele bFak_beli
Set filter to fak_kode=Fak_beli.fak_Kode
Go top
_screen.show
report form c:\Npm\report\fak_beli preview noconsole

```



```
_screen.hide
acti wind form1
sele Fak_beli
go rek
```

❑ *Command4-Click*

```
Thisform.Release
```

❑ *Command5-Click (Mencari Nomor transaksi)*

```
public cari
cari=allt(Thisform.text5.value)
If Empty(cari)
    Return
Else
    Sele fak_beli
    rek=recno()
    Locate for allt(fak_kode)=cari
    If !found()
        go rek
    Endif
Endif
Thisform.refresh
rele cari
```

❑ *Grid1*

Properties List RecordSource diisi dengan *bFak_beli*

❑ *Grid1.Column1* sampai *Grid1.Column7*

Cara mengisi *control source* untuk *column* pada sebuah *grid* dapat dengan cara mengklik *grid* tersebut, kemudian pada *list object properties* pilihlah nama *columnnya*. Isilah *ControlSource* dari *Column-column* tersebut berdasarkan daftar di bawah ini.

1. *Column1*, *bFak_beli.No*
2. *Column2*, *bFak_beli.Brg_kode*
3. *Column3*, *bFak_beli.Brg_nama*
4. *Column4*, *bFak_beli.Brg_satuan*
5. *Column5*, *bFak_beli.Jumlah*
6. *Column6*, *bFak_beli.Brg_harga*
7. *Column7*, *bFak_beli.Sub_tot*

❑ *Grid1-RightClick*

```
Thisform.dHapus
```

❑ *(Grid1.Column1* sampai *Grid1.Column7).Text1-RightClick*

Kecuali *Column2*, mulai dari *Column1* sampai *Column7*, *object Text1* dan *procedure RightClick* diberi perintah seperti di bawah ini.

```
Thisform.dHapus
```



❑ *Grid1.Column2.Combo1-InteractiveChange*

Penulisan *Grid1.Column2.Combo1* adalah menyatakan *Object Combo1* yang terdapat dalam *Column2* pada *Grid1*, jadi yang dimaksudnya adalah *Combo1*. Perintahnya adalah sebagai berikut:

```
This.value=allt(daf_brg.brg_kode)
Sele bFak_beli
Repl brg_kode with daf_brg.brg_kode,;
   brg_nama with daf_brg.brg_nama,;
   brg_harga with daf_brg.brg_hbeli,;
   brg_satuan with daf_brg.brg_sat
Thisform.refresh
```

❑ *Grid1.Column5.Text1-Valid*

```
Sele bFak_beli
a=brg_harga*This.value
repl sub_tot with a
rek=recno()
go top
calc sum(sub_tot) to b
thisform.text6.value=b
go rek
thisform.refresh
```

❑ *Check1-Click*

```
Sele Fak_beli
rek=recno()
IF This.Value=1
   sele fak_beli
   repl Mark_ with '1'
   Sele bFak_beli
   set filt to allt(Fak_kode)=allt(Fak_beli.Fak_kode)
   scan
     if seek(Brg_kode,'Daf_brg','Brg_kode')
       sele Daf_brg
       repl Brg_Stock with;
       Brg_Stock+(bFak_beli.Jumlah);
       Brg_tambah with;
       Brg_tambah+(bFak_beli.jumlah);
       Brg_Hbeli with bFak_beli.Brg_Harga
     endif
   endscan
Else
   sele fak_beli
   repl Mark_ with '0'
   Sele bFak_beli
   set filt to allt(Fak_kode)=allt(Fak_beli.Fak_kode)
   scan
     if seek(Brg_kode,'Daf_brg','Brg_kode')
       sele Daf_brg
```



```

repl Brg_Stock with;
Brg_stock-(bFak_beli.Jumlah);
Brg_tambah with;
Brg_tambah-(bFak_beli.jumlah);
Brg_Hbeli with bFak_beli.Brg_Harga
endif
endscan
Endif
Sele Fak_beli
Go rek
Thisform.Refresh
    
```

❑ Check1-Refresh

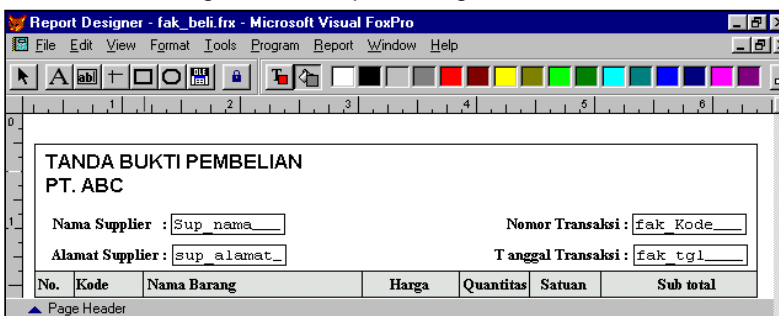
```

Sele Fak_beli
If mark_='1'
This.value=1
Else
This.value=0
Endif
    
```

Membuat Report Transaksi Pembelian

Untuk *report* pembelian sama dengan *report* penjualan hanya saja nama tabel yang digunakan bukan *Fak_jual* dan *bFak_jual* tetapi tabel *Fak_beli* dan *bFak_beli*. Jadi anda buat *report* seperti pembuatan *report* pada *report* penjualan, atau anda buat duplikat dari *report* penjualan, yaitu dengan membuka terlebih dahulu *report designer* *Fak_jual* kemudian klik menu utama *File* dan pada sub menunya klik *Save as*, seperti yang telah dilakukan pada pembuatan duplikat *form* *Form_beli*, dan untuk nama *filenya* anda ketik saja *Fak_beli*, sehingga anda cukup memodifikasi *report* *Fak_beli* tersebut dengan cara sebagai berikut.

Gambar 4.79 Page header report designer Fak_beli



Pada *page header* Label yang harus anda rubah adalah judul *report* menjadi 'TANDA BUKTI PEMBELIAN PT. ABC' cara merubah teks dalam sebuah label adalah dengan cara:

1. Klik *control label* pada *report controls toolbar*,
2. Tempatkan dan klik *mouse pointer* anda di atas label yang akan diubah, maka label tersebut telah siap untuk diedit.

PJ Rubahlah *label-label* pada *Report Fak_Jual* berdasarkan uraian di bawah ini:

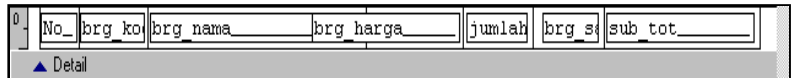
1. Ubahlah 'Nama Customer' menjadi 'Nama Supplier',
2. 'Alamat Customer' menjadi 'Alamat Supplier',
3. 'Nomor Faktur' menjadi 'Nomor Transaksi'
4. 'Tanggal Faktur' menjadi 'Tanggal Transaksi'.

Rubahlah *field-field* yang terdapat pada *Page header*, untuk merubah atau memodifikasi *field* pada *report* adalah dengan menampilkan *properties* atau *report expression window field* tersebut terlebih dahulu dengan cara sebagai berikut:

1. Klik dua kali *field* yang akan dirubah atau mengklik mouse kanan pada *field* tersebut maka akan muncul menu *pull-down*
2. Pada menu tersebut anda pilih sub menu *properties* maka akan muncul tampilan *window Report expression*.
3. Rubahlah nama *fieldnya* pada baris *Expression*,
4. *Field-field* yang harus anda rubah adalah:
 - Fak_jual.Cus_nama menjadi Fak_beli.Sup_nama
 - Fak_jual.Cus_alamat menjadi Sup_alamat,
 - Fak_jual.Fak_kode menjadi Fak_beli.Fak_kode
 - Fak_jual.Fak_tgl menjadi Fak_beli.Fak_tgl.

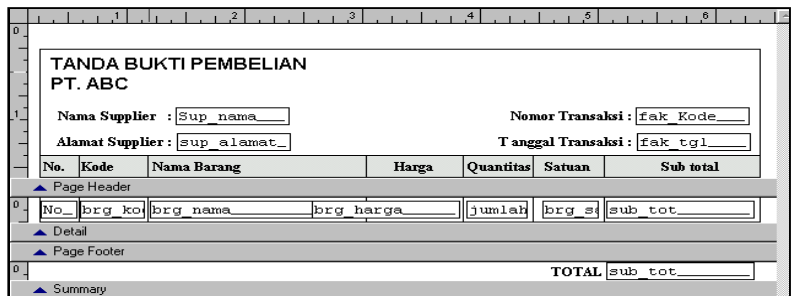
Selanjutnya adalah memodifikasi *field-field* yang ada pada kolom *Detail*, *field-field* yang ada pada *Detail* tidak perlu dirubah nama *fieldnya* tetapi cukup merubah nama tabelnya yang tadinya nama tabelnya *Fak_jual* anda rubah menjadi *Fak_beli* karena nama-nama *field* yang ada pada *Fak_jual* sama dengan nama-nama *field* yang ada pada *Fak_beli*. Contoh, jika *field* tersebut mempunyai *expression* *Fak_jual.No* anda cukup mengganti *Fak_jual* dengan *Fak_beli* menjadi *Fak_beli.No*.

Gambar 4.80 Kolom detail report designer Fak_beli



Dan ubah pula *field* *Sub_tot* yang berada pada kolom *Summary* yang tadinya *Fak_jual.subtot* menjadi *Fak_beli.subtot*.

Gambar 4.81 Hasil akhir report designer Fak_beli



Kalau anda mempunyai inisiatif atau ide untuk membuat *report designer* pembelian silahkan anda tuangkan ide anda sehingga mendapatkan hasil yang maksimal.



Dan hasil dari keseluruhan modul pembelian adalah seperti pada gambar di bawah ini.

Gambar 4.82 Form_beli

Gambar 4.83 Report Fak_beli

TANDA BUKTI PEMBELIAN						
PT. ABC						
Nama Supplier : Multi			Nomor Transaksi : 00002			
Alamat Supplier : Bandung			Tanggal Transaksi : 29/11/2000			
No.	Kode	Nama Barang	Harga	Quantitas	Satuan	Sub total
1	00002	kursi	4,500.00	40	Ff	180,000.00
2	00003	Kasur	25,000.00	10	Pcs	250,000.00
TOTAL						430,000.00

4.2.4 Membuat Form Menu

Untuk menampung semua modul yang telah anda buat di atas anda harus membuat satu *form* sebagai *form* menu utama yang mana *form* ini berfungsi untuk memanggil dan menutup seluruh *form* yang anda buat secara bergantian, cara membuat *form* menu tentunya agak sedikit berbeda dengan pembuatan *form-form* sebelumnya, di sini anda bebas berekspresi dan dituntut agar bisa memanipulasi tampilan sehingga tampilan untuk menu lain dari pada yang lain. Sebagai penuntun anda akan diberi contoh sederhana cara pembuatan *form* menu utama, yang langkah-langkahnya adalah sebagai berikut.

1. Sediakan atau buat *form* dengan ukuran Height=280 dan Width=607



2. Kemudian pada *properties* properti-properti yang digunakan adalah sebagai berikut:

Tabel 4.14 Daftar properties untuk Form Menu

Autocenter	.T. – True
BorderStyle	1 – FixedSingle
Caption	Perusahaan Dagang PT. ABC
Height	280
MaxButton	.F. – False
Name	MnUtama
ShowTip	.T. – True
ShowWindow	2 – As Top – Level Form
Width	607

3. Setelah *Form* selesai masukanlah 4 buah *CheckBox* dengan *properties* seperti pada tabel di bawah ini.

Tabel 4.15 Daftar properties untuk Check1

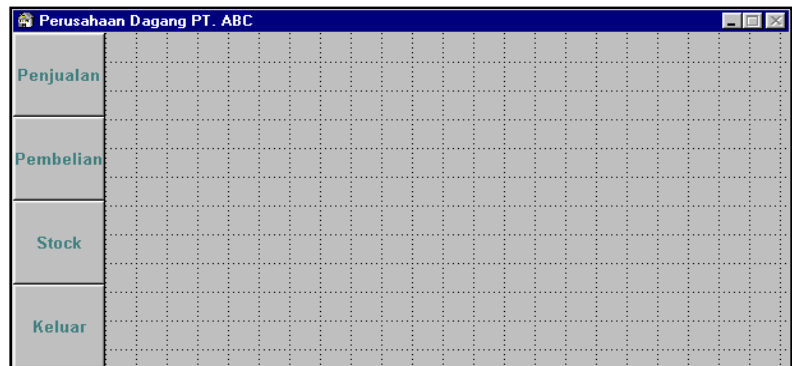
Caption	Penjualan
FontBold	.T. – True
FontSize	10
ForeColor	64,128,128 atau bebas
Height	70
Left	0
Name	Check1
Style	1 – Graphical
Top	0
Width	72

Untuk *Check2*, *Check3* dan *Check4* *properties* yang digunakan sama, yang berbeda hanya *Caption* dan *Top*,

- *Caption* untuk *Check2* adalah 'Pembelian' dan *Top*=70,
- *Check3* *Caption*='Stock', *Top*=140,
- dan untuk *Check4* *Caption*=Keluar dan *Top*=210.

Rekamlah *Form* tersebut pada direktori C:\Npm\Form dengan nama *Menu*.

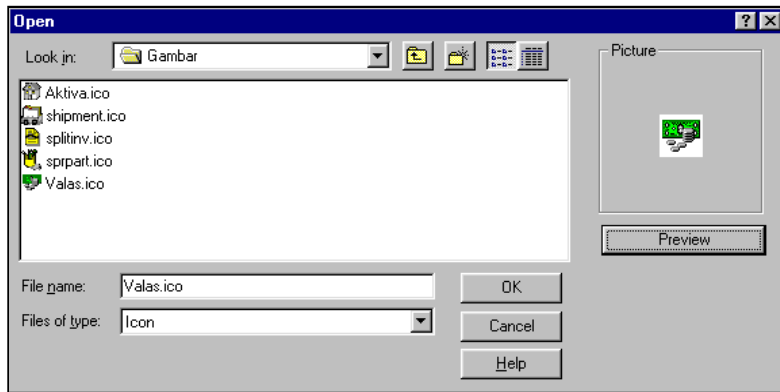
Gambar 4.84 Form designer Menu



4. Agar lebih menarik anda juga dapat menambahkan *Icon* pada setiap *CheckBox* atau tombol pada *form* menu tersebut, untuk mendapatkan sebuah *icon* anda dapat membuat atau meng-copy dari *file* lain dan tempatkan pada direktori C:\Npm\Gambar.
5. Cara memasukan *icon* ke dalam sebuah objek adalah sebagai berikut:
 - Klik objek yang akan diberi gambar
 - Pada *properties* pilih dan klik dua kali properti *Picture*
 - Maka akan muncul kotak dialog *Open*
 - Pilihlah folder C:\Npm\Gambar yang terdapat *icon-icon* yang telah anda buat tadi,
 - Pada baris *File of type* pilih *Icon*, setelah itu pilihlah *Icon* yang anda kehendaki. Lihat gambar di bawah ini.

PJ

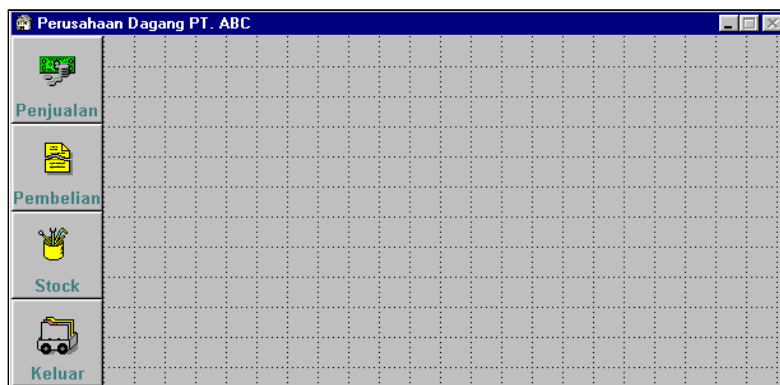
Gambar 4.85 Kotak dialog open untuk gambar



TIP
 Anda juga dapat memberi *background* pada *form* tersebut dengan warna atau juga *background* gambar cara memasukan gambar pada *form* sama dengan memasukan gambar pada *CheckBox*

Lakukan hal yang sama pada semua *CheckBox* yang ada pada *form* menu tersebut, Sehingga mendapatkan hasil seperti pada gambar di bawah ini.

Gambar 4.86 Form designer Menu



6. Pada *form* menu tersebut buatlah *Method*, dan *properties* baru dengan nama *Method* Keluar dan nama *properties* baru dengan nama Wontop yang fungsi dan kegunaannya akan di-



terangkan pada bahasan selanjutnya. Cara membuat *Method* baru telah diterangkan pada bahasan sebelumnya tetapi untuk mengingatkan anda kembali ada baiknya jika diterangkan lagi sedikit cara membuat *New Method* dan *New Properties*, sebagai berikut:

- Membuat New method Keluar
 - Pilihlah menu utama *Form*
 - Pada sub menu pilihlah *New Method*
 - Maka akan muncul kotak dialog *New Method* dan ketiklah nama *methodnya* yaitu Keluar
 - Klik tombol *Add*
 - Membuat New Properties Wontop
 - Pilihlah menu utama *Form* dan sub menu *New Property*
 - Pada kotak dialog *New Property* ketiklah nama propertinya yaitu Wontop,
 - Klik tombol *Add* atau langsung tekan *ENTER*.
7. Mengisi perintah-perintah pada *form* Menu (MnUtama) *Method-method* atau *procedure* yang digunakan untuk menyimpan perintah-perintah dalam *form* menu ini adalah sebagai berikut.

- MnUtama.Keluar

Perintah yang terdapat dalam Objek MnUtama (MnUtama adalah nama dari *Form* menu) dan *procedure* Keluar (*procedure* yang anda buat tadi) adalah sebagai berikut.

```
IF !EMPTY(Thisform.Wontop)
  RELE WINDOW (Thisform.Wontop)
  Empty(Thisform.Wontop)
ENDIF
Thisform.Refresh
```

Wontop adalah nama *property* baru yang anda buat, dan wontop ini juga merupakan sebuah program yang telah tersedia pada *Visual FoxPro* yang berguna untuk mengecek/mencatat *window* yang sedang aktif pada sebuah *window* yang levelnya lebih tinggi dari pada *window* yang akan dicek. Maka dari itu level *form* untuk *form* Menu ini harus paling tinggi dari *form-form* yang akan ditampilkan dalam *form* ini yaitu level 2 - *As top - Level Form*. Maksud dari perintah di atas adalah Jika pada Menu ini ada *Window* atau *form* yang ditampilkan atau aktif maka tutup *Window* yang sedang aktif tersebut dan di atas *Form* menu ini kosongkan atau tidak ada lagi *window* yang aktif.



- MnUtama.Load

Perintah yang tersimpan pada *procedure Load* adalah sebagai berikut.

```
Release Window Form1
```

Menyimpan perintah pada *procedure Load* akan dikerjakan pada saat *Form* tersebut mulai dijalankan atau *Runing form*, pada saat itu *form* ini memerintahkan untuk menutup atau mengclose *Form1* yang tentunya *form* selain *form* MnUtama ini.

- MnUtama.Init

Setelah mengerjakan *Load form*, selanjutnya *form* ini akan mengerjakan *Init*, perintah yang terdapat dalam *Form init* adalah.

```
Empty(thisform.Wontop)
```

Perintah di atas adalah untuk mengosongkan *window* MnUtama ini dari *window* lain

- MnUtama.Activate

Perintah yang terdapat dalam *procedure Activate* adalah perintah.

```
IF !Empty(Thisform.Wontop)
  Thisform.Keluar
ELSE
  Empty(Thisform.Wontop)
ENDIF
This.Refresh
```

Activate akan dikerjakan pada saat *form* ini aktif yaitu setelah mengerjakan perintah-perintah yang tersimpan dalam *procedure* selain *Activate*, perintah di atas memiliki arti sebagai berikut. Jika dalam *window* ini terdapat *window* lain maka *form* ini akan mengerjakan perintah yang tersimpan dalam *procedure* Keluar Jika tidak atau *Else* maka kosongkan *window* ini dari *window-window* lain.

- MnUtama.Unload

Perintah yang tersimpan dalam *unload* adalah

```
CLEAR EVENTS
SET STAT BAR OFF
```

Procedure Unload akan dikerjakan pada saat *Window* atau *form* ini ditutup atau *Closed*, perintah di atas memiliki arti bahwa pada saat *form* ini ditutup seluruh program aplikasi yang terdapat dalam aplikasi ini ditutup.

PJ

8. Mengisi perintah pada objek-objek yang terdapat dalam *Form MnUtama*. Objek-objek yang terdapat dalam *form* ini ada 4 yaitu *Check1*, *Check2*, *Check3* dan *Check4*. Dan *procedure* yang digunakan untuk semua objek ini adalah *Click*.

- *Check1.Click*

```

If this.value=1
    Do form C:\Npm\form\Form_jual
Endif
Thisform.Wontop=Wontop()
Thisform.Check2.Value=0
Thisform.Check3.Value=0
Thisform.Check4.Value=0

```

Seperti yang telah dibahas dalam bahasan sebelumnya bahwa *CheckBox* ini hanya memiliki dua nilai pilihan yaitu 1 dan 0, jika bernilai satu *Check* ini akan terbenam (*sunken*) dan jika nilainya nol *Check* ini akan kembali lagi kesemula yaitu terbit (*raised*). Nilai awal dari *Check* ini adalah nol. Untuk perintah di atas bermakna Jika *Check* ini bernilai 1 atau tenggelam pada saat diklik maka akan memanggil dan menjalankannya *Form_jual* yang terdapat dalam direktori *C:\Npm\form* setelah itu akan menutup *window* yang aktif sebelum *Form_jual* ini dijalankan. Dan untuk perintah *Thisform.Check2.Value=0* adalah memberi nilai nol untuk *CheckBox2* dan *Check-Check* lain selain *Check1*.

- *Check2.Click*

```

If this.value=1
    Do form C:\Npm\form\Form_beli
Endif
Thisform.Wontop=Wontop()

Thisform.Check1.Value=0
Thisform.Check3.Value=0
Thisform.Check4.Value=0

```

- *Check3.Click*

```

If this.value=1
    Do form C:\Npm\form\Form_brg
Endif
Thisform.Wontop=Wontop()

Thisform.Check1.Value=0
Thisform.Check2.Value=0
Thisform.Check4.Value=0

```

- *Check4.Click*

```

Thisform.release

```

Rekamlah *form designer* Menu ini, kemudian agar *form* yang dipanggil dalam menu ini posisinya pas atau sinkron dengan *form* menu maka *Left* dan *Top* pada *Form* yang akan dipanggil itu harus sesuai dengan *Form* menu. Pada *form* menu lebar dari tombol untuk memanggil *form* adalah 72 maka *Left* dari *form* yang akan dipanggil harus 72 dan *Topnya* 0, caranya adalah buka *form* *Form_jual* dan *Form_beli* kemudian pada *properties* isilah *Left=72* dan *Top=0*. Sehingga hasilnya tampak seperti gambar di bawah.



Gambar 4.87 Faktur jual yang dipanggil dalam Menu



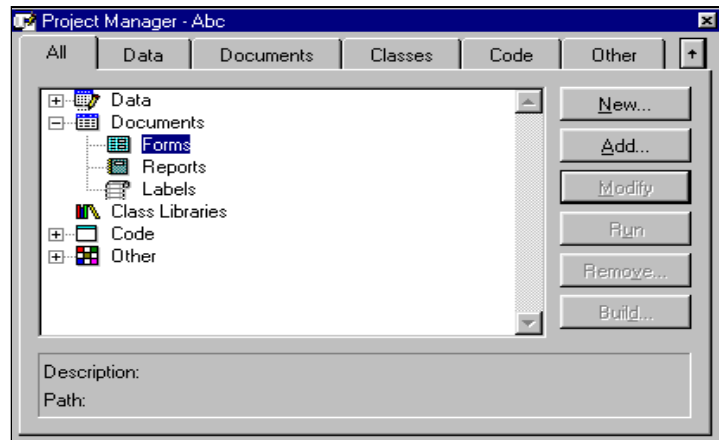
4.2.5 Membuat Project

Seluruh hasil pekerjaan yang anda buat harus dimasukan ke dalam *Project Manager* cara membuat *Project* ini adalah sebagai berikut.

1. Pertama tutup seluruh pekerjaan anda baik itu *Form designer* atau *report designer*.
2. Pilih menu utama *File* dan sub menu *New*, kemudian akan tampil kotak dialog *New*.
3. Pilihlah *Project* pada *File type*, dan klik tombol *New File*, maka akan muncul *window Create*,
4. Pada direktori *C:\Npm* ketiklah nama *filenya* sebagai contoh anda ketik *ABC* pada baris *Enter*.
5. Selanjutnya akan tampil *window Project Manager - Abc* yang masih kosong seperti pada gambar di bawah ini.

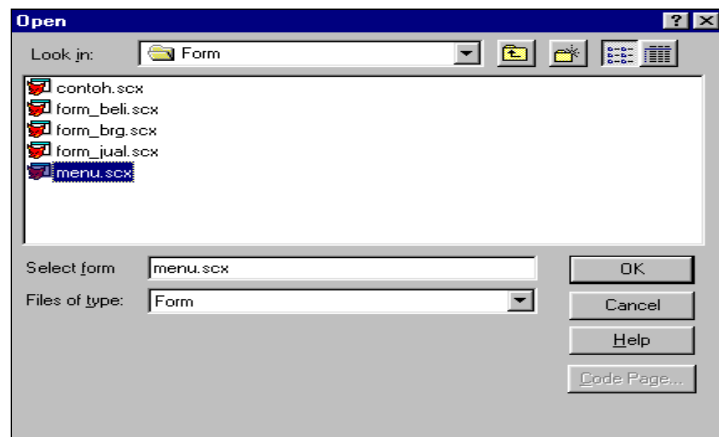
PJ

Gambar 4.88 Project manager Abc



6. Masukkan seluruh *Form* dan *report* ke dalam *project* ini,
 - Form* dimasukkan ke dalam *Documen Form* dengan cara:
 - Klik *Documen form*, lihat gambar di atas
 - Klik tombol *Add*, maka akan muncul *window Open*,

Gambar 4.89 Window open

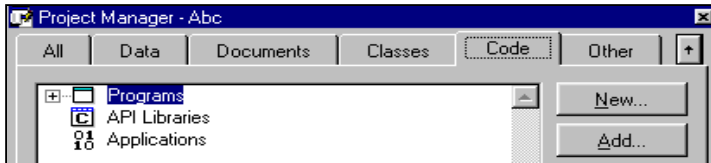


- Buka direktori C:\Npm\Form seperti pada gambar di atas anda masukan seluruh *Form* yang akan anda *Compile* satu per satu (Menu, Form_brg, Form_jual dan Form_beli).
- Report* dimasukkan ke *Documen Report* dengan cara:
 - Klik *Documen Report*, pada *Project manager window*
 - Klik tombol *Add*, maka akan muncul *window Open*,
 - Buka direktori C:\Npm\Report dan masukanlah seluruh *report* yang telah anda buat satu per satu.

7. Setelah semuanya dimasukan anda tinggal membuat sebuah program yang bertindak sebagai *Set main* caranya adalah sebagai berikut:

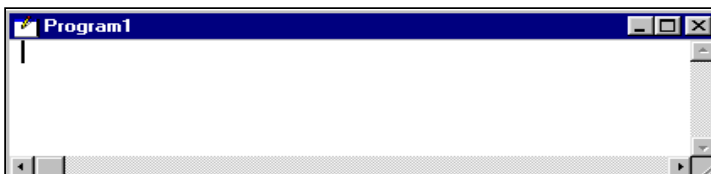
PJ

Gambar 4.90 Projec manager tab code



- Klik *Tab Code* pada *project manager* tersebut
- Kemudian klik baris *program* lihat gambar di atas
- Klik tombol *New* maka akan tampil *Window Program1* seperti gambar dibawah ini

Gambar 4.91 Window program1 yang masih kosong



- Isilah program tersebut dengan perintah di bawah ini.

```

SET DELETE ON
SET DATE TO DMY
SET SAFE OFF
SET TALK OFF
SET EXCLUSIVE OFF
SET EXACT ON
DO FORM Menu
_Screen.hide
READ EVENT
_Screen.Show
MODI WIND SCREEN
RELE ALL
CLOS ALL
CLEA ALL
    
```

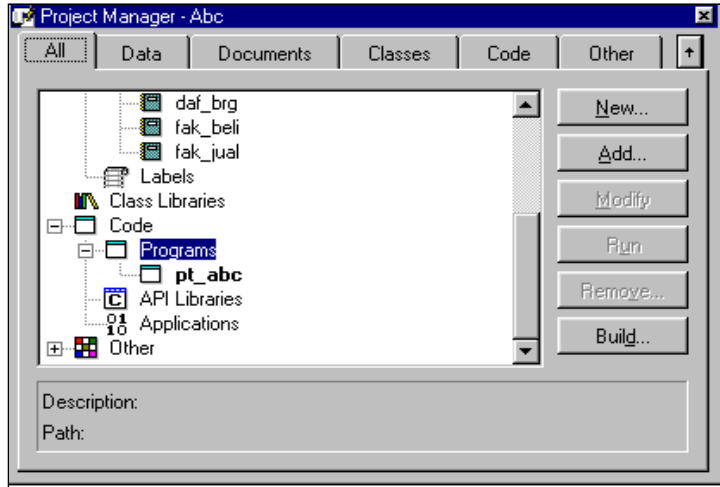
Program di atas adalah program untuk menyeting aplikasi anda dengan seting-seting yang umum di gunakan dalam membuat sebuah aplikasi, SET DELETED ON adalah suatu perintah agar semua *record* yang telah anda hapus tidak ditampilkan karena *record* yang anda hapus pada sebuah *table* sebenarnya tidak hilang tetapi hanya ditandai saja, anda dapat membedakan antara memasang seting-seting tersebut dengan tidak memasangnya. Kemudian perintah *DO FORM Menu* adalah perintah untuk memanggil atau menjalankan *Form Menu* sebagai Menu utama dari aplikasi ini. Rekamlah Program ini dengan nama *Pt_Abc.PRG* pada direktori *C:\Npm*.

PJ

8. Selanjutnya adalah menandai program yang menjadi *set main* untuk aplikasi ini, yang akan menjadi *set main* pada *project* ini adalah program Pt_Abc cara untuk menandai program ini menjadi *Set main* adalah dengan cara:
 - Klik kanan program Pt_Abc ini kemudian akan muncul menu *Pull-down*
 - Pilihlah sum menu *Set main* pada menu tersebut.
 - Setelah menjadi *Set main*, Pt_Abc ini akan tercetak tebal pada *project manager* ini. Lihat gambar.

TIP
Set main adalah program yang akan pertama kali dikerjakan pada saat aplikasi anda dijalankan dari *file* EXE.

Gambar 4.92 Project manager Abc

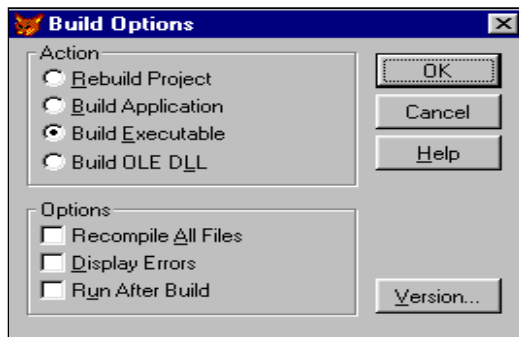


4.2.6 Membuat Executable Program

Untuk menjalankan aplikasi yang anda buat secara langsung pada komputer anda, maka anda harus membuat EXE dari program aplikasi tersebut. Cara membuat *file* EXE atau *Executable* adalah:

1. Klik tombol *Build* pada *Project Manager* yang telah anda buat di atas kemudian akan muncul kotak dialog *Build Option*.

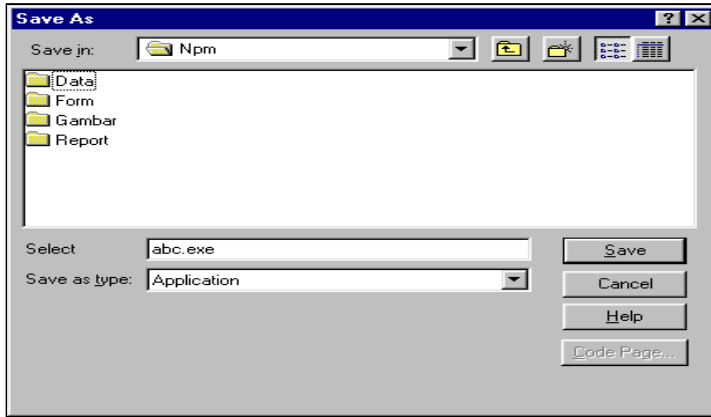
Gambar 20.93 Kotak dialog build option



2. Pada kolom *Action* pilihlah *Build Executable*, dan klik tombol OK, lihat gambar di atas
3. Maka akan muncul *Window Save as*, lihat gambar di bawah.

PJ

Gambar 4.94 Window save as



4. Ketiklah nama *file* EXE yang akan anda buat pada baris *select*, misal *abc.exe* kemudian klik tombol *Save*
5. Selanjutnya *Visual FoxPro* akan memproses pembuatan *file Executable* tersebut, setelah selesai coba anda jalankan EXE dari aplikasi yang telah anda buat tersebut.

Anda telah selesai membuat sebuah program aplikasi untuk sebuah perusahaan jual beli, dari sini diharapkan anda dapat mengembangkan ilmu pemrograman anda ke tingkat yang lebih tinggi atau tingkat mahir. Dan anda coba buat kembali sebuah program yang lebih kompleks permasalahannya misalnya program untuk rental mobil atau koperasi.

4.2.7 Daftar Perintah yang digunakan dalam Pembuatan Program di atas.

1. USE Command

Syntax:

```
USE [[DatabaseName.]Table | SQLViewName | ?]
    [IN nWorkArea | cTableAlias]
    [ONLINE]
    [ADMIN]
    [AGAIN]
    [NOREQUERY [nDataSessionNumber]]
    [NODATA]
    [INDEX IndexFileName | ?]
    [ORDER [nIndexNumber | IDXFileName]
    | [TAG] TagName [OF CDXFileName]
    [ASCENDING | DESCENDING]]]
    [ALIAS cTableAlias]
```

BS

[EXCLUSIVE]
 [SHARED]
 [NOUPDATE]

Perintah *USE* digunakan untuk membuka sebuah *file* data atau *Table* yang nantinya *table* yang anda buka tersebut dapat dimodifikasi, menambah *field*, mengisi data, atau hanya melihat isi dari *table* tersebut. Mungkin anda bertanya mengapa seluruh perintah untuk menginisialisasi *table* yang ada pada *form-form* di atas semuanya menggunakan *Select* bukan *Use* ?. jawabannya adalah karena di dalam *form* telah tersedia tempat untuk menyimpan nama-nama *table* yaitu *Data environment* sehingga anda tidak perlu lagi menyebutkan nama *table* yang akan digunakan. Tetapi jika anda tidak menyimpannya pada *data environment* maka anda harus menggunakan terlebih dahulu perintah *USE* ini sebelum menggunakan perintah *Select*.

Untuk memasukan *table* ke dalam *form* selain lewat *data environment* adalah dengan memasukannya ke dalam prosedur *form-Load* misalnya anda akan memasukan *table* *Daf_brg* *Fak_jual* dan *bFak_jual* caranya adalah dengan perintah sebagai berikut:

```
Use Daf_brg in 1 Alias Daf_brg
Use Fak_jual in 2 Alias Fak_jual
Use bFak_jual in 3 Alias bFak_jual
```

Pada perintah di atas angka 1,2 dan 3 adalah angka untuk membedakan antara *table* yang satu dengan *table* yang lainnya angka-angka tersebut jangan sampai dobel misalnya untuk *table* *Daf_brg* angkanya 1 dan untuk *Fak_jual* juga 1 dan juga anda bebas memberi nama *alias*, yang dipakai untuk menyatakan sebuah *table* adalah *alias* tersebut, misalnya untuk *table* *Daf_brg* anda memakai *alias* *Barang* (*Use Daf_brg in 1 Alias Barang*) yang dipakai untuk menunjuk nama *tablenya* adalah *Barang*, misal dalam sebuah perintah untuk menghapus:

```
Sele Barang
Delete all
```

Dan jangan lupa di dalam *procedure form-Unload* anda harus membuat perintah untuk menutup *table-table* tersebut, yang perintahnya adalah sebagai berikut:

```
Sele Daf_brg
Use
Sele Fak_jual
Use
Sele bFak_jual
Use
```

Itulah langkah-langkah untuk memasukan *table* ke dalam *form* selain dengan memasukannya kedalam *data environment*.

Contoh1: Untuk menambah/membuat *field* baru

```
USE Daf_brg Exclusive
Modify Structure
```

Contoh2: Untuk menampilkan data

```
USE Daf_brg
Browse
```

Contoh3: Untuk mengisi data

```
USE Daf_brg
Append
```

2. SELECT Command

Syntax: SELECT nWorkArea | cTableAlias

Keterangan:

Select adalah untuk menunjuk sebuah data atau *table* yang siap untuk di edit.

Contoh program, lihat pada setiap modul tepatnya pada tombol untuk menambah atau mengurangi data.

3. APPEND Command

Syntax:

```
APPEND [BLANK]
      [IN nWorkArea | cTableAlias]
      [NOMENU]
```

Keterangan:

Append Blank adalah menambahkan satu *record* kosong pada *table* yang aktif.

IN nWorkArea diisi dengan area kerja *table*, *table* dimana sebuah *record* baru ditambahkan. atau

IN cTableAlias diisi dengan nama *Alias* sebuah *table*, yang pada saat itu sebuah *record* ditambahkan pada *table* tersebut.

4. DELETE Command

Syntax:

```
DELETE
      [Scope] [FOR IExpression1] [WHILE IExpression2]
      [IN nWorkArea | cTableAlias]
      [NOOPTIMIZE]
```

Keterangan:

Scope, menandai *record-record* yang akan dihapus, klausa untuk *Scope* adalah : *ALL*, *NEXT nRecords*, *RECORD nRecordNumber*, dan *REST*.

FOR|Expression1 berisi kondisi untuk menandai *record* mana saja yang akan dihapus pada sebuah *table* dengan mengacu pada sebuah *field* atau lebih.

WHILE|Expression2 dapat diisi dengan *Expresi* kedua atau syarat kedua.

Contoh1: Menghapus seluruh *record* Brg_Hjual yang bernilai 20000

```
Use Daf_brg
Delete all for Brg_Hjual=20000
```

Contoh2: Menghapus seluruh *record* Brg_Hjual yang bernilai 20000 dengan syarat Brg_stock bernilai 0

```
Use Daf_brg
Delete all For Brg_Hjual=20000 While Brg_Stock=0
```

Record-record yang telah terhapus sebenarnya belum terhapus tetapi hanya ditandai saja, jadi kalau anda ingin membatalkan penghapusan itu, masih bisa yaitu dengan perintah *Recall*, dan sebaliknya untuk membuat bahwa si *record* yang dihapus tadi benar-benar dihapus maka gunakan perintah *Pack*.

5. FILTER() Command

Syntax:

```
SET FILTER TO [IExpression]
```

Filter adalah mengelompokan atau memfilter data berdasarkan isi dari *Expression* yang bersangkutan. Yang contohnya dapat anda lihat pada perintah yang terdapat dalam tombol cetak command3, *Form* persediaan. Kutipan dari perintah tersebut adalah sebagai berikut:

```
Sele bFak_jual
Set filter to fak_kode=Fak_jual.fak_Kode
```

6. IF...ENDIF Command

Syntax:

```
IF IExpression [THEN]
    Commands
[ELSE
    Commands]
ENDIF
```

Perintah IF...ENDIF adalah suatu perintah yang memiliki dua kemungkinan, kemungkinan pertama adalah benar (.T. True) dan kemungkinan lainnya adalah salah (.F. False). Contoh anda memiliki sebuah variable katakan variable A, yang mewakili sebuah kata yaitu 'PUTIH' permasalahannya adalah apakah huruf 'P' merupakan anggota dari variable A. cara menuangkan masalah tersebut ke dalam sebuah program adalah sebagai berikut.

```
A='PUTIH'
IF 'P' $ A
    Wait window 'Benar'
ELSE
    Wait window 'Salah'
ENDIF
```

7. SCAN...ENDSCAN Command

```
Syntax
SCAN [NOOPTIMIZE]
    [Scope] [FOR IExpression1] [WHILE IExpression2]
        [Commands]
    [LOOP]
    [EXIT]
ENDSCAN
```

SCAN...ENDSCAN adalah salah satu cara untuk membaca seluruh *record* dari atas ke bawah pada salah satu *table* secara otomatis dan mudah sehingga pada saat pembacaan *record* tersebut anda dapat menyimpan perintah-perintah lain di dalamnya yang bersangkutan dengan *table* tersebut atau *table* lain.

Contoh1: penggalan perintah yang terdapat dalam *object Check1 form_jual*, yang di dalamnya terdapat perintah *Seek*.

```
Select bFak_jual
Scan
    If seek(Brg_kode,'Daf_brg','Brg_kode')
        select Daf_brg
        Repl Brg_Hjual with bFak_jual.Brg_Harga
    Endif
Endscan
```

Contoh2:

```
Sele bFak_jual
i=0
Scan For Fak_kode='00001'
    i=i+1
    Repl no with i
Endscan
```

8. SEEK Command

Syntax

SEEK eExpression

```
[ORDER nIndexNumber | IDXIndexFileName
| [TAG] TagName [OF CDXFileName]
[ASCENDING | DESCENDING]]
[IN nWorkArea | cTableAlias]
```

Contoh1:

Sele bFak_jual

IF Seek(Brg_kode,'Daf_brg','Brg_kode')

Select Daf_brg

Repl Brg_Hjual with bFak_jual.Brg_Harga

Endif

Contoh2:

USE c:\vfp\samples\data\customer order company

set exac off

store 'Sa' to A

Seek A

IF Found()

 Display *field* company, contact

Endif

Contoh di atas diambil dari *table* yang terdapat dalam *Visual FoxPro*, contoh ini dapat anda catat pada sebuah program yang memiliki *extention* .PRG, misalnya program Cari.PRG dan coba anda jalankan pada *command window* dengan perintah DO CARI Maka hasil yang dapat anda lihat pada *window view* adalah seperti pada tampilan di bawah ini. Maksud dari contoh di atas adalah mencari *record* dari *field* Company yang dimulai dari suku kata 'Sa'

<u>Record#</u>	<u>Company</u>	<u>Contact</u>
70	Sante Gourmet	Jonas Bergulfsen

9. LOCATE Command

Syntax:

LOCATE FOR IExpression1

[Scope]

[WHILE]

[NOOPTIMIZE]

Perintah LOCATE FOR mempunyai kegunaan yang sama dengan perintah SEEK yaitu mencari dan menempatkan *pointer* pada *record* yang bernilai sama dengan nilai yang anda tentukan atau masukan.

```

Contoh1:
Sele Fak_jual
Locate for allt(fak_kode)='00001'
  If found()
    Wait window ' Ada'
  Else
    Wait window ' Tidak Ada'
  Endif

```

10. WONTOP() Function

Wontop() Function merupakan fungsi yang berguna untuk mengetahui nama *window* yang aktif pada saat ini, nama sebuah *window* dapat ditentukan pada saat anda membuat sebuah program, misalnya jika anda membuat sebuah *form* dan *form* tersebut diberi nama *Form1* maka *Wontop()* akan berisi *Form1*.

Syntax:
WONTOP([WindowName])

```

Contoh:
IF NOT EMPTY(WONTOP( ))
  WAIT WINDOW 'Window paling depan: ' + WONTOP( )
ELSE
  WAIT WINDOW 'Tidak ada Window yang terbuka'
ENDIF

```

11. LEFT() Function

Syntax
LEFT(cExpression, nExpression)

LEFT merupakan satu fungsi untuk mengambil beberapa *character* dari sebuah kalimat atau kata yang jumlah dari pengambilan kata tersebut ditentukan.

Contoh:

```
Left( 'Microsoft Vfp', 5)
```

Perintah tersebut memiliki arti, mengambil lima *character* dari kalimat 'Microsoft Vfp' mulai dari kiri ke kanan. Maka hasil dari perintah di atas adalah 'Micro'

12. SUBSTR() Function

Syntax
SUBSTR(cExpression, nStartPosition [,CharactersReturned])

SUBSTR() merupakan satu fungsi untuk mengambil beberapa *character* dari sebuah kalimat atau kata yang jumlah dari pengambilan kata dan mulai pengambilan kata tersebut ditentukan dengan nilai *numeric*.

Contoh1:

SUBSTR('Microsoft Vfp', 4,5)

Perintah tersebut memiliki arti, mengambil lima *character* dari kalimat 'Microsoft Vfp' dari kiri ke kanan dimulai dari *character* ke empat kalimat tersebut. Maka hasil dari perintah di atas adalah 'rosfo'

Contoh2:

SUBSTR('Microsoft Vfp', 6,4)

Hasil dari contoh2 adalah 'soft'

Contoh2:

SUBSTR('Microsoft Vfp', 6)

Hasil dari contoh3 adalah 'soft Vfp'

13. RIGHT() Function

Fungsi *RIGHT()* adalah untuk pengambilan *character* yang dimulai dari akhir kalimat. Dan jumlah *character* yang anda ambil ditentukan jumlahnya.

Syntax

RIGHT(cExpression, nCharacters)

Contoh1:

RIGHT('Fox Pro' , 3)

Hasil dari contoh1 adalah 'Pro'

14. LEN() Function

Syntax

LEN(cExpression)

LEN() function merupakan fungsi untuk mengetahui panjang sebuah string atau kalimat dapat juga untuk mengetahui panjang *field* pada sebuah *table*.

Contoh1:

LEN('Microsoft')

Hasil dari contoh1 adalah 9

Contoh2:

Use Daf_brg

LEN(Brg_nama)

Hasil dari contoh2 adalah 30.

15. RELATION() Function

Syntax

RELATION(nRelationNumber [, nWorkArea | cTableAlias])

Menghubungkan dua buah *table* selain dengan membuat garis relasi pada *Data Environment* ada juga cara lain yaitu dengan menggunakan *Relation() function*.

Contoh:

```
SELE Fak_Jual
SET RELATION TO Fak_kode INTO bFak_jual
```

Contoh di atas sama hasilnya dengan menghubungkan dua buah *table* dalam *Data Environment*, *Fak_jual* bertindak sebagai *Parent* dan *bFak_jual* sebagai *Child* dan *field* penghubung antara dua tabel tersebut adalah *Fak_kode*. Untuk menyimpan perintah tersebut dalam *form* adalah pada *procedure Form-Load*.

16. MESSAGE BOX () Function

Syntax

```
MESSAGEBOX(cPesan, [, nTipeKotak [, cJudul]])
```

cPesan diisi dengan pesan atau informasi yang akan disampaikan untuk user berhubungan dengan yang dilakukan oleh user.

nTipeKotak diisi dengan angka-angka yang telah disediakan oleh *Visual FoxPro* untuk tipe kotak dialog yang akan anda gunakan,

cJudul diisi dengan judul untuk judul *window message box* tersebut.

Contoh:

Contoh diambil dari perintah hapus dalam pemrograman *form form_brg* diatas:

```
Sele daf_brg
```

```
IF messagebox('Anda yakin akan menghapus ',36,'Hapus')=6
```

```
    sele daf_brg
```

```
    Delete
```

```
    skip
```

```
    if eof()
```

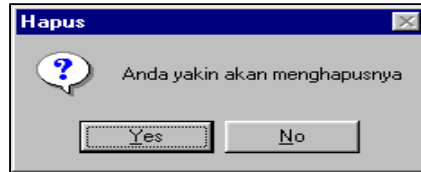
```
        go bott
```

```
    endif
```

```
ENDIF
```

```
Thisform.refresh()
```

Perintah yang dicetak tebal di atas itu maksudnya adalah memberikan tampilan atau tipe dari *messagebox*, sedangkan nilai 6 adalah untuk menunjuk tombol *Yes* yang akan menjadi *return* tombolnya Hasil dari perintah di atas yaitu seperti gambar di bawah ini:

Gambar 4.95 Message box

Jika anda mengklik tombol Yes (=tombol pertama) maka akan melanjutkan perintah yang ada di bawahnya yaitu perintah *Delete* sebaliknya jika anda memilih tombol No (=tombol kedua) maka anda melewati perintah yang ada di bawahnya atau mengerjakan perintah yang berada di lokasi *Else*. Berikut daftar dari nilai untuk mengisi nTypeKotak pada perintah *Message Box*.

Tabel 4.16 Daftar dari nilai untuk jumlah tombol

Nilai	Tombol Kotak Dialog
0	Tombol OK saja.
1	Tombol OK dan Cancel .
2	Tombol Abort, Retry, dan Ignore.
3	Tombol Yes, No, dan Cancel.
4	Tombol Yes dan No.
5	Tombol Retry dan Cancel.

Tabel 4.17 Daftar dari nilai untuk gambar icon

Nilai	Icon
16	Stop sign / tanda Stop.
32	Question mark / tanda tanya.
48	Exclamation point / tanda seru.
64	Information (I) icon / gambar (i).

Tabel 4.18 Daftar dari nilai untuk default tombol

Value	Tombol Default
0	Tombol pertama
256	Tombol kedua
512	Tombol ketiga

Tabel 4.19 Daftar dari nilai untuk return tombol

Nilai	Tombol
1	OK
2	Cancel
3	Abort
4	Retry
5	Ignore
6	Yes
7	No

Cara pengambilan nilai ini yaitu dengan cara menambahkan antara (Jumlah tombol + gambar *icon* + *default* tombol) contoh untuk nilai 36 yang dipakai pada perintah diatas yaitu (4+32+0) dan nilai returnnya 6.

D a f t a r P u s t a k a

Blake, 1985, Perception, Alfred A.Knopf, USA

Basset,2005,Internal Control on Accounting Information Systems,3rd,Manchester

Beard dan Wen,2007,Reducing the Threat levels for Accounting Information Systems: Challenges for management,Accountans,Auditor and Academicians, Harrison College of Business,USA

Bodnar,1995, Accounting Information Systems, 6Th, Prentice-Hall,USA

Boockholdt ,1999, Accounting Information Systems, 5Th, Mc Graw Hill, USA

Boisot ,1994, Information and Organization:The manager as antropologis, HarperCollins Publisher, Great Britain

Burch 1989, Information System:Theory and Practice, 3th, John Willey & Sons, New York

Chow,2005,Internal Control and Risk Management Guide Task Force,Hongkong ICPA, Hongkong

Date, 1990, An Introduction to Database Systems, 5Th, Addison-Wesley Publishing Company, USA

Gallier,1987, Information Analysis:Selected reading, Addison Wesley, Sydney

Gelinas and Sutton,2002,Accounting Information Systems,5th,South-Western Thomson Learning,USA

Hoffer,2005,Modern Database Manajemen,7th,Pearson Prentice Hall,USA

Kendal Kendal,1999,Systems Analysis and Design,4th,Prentice Hall,USA

Korth,1997, Database System Concepts, 2nd, Mc Graw-Hill Book Co, Singapore

Leon,1999,Enterprise resource Planning,Tata Mc Graw-Hill,India

Long ,1996, Computers, Prentice Hall International, Inc. USA

Maltin, 1994, Cognition, 3th, Harcourt Brace Publisher, USA

Marakas (2003),Decision Support Systems:in the 21st Century,Prentice Hall,USA

Nivra,2007,Risk Manajemen and Internal Control Systems,Amsterdam

Shelly ,2003,Systems Analysis and Design,5th,Thomson Course Technology,USA

Satzinger,2002,Systems Analysis and Design:in a Changing World,2nd, Thomson Course Technology,USA

Turban,1999,Information Technology for Management:Making Connection for strategic Advantage,2nd,John Wilwy&Sons.Newyork

Weber ,1999, Information Systems Control and Audit, Prentice Hall International Inc., USA

Whitten ,2004, System Analysis and Design Methods, 6th, Irwin/McGrowHill,USA

Wilkinson,2000, Accounting and Information Systems, 3th, John Wiley&Sons, USA