BAB II
LANDASAN TEORI

2.1 Rancang Bangun

Menurut Kamus Besar Bahasa Indonesia, Rancang berarti mengatur segala sesuatu sebelum bertindak, mengerjakan atau melakukan sesuatu untuk merencanakan. Sedangkan kata bangun berarti sesuatu yang didirikan (Departemen Pendidikan Nasional, 2002). Rancang bangun berarti merencanakan atau mendesain sesuatu yang akan dibuat (Departemen Pendidikan Nasional, 2002).

2.2 Sistem Penunjang Keputusan

Sistem penunjang keputusan sebagai suatu sistem interaktif berbasis komputer yang dapat membantu para pengambil keputusan dalam menggunakan data dan model untuk memecahkan persoalan yang bersifat tidak terstruktur. (Marimin, 2004)

Terdapat empat karakteristik utama dari sistem penunjang keputusan, dari definisi tersebut adalah sebagai berikut :

1. Sistem Penunjang Keputusan menggabungkan model dan data menjadi satu bagian.
2. Sistem Penunjang Keputusan dirancang untuk membantu para manajer (pengambil keputusan) dalamproses pengambilan keputusan dari masalah yang bersifat semi terstruktur.
3. Sistem Penunjang Keputusan cenderung dipandang sebagai pendukung penilian manajer tetapi tidakuntuk menggantikan posisi manajer.
4. Teknik Sistem penunjang keputusan dikembangkan untuk meningkatkan efektivitas dari pengambilkeputusan.

2.2.1 Proses Pengambilan Keputusan

Menurut Herbert A. Simon (Suprapto, 2005) tahap-tahap yang harus dilakukan dalam proses pengambilan keputusan sebagai berikut :

a. Tahap pemahaman (*inteligence Phase*)

   Tahap ini merupakan proses penelusuran dan pendeteksian dari lungkup problematika serta proses pengenalan masalah. Data masukan doiperoleh, diproses, dan diuji dalam rangka mengidentifikasikan masalah.

b. Tahap perancangan (*design phase*)

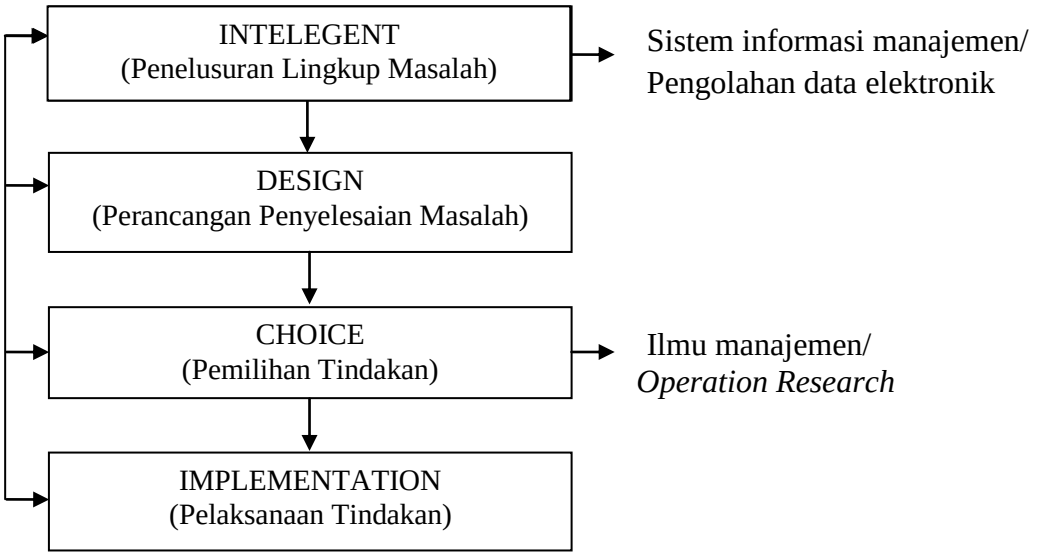
   Tahap ini merupakan proses pengembangan dan pencairan alternatif tindakan atau solusi yang dapat diambil. Hal tersebut merupakan representasi kejadian nyata yang disederhanakan, sehingga diperlukan proses validasi dan verifikasi untuk mengetahui model dalm meneliti masalah yang ada.

c. Tahap pemilihan (*Choice Phase*)

   Tahap ini dilakukan pemilihan terhadap diantara berbagai alternatif solusi yang dimunculkan pada tahap perencanaan agar ditentukan atau dengan memperhatikan kriteria-kriteria berdasarkan tujuan yang akan dicapai.

d. Tahap implementasi (*Implementation Phase*)

   Tahap ini dilakukan penerapan terhadap rancangan sistem yang telah dibuat pada tahap perancangan serta pelaksanaan alternatif tindakan yang telah dipilih padatahap pemilihan.



Gambar 2.1 Fase Proses Pengambilan Keputusan

2.2.2 Karakteristik sistem penunjang keputusan

(Turban,2005) mengumumkan karakteristik dan kapabilitas kunci dari sistem penunjang keputusan, sebagai berikut :

a. Dukungan untuk pengambilan keputusan, terutama pada situasi semi tersetruktur dan tak terstruktur.
b. Dukungan untuk semua level manajerial, dari eksekutif pucak sampai manager ini.
c. Dukungan untuk individu dan kelompok.
d. Dukungan untuk semua keputusan independen dan sekuential.
e. Dukungan disemua fase pengambilan keputusan
f. Dukungan dalam berbagai proses dan gaya pengambilan keputusan dapat menghadapi masalah-masalah baru dan pada saat yang sama dapat menanganinya dengan cara mengadaptasikan sistem terhadap kondisi-kondisi perubahan yang terjadi.
g. Pengguna menrasa seperti rumah. *User-friendly,* kapabilitas grafis yang kuat, dan sebuah bahasa yang interaktif dan alami.

h. Peningkatan terhadap keefektifan pengambilan keputusan(akurasi, *time lines*, kualitas) dari pada efesiensi (biaya).
i. Pengambilan keputusan mengontrol penuh semua langkah proses pengambilan keputusan.
j. Pengguana akhir dapat mengembangkan dan memodifikasi sistem sederhana.
k. Menggunakan model-model dalam penganalisisan situasi pengambilan keputusan.

2.2.3 Keuntungan Sistem Penunjang Keputusan

Keuntungan dari sistem penunjang keputusan sebagai berikut:

1. Sistem penunjang keputusan memperluas kemampuan pengambilan keputusan dalam memproses data atau informasi bagi pemakainya.
2. Sistem penunjang keputusan membantu pengambilan keputusan untuk memecahkan masalah terutsama sebagai masalah yang sangat kompleks dan tidak tersetruktur.
3. Sistem penunjang keputusan dapat menghasilkan solusi dengan lebih cepat serta hasilnya dapat diandalkan.
4. Sistem penunjang keputusan dapat menjadi stimulasi bagi pengambil keputusan dalam memahami persoalannya, karena mampu menyajikan berbagai alternatif pemecahan.
5. Mampu menyediakan bukti tambahan untuk memberikan pembenaran, sehingga dapat memperluas posisi pengambilan keputusan.

2.2.4 Komponen Sistem Penunjang Keputusan

1. Sub Sistem Pengelolaan Data (*Database)*

Sub sitem pengelolaan data *(database)* merupakan komponen sistem penunjang keputusan yang berguna sebagai penyedia data bagi sistem. Data tersebut disimpan dan diorganisasikan dalam sebuah basis data yang

diorganisasikan oleh suatu sistem yang disebut dengan sistem manajemen basis data (*database management system).*

2. Sub Sistem Pengelolaan Model (*Model Base).*

Keunikan dari sistem penujang keputusan adalah kemampuanya dalam mengintegrasikan data dengan model-model keputusan.Model adalah suatu tiruan dari alam nyata. Kendala yang sering dihadapi dalam merancang suatu model adalah bahwa model yang dirancang tidak mampu mencerminkan seluruh fariabel alam nyata, sehingga keputusan yang diambil tidak sesui dengan kebutuhan, oleh karena itu dalam menyimpan berbagai model yng harus di perhatikan adalah pada setiap model yang disimpan hendaknya ditambahkan rincian keterangan dan penjelasan yang komprehensif mengenai model yang dibuat.

3. Sub Sistem Pengelolaan Dialog (*User Interaktif.*

Keunikan lainya dari sistem penunjang keputusan adalah adanya fasilitas yang mampu mengintegrasikan sistem yang terpasang dengan pengguna secara interaktif yng dikenal dengan sub sistem dialog. Melalui sub sistem dialog sistem diimplementasikan sehingga pengguna dapat berkomunikasi dengan sistem yang dibuat.

2.2.5 Metode SAW (*Simple Additive Weighting*)

Metode SAW*(Simple Additive Weighting)* sering juga dikenal istilah metode penjumlahan terbobot. Konsep dasar metode SAW*(Simple Additive Weighting)* adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut. Metode SAW *(Simple Additive Weighting)*membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada.Metode ini merupakan metode yang paling dikenal dan paling banyak digunakan orang dalam menghadapi situasi MADM *(multiple attribute decision making).*Metode ini mengharuskan pembuat keputusan menentukan bobot bagi setiap atribut. Skor total untuk sebuah alternatif diperoleh dengan menjumlahkan seluruh hasil perkalian antara rating (yang dapat

dibandingkan lintas atribut) dan bobot tiap atribut. Rating tiap atribut haruslah bebas dimensi yang artinya telah melewati proses normalisasi sebelumnya (Kusumadewi, Sri *dkk* ,2006).

$$R_{ij} = \begin{cases} \dfrac{x_{ij}}{Max_i x_{ij}} & \text{jikaj adalah atribut keuntungan (}\textit{benefit}\text{)} \\ \dfrac{Min_i\ x_{ij}}{x_{ij}} & \text{jika j adalah atribut biaya (}\textit{cost}\text{)} \end{cases} \quad \text{.....(1)}$$

Keterangan:

1. R_{ij} = Nilai ranting kinerja ternormalisasi.
2. x_{ij} = Nilai atribut yang dimiliki dari setia kriteria.
3. $Max_i\ x_{ij}$ = Nilai terbesar dari setiap kriteria.
4. $Min_i\ x_{ij}$ = Nilai terkecil dari setiap atribut.

Dimanar*ij*adalahratingkinerjaternormalisasidarialternatifA_i padaatributC_j; i=1,2,...,mdanj=1,2,...,n.Nilaipreferensiuntuksetiapalternative(V_i)diberikansebagai:

$$V_i = \sum_{j=1}^{n} w_i r_{ij} \quad \text{.....(2)}$$

Keterangan:

1. V_i = Rangkinguntuksetiapalternatif.
2. W_j = Nilaibobot darisetiapkriteria.

3. r_{ij} = Nilairatingkinerjaternormalisasi.

Nilai$V_i$yanglebihbesarmengindikasikanbahwaalternatif$A_i$lebihterpil
ih.

Langkah Penyelesaian SAW *(Simple Additive Weighting)*:

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilankeputusan, yaitu C_i.
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan kriteria(C_i), kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi R.
4. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vektor bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (A_i)sebagai solusi.

2.3 Guru Berprestasi

Guru Berprestasi adalah guru yang memiliki kinerja melampaui standard yang ditetapkan oleh satuan pendidikan, mencangkup empat kompetensi yaitu pedagodik, kepribadian, social dan professional, menghasilkan karya inovatif yang diakui baik pada tingkat daerah, nasional maupun internasional, dan secara langsung membimbing peserta didik hingga mencapai prestasi dibidang intrakulikuler dan atau ekstrakulikuler.

Menurut Undang-undang 14 tahun 2005 tentang guru dan dosen pasal 36 ayat 1 yang mengamantkan bahwa guru yang berprestasi, berdedikasi luar biasa dan atau bertugas didaerah khusus berhak memperoleh penghargaan.

2.4 Teori Pengembangan Sistem

Teori-teori pengembangan sistem yang digunakan dalam penelitian ini adalah :

2.4.1 Metode Pengembangan Perangkat Lunak

Pengembangan sistem (*system development*) dapat berarti menyusun suatu sistem yang baru untuk menggantikan yang lama secara keseluruhan atau memperbaiki sistem yang telah ada (Jogiyanto,2005). Adapun beberapa hal yang menyebabkan perlunya pengembangan sistem adalah adanya permasalahan-permasalahan yang timbul pada sistem yang lama, masalah yang timbul dapat berupa ketidakberesan sistem dan pertumbuhan organisaso, selain itu penyebab perlunya pengembangan sistem adalah untuk meraih kesempatan-kesempatan serta adanya intruksi-intruksi.

Beberapa prinsip dalam proses pengembangan sistem (Jogiyanto,2005) adalah sebagai berikut:

a. sistem yang dikembangkan adalah untuk manajemen.
b. Sistem yang dikembangkan adalah investasi modal yang besar.
c. Sistem yang dikembangkan memerlukan orang terdidik.
d. Tahapan kerja dan tugas-tugas yang harus dilakukan dalam proses pengembangan sistem.
e. Proses pengembangan sistem tidak harus urut.
f. Jangna takut membatalkan proyek.
g. Dokumentasi harus ada untuk pemedoman dalam pengembangan sistem.

Selain itu, dalam melakukan pengembangan sistem (Andri Kristanto,2003) terdapat beberapa fase yang digambarkan padaa siklus hidup pengembangan sistem, desain sistem, implementasi sistem, dan pemeliharaan sistem, serta dilengkapi dengan fase fase evaluasi yang dilakukan untuk memastikan bahwa pengembangan sistem sesuai dengan rencana yang telah ditetapkan baik dari segi waktu, biaya, maupun teknis.

Untuk melakukan pengembangan sistem dibutuhkan suatu metodologi-metodologi pengembangan sistem adalah metode-metode, prosedur-prosedur, dan aturan-aturan yang akan digunakan untuk mengembangkan suatu sistem informasi. Adapun beberapa metodologi pengembangan sistem tergantung pada orientasinya, mulai dari berorientasi keluaran, proses, dan objek. Adapun dalam pengembangan sistem ini penulis menggunakan metodologi berorientasi proses yang diperkenalkan sekitar tahun 1970 yakni metodologi struktur perancangan dan desain, metodologi ini telah dilengkapi dengan alat-alat dan tekhnik-tekhnik yang dibutuhkan untuk pengmbangan sistem khususnya pemrograman terstruktur atau modular. Berikut adalah alat dan teknik perancangan sistem yang digunakan untuk membantu merancang dan menganalisa sebuah sistem.

2.3.2 DFD (*Data Flow Diagram*)

DFD (*Data Flow Diagram*)awalnya dikembangkan oleh Chris Gane dan Trish Sarson pada tahun 1979 yang termasuk dalam *Structurred Systems Analysis and Design Methodology* (SSADM) yang ditulis oleh Chris Gane dan Trish Sarson. System yang dikembangkan ini berbasis pada dekomposisi fungsional dari sebuah system.

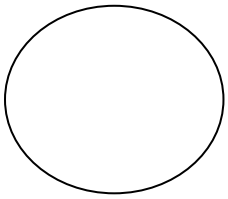
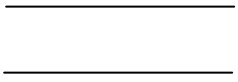
Edward Yourdon dan Tom DeMarco memperkenalkan metode yang lain pada tahun 1980-an dimana mengubah persegi dengan sudut lengkung (pada DFD Chris Gane dan Trish Sarson) dengan lingkaran untuk menotasikan. DFD Edward Yourdon dan Tom DeMarco popular digunakan sebagai model analisis system perangkat lunak untuk system perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur. *Data Flow Diagram* (DFD) atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasikan grafik yang menggambarkan aliran informasi dan ransformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*).

DFD dapat digunakan untuk mempresentasikan sebuah system atau perangkat lunak pada beberapa level abstraksi.DFD daapat dibagi menjadi

beberapa level lebih detail untuk mempresentasikan aliran informasi atau fungsi yang lebih detail. DFD menyediakan mekanisme untuk pemodelan fungsional ataupun pemodelan aliran informasi. Oleh karena itu itu, DFD sering digunakan untuk memodelkan fungsi-fungsi perangkat lunak yang akan diimplementasikan menggunakan pemrograman terstruktur karena pemrograman terstruktur membagi-bagi bagiannya dengan fungsi-fungsi dan prosedur-prosedur (Rosa A. S., 2011).

Simbol yang digunakan dalam DFD adalah seperti pada tabel 2.1 berikut

Tabel 2.1 Simbol *Data Flow Diagram.*

Simbol	**Keterangan**
External Entity 	Sumber atau tujuan dari aliran data dari atau ke sistem.
Arus Data *(data flow)*	Menggambarkan aliran data.
Proses (process) 	Proses atau fungsi yang mentransformasikan data masukan menjadi data keluaran.
Simpan Data *(Data Store)* 	Komponen yang berfungsi untuk menyimpan data atau *file*.

Berikut ini adalah tahapan-tahapan perancangan dengan menggunakan DFD :

1. Membuat DFD Level 0 atau sering disebut juga *Context Diagram*

   DFD Level 0 menggambarkan system yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun system lain.

DFD Level 0 digunakan untuk menggambarkan interaksi antara system yang akan dikembangkan dengan entitas luar.

2. Membuat DFD Level 1

   DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam system yang akan dikembangkan. DFD Level 1 merupakan hasil *breakdown* DFD Level 0 yang sebelumnya sudah dibuat.

3. Membuat DFD Level 2

   Modul-modul pada DFD Level 1 dapat di *breakdown* menjadi DFD Level 2. Modul mana saja yang harus di *breakdown* lebih detail tergantung pada tingakat kedetailan modul tersebut.

4. Membuat DFD Level 3 dan Seterusnya

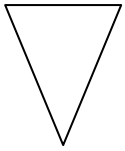
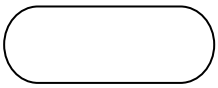
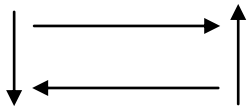
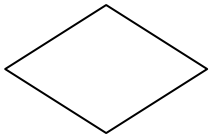
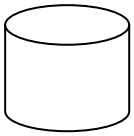
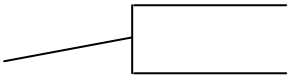
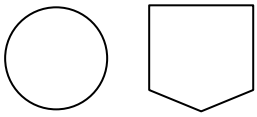
   DFD Level 3, 4, 5 dan seterusunya merupakan *breakdown* dari modul pada DFD Level diatasnya.

2.3.3 Bagan Alir Dokumen(*document flowchart*)

Bagan alir dokumen (*document flowchart*) merupakan bagan alir yang menunjukan arus data dari laporan dan formulir termasuk tembusan-tembusannya (Jogiyanto,2005). Bagan alir dokumen ini menggunakan simbol-simbol seperti yang terlihat pada tabel 2.2.

Tabel 2.2 Simbol Bagan Alir Dokumen

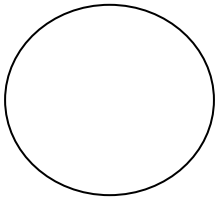
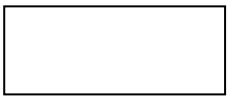
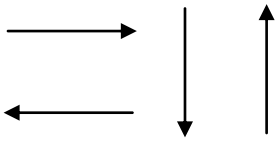
Simbol	Keterangan
Dokumen 	Menunjukan dokumen yang digunakan untuk *input* dan *output* baik secara manual maupun komputerisasi
Proses Manual 	Menunjukan pekerjaan yang dilakukan secara manual

Proses Komputerisasi 	Menunjukan proses dari operasi program computer
Simpan 	Menunjukan arsip
Terminator 	Digunakan untuk memberikan awal dan akhir suatu proses
Garis Alir 	Digunakan untuk menunjukan arus dari proses
Decision 	Digunakan untuk suatu penyeleksian kondisi didalam program
Keyboard 	Menunjukan *input* yanng menggunakan *keyboard*
Hard Disk 	Media penyimpanan, menggunakan perangkat *hard disk*
Keterangan 	Digunakan untuk memberikan keterangan yang lainnya.
Penghubung 	Simbol yang digunakan untuk menunjukan sambungan dari bagian alir yang terputus dihalaman yang sama maupun dihalaman yang lain.

2.3.4 Diagram Konteks *(context diagram)*

Diagram konteks *(context diagram)* adalah suatu alat yang digunakan untuk menggambarkan alir data dan interaksi sistem secara umum. Ada tiga simbol yang digunakan pada konteks diagram untuk menggambarkan alir data, yaitu:

Tabel 2.3 Simbol Diagram Kontes (*context diagram*)

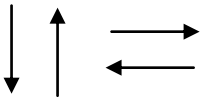
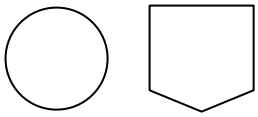
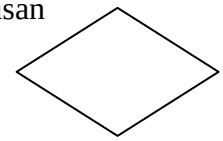
Simbol	Keterangan
Proses 	Orang atau unit yang mempergunakan atau melakukan transformasi data. Komponen fisik tidak didefinisikan
Kesatuan Luar (*enternal entity*) 	Entitas eksternal dapat berupa orang atau unit terkait yang berinteraksi dengan sistem tetapi diluar sistem
Arus Data (*data flow*) 	Arus data dengan arah khusus dari sumber ke tujuan

2.3.5 Bagan Alir Program (*program flowchart*)

Bagan alir program (*program flowchart*) adalah saat yang penting bagi pengguna atau *user* untuk dapat memahami logika program secara terperinci. Bagan alir ini dibuat dari derifikasi bagan alir sistem. Bagan alir program (*program flowchart*) merupakan bagan yang menjelaskan secara

rinci langkah-langkah dari proses program (Jogiyanto,2005). Bagan alir logika program digunakan untuk menggambarkan tiap-tiap langkah didalam program komputer secara logika. Untuk simbol-simbol program *flowchart*hampir sama dengan bagan alir dokumen, hanya terdapat perbedaan pada simbol proses, penyimpanan data *input/output* data. Adapun simbol-simbol program-program *flowchart* sebagai berikut.

Tabel 2.4 Simbol bagan alir program

Simbol	**Keterangan**
Input/Output 	Simbol *input/output* digunakan untuk mewakili data *input/output*
Proses 	Simbol proses (*processing symbol*) atau simbol pengolah yang digunakan suatu poses
Garis Alir 	Simbol garis alir (*flow lines symbol*), digunakan untuk menunjukan arus dari proses
Penghubung 	Simbol penghubung (*connctor symbol),* digunakan untuk menunjukan sambungan dari bagan alir yang terputus dihalaman yang sama / dihalaman yang lain.
Keputusan 	Simbol keputusan (*decison symbol),* digunakan untuk suatu penyelesaian kondisi didalam program.
Proses terdefinisi 	Simbol proses terdefinisi digunakan untuk menunjukan suatu operasi yang rinciannya ditunjukan ketempat lain.
Terminal 	Simbol digunakan untuk menunjukan awal dan akhir dar suatu program

2.3.6 Kamus Data (*Data Dictionary*)

Kamus Data (*data dictionary*) dipergunakan untuk memperjelas aliran data yang digambarkan pada DFD. Kamus data adalah kumpulan daftar ellemen data yang mengalir pada system perrangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum / memiliki standard cara penulisan (Rosa A. S., 2011).

Berikut contoh tampilan format kamus data pada Gambar 2.2.

Nama Database :

Nama Tabel :

Primary Key :

Foreign Key :

Nama Field	Type	Size	Kondisi	Keterangan

Keterangan kondisi berisi (contoh: NULL/NOT NULL)

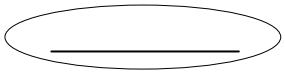
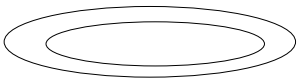
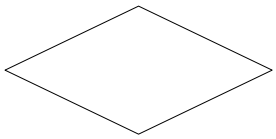
Gambar 2.2 Format Kamus Data

2.3.7 ERD (*Entity Relationship Diagram*)

Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram* (ERD).ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika.ERD digunakan untuk permodelan data rasional.ERD adalah sebuah diagram yang menggambarkan model relasi antar rancangan data tersimpan (file) atau bentuk logika yang dipakai analisis dan desain suatu sistem informasi. Model relasi ini diperlukan untuk menggambarkan struktur data dan relasi antar data (Rosa A. S., 2011).

Berikut ini adalah symbol-simbol yang digunakan padda ERD :

Table 2.4 Simbol *Entity Relathionship Diagram* (ERD)

Entitas / *entity* 	Entitas merupakan data inti yang akan disimpan, bakal tabel pada basis data.
Atribut 	*Field* atau kolom data yang butuh disimpan dalam suatu entitas.
Atribut Kunci Primer 	*Field* atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses *record* yang diinginkan.
Atribut Multinilai / *multivalue* 	*Field* atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu
Relasi 	Relasi yang menghubungkan antar entitas.
Asosiasi / *Association*	Penghubung antar relasi dan entitas.

2.3.8 Basis Data (*database*)

Sistem basis data adalah sisteem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan.Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat(Rosa A. S., 2011).(Indrajani,2009), menambahkan pendapat, bahwa

basis data merupakan kumpulan terpadu dari elemen data logis yang saling berhubungan. Basis data mengkonsolidasi banyak catatan yang sebelumnya disimpan dalam file terpisah. Dari beberapa pendapat tersebut dapat disimpulkan bahwa basis data adalah sekumpulan informasi yang berhubungan dengan subjek atau tujuan tertentu seperti melacak pemesanan konsumen, *maintaining inventory* atau pada umumnya mengumpulkan semua data yang berhubungan pada satu tempat. Data tersebut dapat disimpan, dimanipulasi dan digunakan dalam banyak cara. (Linda Marlinda,2004) berpendapat bahwa terdapt istilah-istilah yang digunakan dalam basis data yakni sebagai berikut:

a. Entitas/*Entity*

Suatu objek yang dapat dibedakan dengan objek lainnya yang dapat diwujudkan didalam basis data.

b. *Atribute/field*

Karakteristik entitas tertentu.

c. *Data Value*

Merupakan data *actual* atau informasi yang disimpan ditiap data elemen atau *atribute*. Isi *atribute* disebut nilai data.

d. *Record/Tuple*

Kumpulan isi elemen data yang saling berhubungan menginformasikan tentang suatu entitas secara lengkap.

e. *File*

Kumpulan *record* sejenis yang mempunyai panjang elemen dan *atribute* yang sama, namun berbeda-beda data *value*-nya.

f. Kunci elemen data

Sebagai tanda pengenalan yang secara unik mengidentifikasikan entitas dari suatu kumpulan entitas.

g. *Database management system*

File yang saling berkaitan dengan program untuk pengelolaannya.

Didalam *database* juga terdapat beberapa *atribute-atribute key (relational key)* yang diantaranya adalah sebagai berikut:

1. *Candidate Key*

   Merupakan *atribute* yang berbeda didalam relasi yang biasanya mempunyai nilai-nilai unik.

2. *Primary Key*

   Merupakan *candidate key* yang dipilih untuk mengidentifikasi entitas secara unik.

3. *Foreign Key*

   Merupakan *atribute* dengan *domain* yang sama, yang menjadikan ciri utama dari sebuah relasi tetapi pada relasi lain *atribute* tersebut hanya *atribute* biasa.

4. *Alternative Key*

   Merupakan *candidate key* yang tidak dipilih sebagai *primary key*.

Pada model data *relational*, hubungan antar *file* direlasikan dengan kuci relasi (*relational key*) yang merupakan kunci utama dari masing-masing *file.* Relasi antar dua tabel dapat dikategorikan menjadi tiga macam. Demikian pula untuk membantu menggambarkan relasi secara lengkap terdapat juga beberapa relasi dalam hubungan atribut yang ada didalam satu *file* atau dua *file*, yaitu sebagai berikut :

a. *One to one relationship* dua *file*

   Hubungan antar *file* pertama dengan *file* kedua adalaah satu berbanding satu. Seperti pada pelajaran privat, dimana satu guru hanya mengajar satu siswa. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukan tabel dan relasi antar keduanya diwakilkan dengan tanda panah tunggal.

b. *One to many relationship* dua *file*

   Hubungan antara *file* pertama dengan *file* kedua adalah satu berbanding banyak atau dapat pula dibalik, banyak lawan satu. Seperti pada sistem pengajaran disekolah, dimana satu guru mengajar banyak

siswa dan banyak siswa hanya diajar oleh satu guru. Hubungan tersebut dapat digambarkan dengan tanda lingkar untuk menunjukan tabel dan relasi antar keduanya dengan tanda panah ganda untuk menunjukan hubungan banyak tersebut.

c. *Many to many relationship* dua *file*

Hubungan antara *file* pertama dengan *file* kedua adalah banyak berbanding banyak. Seperti pada sistem pengajaran diperguruan tinggi, dimana banyak dosen mengajar satu mahasiswa dan mahasiswa diajar oleh banyak dosen. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukan tabel dan relasi antar keduannya diwakilkan dengan tanda panah ganda untuk menunjukan hubungan banyak tersebut.

d. Relasi *one to one* dua atribut dalam satu *file.*

Hubungan antar satu atribut dengan atribut yang lain dalam satu *file* yang sama mempunyai hubungan satu lawan satu. Misalnya atribut nomor pegawai yang unik dan atribut nomor KTP pegawai tersebut mempunyai hubungan satu lawan satu. Satu nomor pegawai hanya satu nomor KTP, tidak ada yang berganda.

Setelah itu, dalam perancangan *database* terdapat sistem kode yang digunakan untuk mengklasifikasikan data, memasukan data kedalam koputer dan untuk mengambil bermacam-macam informasi yang berhubungan dengannya. Kode dapat dibentuk dari kumpulan angka, huruf dan karakter-karakter khusus. Menurut (Jogiyanto,2005) beberapa tipe kode yang digunakan diantaranya sebagai berikut:

a. Kode Mnemonik (*mnemonic code*)

Kode mnemonik digunakan untuk tujuan supaya mudah diingat dengan dasar singkatan.

b. Kode Urut (*sequential code*)

Kode urut disebut juga kode seri, merupakan kode yang nilainya urut antara satu kode dengan kode berikutnya.

c. Kode Blok (*block code*)

Kode blok mengklasifikasikan *item* kedalam kelompok blok tertentu yang mencerminkan suatu klasifikasi tertentu atas dasar pemakaian maksimum yang diharapkan.

d. Kode Grup (*group code*)

Kode grup merupakan kode yang berdasarkan *field-field* dan taip-tiap *field* kode memiliki arti.

e. Kode Desimal (*Decimal code*)

Kode desimal mengklasifikasikan kode atas dasar 10 unit angka desimal dimulai dari angka 0 sampai dengan angka 9 atau dari 00 sampai dengan 99 tergantung dari banyaknya kelompok.

2.4 Perangkat Lunak Pendukung

Perangkat lunak pendukung yaitu *software* yang akan digunakan untuk menganalisa dan merancang sistem yang diusulkan, diantaranya seperti berikut:

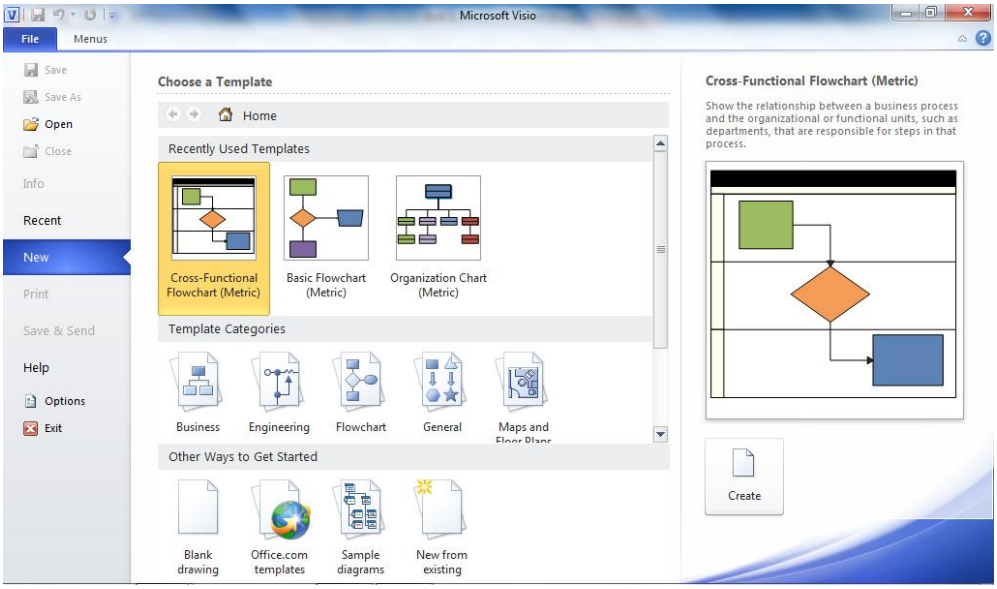
2.4.1 *Microsoft Visio 2010*

Microsoft Visio (atau sering disebut Visio) adalah sebuah program aplikasi computer yang sering digunakan untuk membuat diagram, diagram alir (*flowchart*), *bainstrom*, dan skema jaringan yang dirilis oleh *Microsoft Corporation.*Aplikasi ini menggunakan grafik vector untuk membuat diagram-diagramnya. (Wikipedia Bahasa Indonesia)

Ada lima langkah utama untuk membuat sebuah diagram di *microsoft visio 2010* yaitu :

1. Tentukan *template* yang sesuai.
2. Buat *shape* pada halaman (*page*).
3. Buat *connector* untuk menyambung antar *shape.*
4. Buat *text* (keterangan *shape* atau *connector*).
5. Simpan dokumen *visio* tersebut.

Seperti aplikasi *microsoft* lainnya, lingkungan kerja *visio* terdiri dari menu utama dan *toolbar*, namun perbedaannya terletak pada tampilan lembar kerja yang terdiri atas 2 jendela utama yaitu jendela pilihan *toolbox* disebelah kiri dan halaman pengerjaan disebelah kanan. Pada tampilan awal *visio*, sebelah kiri diisi jenis-jenis kategori dan sebelah kanan diisi dengan jenis *template* dari tiap kategori seperti ditunjukan pada gambar berikut :



Gambar 2.3 Tampilan Awal *Microsoft Visio 2010*

Microsoft visio menyediakan sembilan pilihan menu utama, antara lain sebagai berikut :

1. *File*

   Untuk mengatur yang berhubungan dengan *file*, seperti membuat *file* baru, membuka *file*, menutup *file*, *mengatur* file, dan lain-lain.
2. *Edit*

   Digunakan untuk proses pengeditan seperti *copy*, *paste*, *delete* dan lain-lain.
3. *View*

   Digunakan untuk mengatur tampilan lingkungan kerja *visio.*
4. *Insert*

Digunakan untuk menyisipkan objek, gambar, simbol, komentar dan lain-lain.

5. *Format*

   Digunakan untuk mengatur halaman pengerjaan.

6. *Tools*

   Berhubungan dengan fasilitas tambahan yang disediakan *visio* seperti *export to database, micro, ruler & grid, report* dan sebagainya.

7. *Shape*

   Digunakan untuk mengatur objek *shape* pada halaman pengerjaan.

8. *Window*

   Digunakan untuk mengatur jendela *visio.*

9. *Help*

   Digunakan untuk bantuan penggunaan *visio.*

Selain itu *visio* menyediakan beberapa *template* untuk membantu dalam pembuatan diagram. Untuk menggunakan *template* dari *visio,* dapat dilakukan dengan memilih menu *file-new-getting started,* maka akan muncul *window* baru.

Pada bagian *template categories* berisi daftar macam-macam tipe *drawing* yang telah dikelompokan berdasarkan fungsi dan keperluanya. Masing-masing tipe *drawing* ini memiliki anggota (*layout drawing* yang lebih spesifik) yang dapat dipilih pada *window template,* setiap *drawing* dalam *visio* menyediakan dalam 2 versi pengukuran yaitu *US Units dan Matric. Visio* memiliki 8 macam tipe *drawing*, yaitu:

1. *Business*

   Menggambarkan proses bisnis perusahaan. *Template* yang disediakan adalah *brainstroming, diagram, organization chart, pivot diagram, audit diagram, basic flowchart, cause and effect diagram, chart and graphs, cross functional flowchart, data flow diagram, epc diagram, fault tree analysis diagram, ITIL diagram, marketing charts and*

diagram, organization chart wizard, value stream map, TQM diagram, and work flow diagram.

2. *Engineering*

   Menggambarkan diagram yang berhubungan dengan bidang teknik.*Template* yang disediakan adalah *basic electrical, circuits and logic, industrial control systems, systems, fluid power, piping and instrumentation diagram, process flow diagram* dan *part and assembly drawing.*

3. *Flowchart*

   Menggambarkan diagram alir proses dan data. *Template* yang disediakan antara lain *basic flowchart, cross functional flowchart, data flow diagram, IDEF0 diagram, work flow diagram* dan *SDL diagram.*

4. *General*

   Menggambarkan serangkaian proses diagram umum. *Template* yang disediakan yaitu *basic flowchart, cross functional flowchart, data flow diagram, block diagram,* dan*block diagram with perspective.*

5. *Maps and floor plans*

   Menggambarkan denah pembangunan peta. *Template* yang disediakan antara lain*directional map, directional map 3d, electrical and telecom plan, floor plan, home plan, hvac control login diagram, hvac plan, office layout, plant layout, plumbing and piping diagram, reflected ceiling plan, security and access plan, site plan,* dan *space plan.*

6. *Network*

   Menggambarkan diagram yang berhubungan dengan jaringan. *Template* yang disediakan antara lain *active directory, besic network diagram, LDAP directory, conceptual web site, web site map*dan *rack diagram.*

7. *Schedule*

Menggambarkan diagram yang berhubungan dengan penjadalan projek. *Template* yang disediakan antara lain*calender, gantt chart, pert chart,* dan *timeline.*

8. *Software and database*

   Menggambarkan antar muka tampilan diagram dalam pembuatan perangkat lunak dan *database*. *Template* yang disediakan antara lain *COM and OLE, conceptual web site, data flow model diagram, enterprise application, jackson, program struckture, ROOM, UML model diagram, database model diagram, express-G,website map, ORM diagram* dan *windows XP user interface.*

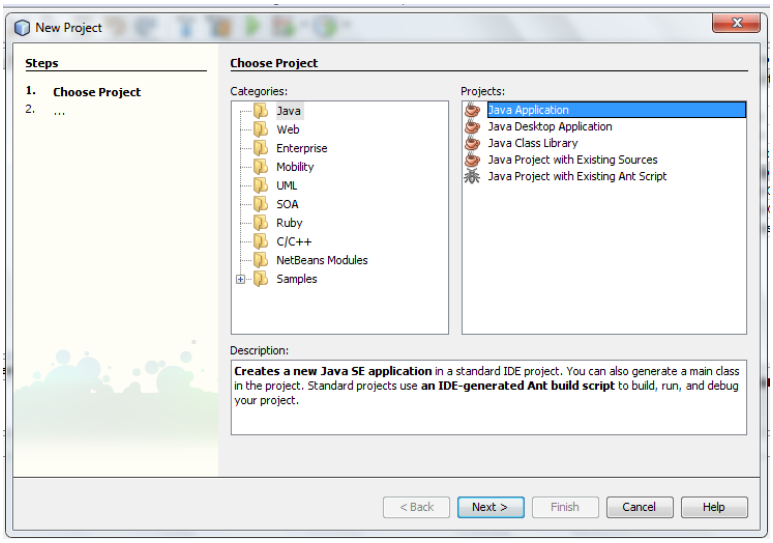
2.4.2 *Netbeans*

Netbeans adalah *integrated development environment (IDE)* berbasis *java* dari *sun microsystems* yang berjalan diatas *swing. Swing* sebuah sebuah teknologi *java* untuk pengembangan aplikasi *dekstop* yang berjalan diberbagai macam *platforms* seperti *indows, linux, mac OS X and solaris. Netbeans* merupakan *software development* yang *open source,* dengan kata lain *software* ini dibawah pengembangan bersama, bebas biaya.

Netbeans memiliki kelebihan-kelebihan seperti program yang digunakan gratis, *netbeans* sangat kompetibel dan *swing* karena memang langsung dikembangkan oleh *sun* yang notabennya sebagai pengembang *swing.* Selain itu *netbeans* juga memiliki kekurangan seperti *netbeans* mematenkan *source* untuk *java* yang sedang dikerjakan dalam sebuah *generated code,* sehingga *programmer* tidak dapat mengeditnya secara manual. Selain *netbeans* hanya men-*support* satu pengembangan *java* yaitu *swing.* Berikut ini akan dijelaskan langkah-langkah dalam menjalankan *netbeans.*

a. Untuk memulai, buka dahulu program *netbeans* yang ada dikomputer.
b. Membuat *project* baru, langkah-langkahnya sebagai berikut:
   1. Pilih *file.*
   2. *New project.*

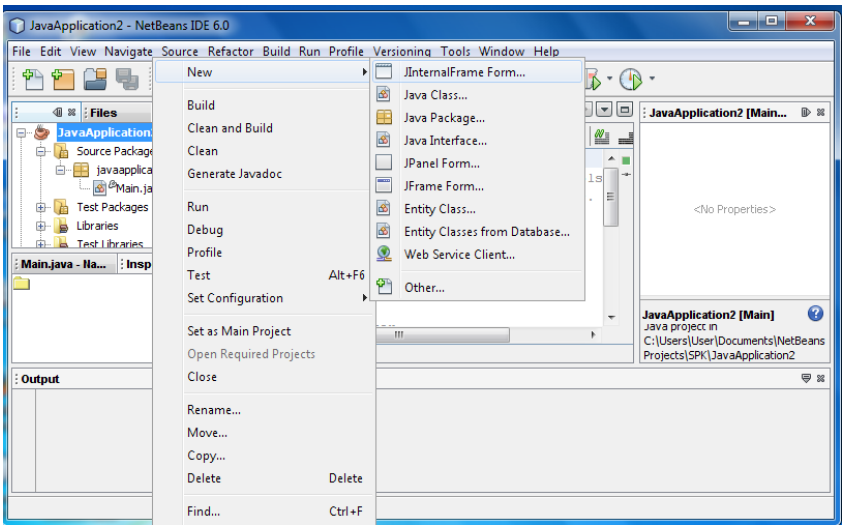
3. Pada *category:* pilih *java.*
4. Pada*project:* pilih *java application.*
5. *Click* tombol *next* pada *project name:* ketik nama yang diinginkan.
6. Pada *project location: click browse* untuk memilih *folder* dimana akan menyimpan *project.*
7. *Checklist* pada *set as main project.*
8. *Checklist* juga pada *create main class.*
9. Kemudian *click finish.*Berikut tampilan membuat *project* baru.



Gambar 2.4 Tampilan Membuat *Project* Baru

c. Membuat *form*, langkah-langkahnya sebagai berikut :

1. *Click* kanan pada nama *project* Anda disisi kiri yang ada pada *project explorer* pilih *jframe from.*
2. Kemudian pada *class name :* ketik nama *form* misalnya *frmButton.*
3. Pada *location :* pilih *source packages (*ini pilihan *default.*
4. Pada *package :* pilih *pallet button (*sesuai nama *projrct).*
5. Setelah itu *click finish.*



Gambar 2.5 Tampilan Membuat *Form*

d. Menambahkan kode program pada *main.java*

Karena pada saat membuat *project* pertama sekali di *checklist* pada *create main class* dan *set as main project*, maka yang selalu dijalankan pertama sekali ketika program di *run* adalah *main.java*, dari program inilah dipanggil *form frmbutton (*nama *form/class name)*, langkah-langkahnya sebagai berikut.

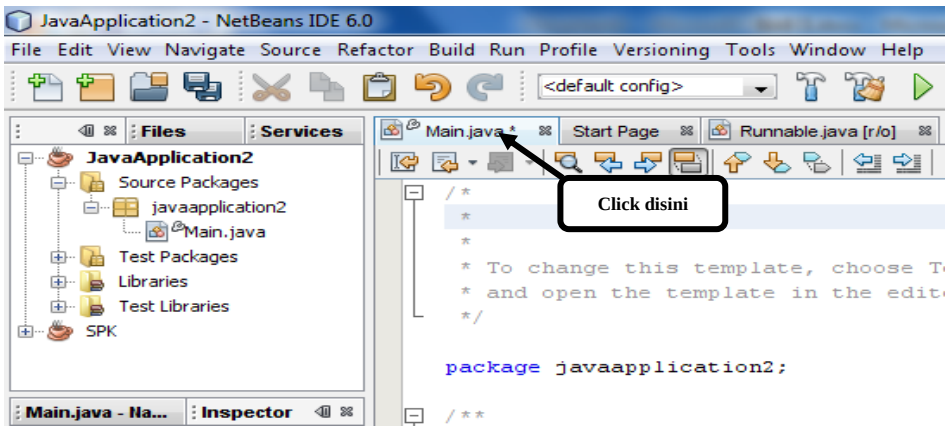
1. Buka *code main.java* dengan mengklik pada *tab main.java* yang ada dibawah *toolbar.*
2. Kemudian tambahkan kode pada *class* utama (*class main*).

```
Public static void maib (String[]args){

//TODO code application login here

New frmButton().setVisible(true);

}
```



Gambar 2.6 Tampilan Menambahkan Kode

e. Setelah itu coba jalankan program dengan menekan tombol F6 yang terdapat pada *toolbar,* lalu akan tampil *form* yang masih kosong, klik pada suduk *form* (x) untuk menutup *form* yang sedang *running* dan kembalilah *edit form* dan tambahkan beberapa sesuai dengan kebutuhan yang diinginkan.

2.4.3 SQL (*Structured Query Language*)

SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus.(Rosa A.S, 2011)

SQL tidak terbatas hanya untuk mengambil data (*query*), tetapi juga menciptakan table, menghapus table, menambahkan data ke table, menghapus data ditabel, mengganti data ditabel dan berbagai operasi lain.(Abdul Kadir, 2014)

SQL mulai berkembang pada tahun 1970-an. SQL mulai digunakan sebagai standar resmi pada tahun1986 oleh ANSI (*American National Standards Institute*) dan pada tahun 1987 oleh ISO (*International Organization for Standardization*) dan disebut sebagai SQL-86.