

BAB II

LANDASAN TEORI

Berikut ini akan dijelaskan beberapa pengertian dan teori yang berhubungan dengan penelitian ini meliputi pengertian dasar yang berkaitan dengan judul, teori umum pengembangan aplikasi dan sekilas teori tentang pemrograman yang dipakai serta database yang digunakan.

Adapun beberapa jurnal skripsi yang akan digunakan sebagai acuan dalam penyusunan laporan penelitian ini yaitu sebagai berikut:

1. Penelitian yang dibangun dengan menggunakan metodologi SSAD (*Structured System Analysis and Design*). Penelitian ini dilakukan dengan menggunakan metodologi *prototipe* dengan beberapa tools yang digunakan yaitu diagram konteks, diagram alir data, kamus data, dan spesifikasi proses. Penelitian ini akan melakukan rancangan bangun server jaringan e-kiosk (*multi client e-kiosk*) yang dapat melakukan fungsi memberi informasi dan akses akademik dilingkungan PRIIK kepada civitas akademik. (Nurwarsito, 2013).
2. Subyek penelitian yang membahas tentang analisis arsitektur jaringan *client server database* terpusat terdapat berbagai macam tahap-tahap diantaranya adalah pengumpulan data meliputi studi literatur, observasi dan eksperimen. Sistem disusun dengan prosedur yang mencakup identifikasi masalah, analisis kebutuhan, merancang konsep, implementasi sistem selanjutnya ke tahap pengujian sistem untuk memperoleh hasil kelayakan sebuah arsitektur *client server database* terpusat. Dari hasil penelitian dapat diketahui bahwa arsitektur client server menggunakan database terpusat dapat memaksimalkan penggunaan jaringan komputer di SMP Muhammadiyah Purwodadi, serta memudahkan pengelola jaringan dalam memantau terhadap aktifitas pengguna jaringan diinstansi sekolah tersebut. (Nuril Anwar, 2013),

2.1 Teori Dasar

Teori-teori dasar yang digunakan dalam penelitian akan dijelaskan satu persatu oleh penulis yaitu sebagai berikut.

2.1.1 Sistem

Sistem adalah suatu himpunan dari berbagai bagian atau elemen yang saling berhubungan secara terorganisasi berdasarkan fungsi-fungsinya menjadi satu kesatuan. Berdasarkan definisi tersebut dapat diketahui bahwa sistem mengandung dua macam konotasi utama yaitu suatu benda atau entitas dan suatu proses atau metode atau cara untuk mencapai tujuan. (Bambang Hartono ,2013)

Beberapa karakteristik dari sistem yaitu sebagai berikut:

1) Komponen Sistem

Suatu sistem terdiri dari komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem dapat berupa subsistem atau bagian dari sistem. Setiap subsistem mempunyai sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2) Batasan Sistem

Batasan sistem merupakan daerah yang membatasi suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan sistem dipandang sebagai suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup sistem tersebut.

3) Lingkungan Luar Sistem

Lingkungan luar sistem adalah apapun diluar batas suatu sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan juga merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan demikian harus tetap dijaga dan dipelihara. Sedangkan yang tidak menguntungkan harus dikendalikan, karena akan mengganggu kelangsungan sistem tersebut.

4) Penghubung Sistem

Penghubung merupakan penghubung dari satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke sub sistem lainnya.

5) Masukan Sistem

Masukan adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan dan masukan sinyal. Masukan perawatan adalah energi yang dimasukkan supaya sistem tersebut dapat dimasukan. Masukan sinyal adalah energi yang diproses untuk mendapatkan keluaran. Sebagai contoh didalam sistem komputer, program adalah masukan perawatan yang digunakan untuk mengoperasikan komputernya dan data adalah inputan sinyal yang akan diolah menjadi informasi.

6) Keluaran Sistem

Keluaran adalah hasil dari energi yang diolah dan yang diklasifikasikan menjadi keluaran yang berguna. Keluaran dapat merupakan masukan bagi subsistem lainnya.

7) Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolahan yang akan merubah masukan menjadi keluaran. Suatu sistem akan mengolah masukan berupa bahan baku dan bahan lainnya menjadi bahan jadi.

8) Sasaran Sistem.

Suatu sistem pasti mempunyai tujuan atau sasaran yang akan dicapai. Sasaran dari suatu sistem dapat menentukan masukan yang dibutuhkan dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil jika mengenai sasaran yang telah ditentukan.

Sistem diklasifikasikan dari beberapa sudut pandang diantaranya sebagai berikut:

1) Sistem diklasifikasikan sebagai sistem abstrak dan sistem fisik.

Sistem abstrak sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sistem fisik merupakan sistem yang ada secara fisik.

2) Sistem diklasifikasikan sebagai sistem alamiah dan sistem buatan manusia.

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Sistem buatan yang melibatkan interaksi manusia dengan mesin.

3) Sistem diklasifikasikan sebagai sistem tertentu dan sistem tak tentu.

Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diramalkan. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksikan.

4) Sistem diklasifikasikan sebagai sistem tertutup dan sistem terbuka.

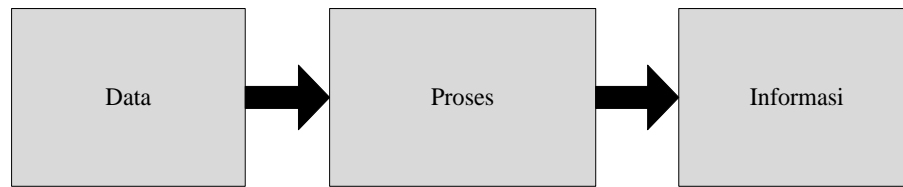
Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem terbuka merupakan sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya.

2.1.2 Informasi

Informasi pada dasarnya adalah sehimpunan data yang telah diolah menjadi sesuatu yang memiliki arti dan kegunaan lebih luas. *Lippeveld, Sauerborn, dan Bodart* mendefinisikan informasi sebagai himpunan fakta atau data yang memiliki makna.

Sedangkan data sendiri adalah hasil pengukuran atau pencatatan terhadap fakta tentang sesuatu, keadaan, tindakan atau kejadian. Transformasi data menjadi suatu informasi dilakukan di dalam sebuah sistem, dimana data yang merupakan sebuah masukan kemudian diolah atau diproses oleh sistem menjadi suatu keluaran yang berupa informasi. (Bambang Hartono, 2013)

Untuk lebih jelasnya proses transformasi data menjadi informasi dapat dilihat pada gambar berikut:



Gambar 2.1 Transformasi data menjadi informasi

Sumber : (Abdul Kadir, 2014)

Pada umumnya kualitas informasi bergantung pada 3 (tiga) hal yang sangat domain yaitu:

1) Informasi harus akurat.

Sebuah informasi harus akurat karena dari sumber informasi hingga penerima informasi kemungkinan banyak terjadi gangguan yang dapat mengubah atau merusak informasi tersebut. Informasi dikatakan akurat apabila informasi tersebut tidak bias atau menyesatkan, bebas dari kesalahan-kesalahan dan harus jelas mencerminkan maksudnya.

2) Informasi harus tepat waktu.

Informasi yang dihasilkan dari suatu proses pengolahan data, datangnya tidak boleh terlambat (usang). Informasi terlambat tidak akan mempunyai nilai yang baik, karena informasi merupakan landasan dalam pengambilan keputusan.

3) Informasi harus relevan.

Informasi dikatakan berkualitas jika relevan bagi pemakainya. Hal ini berarti bahwa informasi tersebut harus bermanfaat bagi pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda.

2.1.3 Sistem Informasi

Sistem informasi adalah seperangkat komponen yang saling berhubungan, yang bekerja untuk mengumpulkan dan menyimpan data serta mengolahnya menjadi informasi untuk digunakan. (Bambang Hartono, 2013) Komponen sistem informasi adalah sebagai berikut :

1) Blok Masukan

Input mewakili data yang masuk ke dalam sistem informasi, input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2) Blok Model

Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tentu untuk menghasilkan keluaran yang diinginkan.

3) Blok Keluaran

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta sesuai pemakai sistem.

4) Blok Teknologi

Teknologi merupakan “kotak alat” (*tool box*) dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan.

5) Blok Basis Data

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan diperangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

6) Blok Kendali

Banyak hal yang dapat merusak sistem informasi, seperti misalnya bencana alam, api, temperatur, air, debu, kecurangn-kecurangan, kegagalan-kegagalan sistem itu sendiri. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat lagsung cepat diatasi.

2.1.4 Basis Data

Database merupakan pangkalan data (tempat data) yang disusun sedemikian rupa dengan tujuan database tersebut efisien dan efektif, serta mengikuti aturan alat desain database terlepas dari perangkat lunak pemrograman yang digunakan. (Yuniar Supardi, 2011)

Database didefinisikan sebagai kumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat dimanipulasi, diambil, dan dicari secara cepat. (Budi Raharjo, 2011)

Adapun elemen-elemen penyusun basis data adalah sebagai berikut:

1. Data Value (Nilai Data)

Data Value adalah data aktual atau informasi yang disimpan pada tiap data, elemen, atau atribut. Atribut nama pegawai menunjukkan tempat di mana informasi nama karyawan disimpan, nilai datanya misalnya adalah Anjang, Arif, Suryo, dan lain-lain yang merupakan isi data nama pegawai tersebut.

2. Penyimpanan

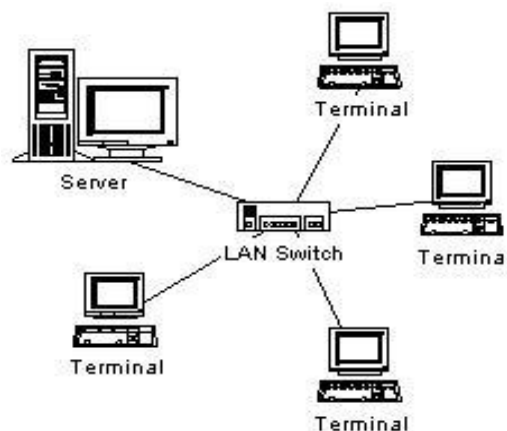
- 1) *File/Tabel* : Kumpulan *record* sejenis yang mempunyai panjang elemen yang sama, atribut yang sama, namun berbeda nilai datanya.
- 2) *Record/Tuple* merupakan kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu entitas secara lengkap. Satu *record* mewakili satu data atau informasi.
- 3) *Entitas* adalah sekumpulan objek yang terdefiniskan yang mempunyai karakteristik sama dan bisa dibedakan satu dengan lainnya. Objek dapat berupa barang, orang, tempat, suatu kejadian atau konsep yang informasinya direkam.
- 4) *Atribut* adalah deskripsi data yang bisa mengidentifikasi entitas yang membedakan entitas tersebut dengan entitas yang lain. Seluruh atribut harus cukup untuk menyatakan identitas obyek, atau dengan kata lain, kumpulan atribut dari setiap entitas mengidentifikasi keunikan individu.

- 5) *Primary key* adalah suatu kolom (field) yang menjadi titik acuan pada sebuah tabel, bersifat unik dalam artian tidak ada satu nilai pun yang sama atau kembar dalam tabel tersebut, dan dalam satu tabel hanya boleh ada satu primary key.
- 6) *Foreign key* atau disebut juga kunci relasi adalah suatu kolom dalam tabel yang digunakan sebagai “kaitan” untuk melengkapi satu hubungan yang didapati dari tabel induk, dan biasanya hubungan yang terjalin antar tabel adalah satu ke banyak (*one to many*).
- 7) *Index* adalah struktur basis data secara fisik, yang digunakan untuk optimalisasi pemrosesan data dan mempercepat proses pencarian data.

2.1.5 Jaringan Komputer

Menurut Jurnal Skripsi terdahului (Ahmad Badu,2012), Jaringan komputer (*computer network*) atau sering disingkat jaringan adalah hubungan dua simpul (umumnya berupa komputer) atau lebih yang tujuan umumnya adalah untuk melakukan pertukaran data. Dalam prakteknya, jaringan komputer memungkinkan untuk melakukan berbagi perangkat lunak, perangkat keras, dan bahkan berbagi kekuatan pemrosesan. Ada beberapa jenis jaringan computer antara laian:

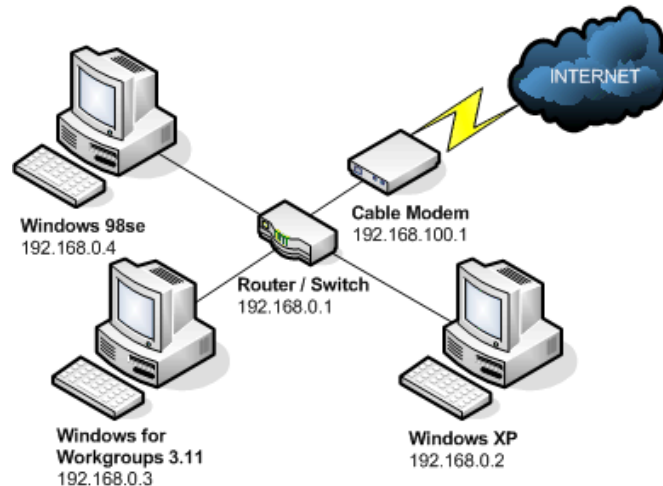
1) *Local Area Network (LAN)*; merupakan jaringan milik pribadi di dalam sebuah gedung / kantor atau kampus yang berukuran sampai beberapa kilometer.



Gambar 2.2 Local Area Network (LAN)

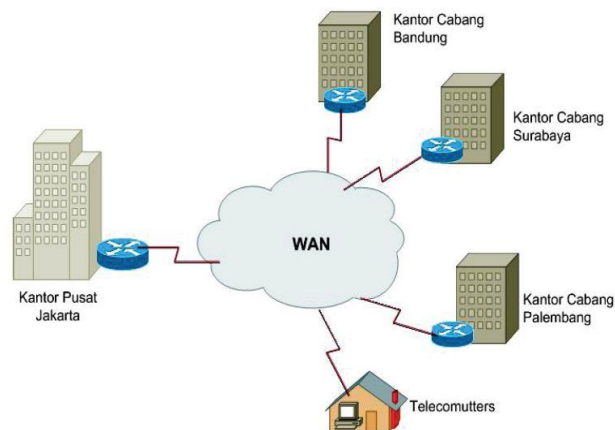
2) *Metropolitan Area Network (MAN)*; merupakan versi LAN yang berukuran lebih besar dan biasanya menggunakan teknologi yang sama dengan LAN. MAN dapat

mencakup kantor-kantor perusahaan yang letaknya berdekatan atau juga sebuah kota dan dapat dimanfaatkan untuk keperluan pribadi (swasta) atau umum. MAN mampu menunjang data dan suara, bahkan dapat berhubungan dengan jaringan televisi kabel.



Gambar 2.3 Metropolitan Area Network (MAN)

3) *Wide Area Network (WAN)*; jangkauannya mencakup daerah geografis yang luas, seringkali mencakup sebuah negara bahkan benua. WAN terdiri dari kumpulan mesin-mesin yang bertujuan untuk menjalankan program-program (aplikasi) pemakai. Misalnya, jaringan yang menghubungkan ATM (Anjungan Tunai Mandiri), Internet.



Gambar 2.4 *Wide Area Network (WAN)*

2.2 Teori Pengembangan Sistem

Untuk mengembangkan suatu sistem informasi, kebanyakan perusahaan menggunakan metodologi yang sering disebut metodologi pengembangan sistem. Metodologi itu sendiri adalah suatu proses standar yang diikuti organisasi untuk melaksanakan seluruh langkah yang diperlukan untuk menganalisa, merancang, mengimplementasikan, dan melihat sistem informasi. (Abul kadir, 2014)

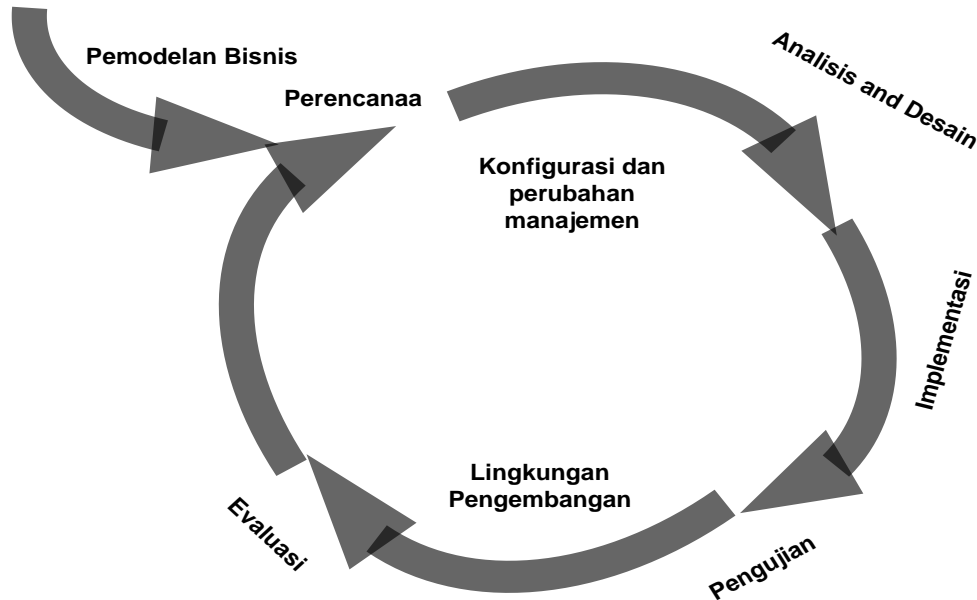
Beberapa prinsip dalam proses pengembangan sistem informasi adalah sebagai berikut.

- 1) Sistem yang dikembangkan adalah untuk manajemen.
- 2) Sistem yang dikembangkan adalah investasi modal yang besar.
- 3) Sistem yang dikembangkan memerlukan orang terdidik.
- 4) Tahapan kerja dan tugas-tugas yang harus dilakukan dalam proses pengembangan sistem.
- 5) Proses pengembangan sistem tidak harus urut.
- 6) Jangan takut membatalkan proyek.
- 7) Dokumentasi harus ada untuk pedomannya dalam pengembangan sistem.

Dalam pengembangan sistem ini penulis menggunakan analisis berorientasi objek. Orientasi objek merupakan suatu teknik dalam menyelesaikan masalah yang kerap muncul dalam perancangan ataupun pengembangan perangkat lunak.

Teknik ini merupakan titik kulminasi dalam menemukan cara yang efektif dalam membangun sistem dan menjadi metode yang paling banyak dipakai oleh para perancang dan pengembang perangkat lunak saat ini. Alat yang digunakan dalam metodologi ini salah satunya adalah UML (*Unified Modeling Language*). UML (*Unified Modeling Language*) adalah alat komunikasi yang konsisten dalam mensupport para pengembang sistem saat ini. Sebagai perancangan sistem, mau tidak mau pasti akan menjumpai UML, baik kita sendiri yang membuat atau sekedar

membaca diagram UML buatan orang lain. (Rosa A.S & M.Shalahuddin, 2015)
Tahapan yang umum dilakukan dalam pengembangan adalah sebagai berikut.



Gambar 2.5 Gambar Tahapan Metodologi Penelitian

Keterangan :

1. **Pemodelan Bisnis**

Pemodelan bisnis adalah gambaran dari realita yang simpel dan dituangkan dalam bentuk pemetaan dengan aturan tertentu. Pemodelan bisnis digunakan untuk menggambarkan desain sistem.

2. **Perencanaan**

Perencanaan merupakan tahapan studi untuk menentukan kebutuhan user, studi kelayakan baik secara teknis maupun teknologi serta penjadwalan pengembangan suatu proyek sistem informasi yang akan dilakukan.

3. **Analisis dan Desain**

Tahapan analisis ini merupakan tahap untuk mengenali segenap permasalahan yang muncul pada sistem yang telah digunakan, mengenali komponen-komponen sistem, obyek-obyek, hubungan antar obyek dan sebagainya.

Sedangkan tahapan desain merupakan tahap yang dilakukan untuk mencari solusi dari permasalahan yang didapat dari tahap analisa.

4. Implementasi

Tahap implementasi ini dimulai dengan proses pemilihan perangkat keras, penyusunan perangkat lunak aplikasi, melihat apakah sistem yang dibuat sudah sesuai dengan kebutuhan user atau belum.

5. Pengujian

Pengujian merupakan tahapan yang dilakukan untuk mengetahui sistem yang telah dibuat sudah sesuai dengan kebutuhan yang telah ditargetkan sebelumnya. Sistem dinyatakan berkualitas apabila sudah sesuai dengan kebutuhan pengguna, bebas dari kesalahan, dan memiliki kegunaan bagi penggunanya.

6. Evaluasi

Tahapan ini dilakukan untuk mengevaluasi apakah sistem yang sudah dibuat sudah sesuai dengan yang diharapkan dan menghasilkan output yang akan dibutuhkan oleh pengguna sistem nantinya.

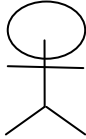
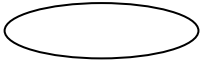
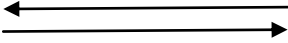
Agar dapat melakukan langkah-langkah sesuai dengan metodologi pengembangan analisis dan desain berorientasi objek, maka dibutuhkan alat dan teknik untuk melaksanakannya. Alat-alat yang digunakan dalam suatu metodologi umumnya berupa suatu gambar diagram atau grafik. Adapun alat pengembangan sistem yang digunakan yaitu sebagai berikut.

1) *Use Case Diagram*

Use case atau diagram use case merupakan pemodelan untuk melakukan kelakuan (behavior) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang dibuat. Secara umum, use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang hendak menggunakan fungsi-

fungsi itu. (Rosa AS. & M. Salahuddin, 2015) Simbol *Use-Case* dapat dilihat pada tabel 2.1 berikut:




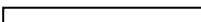

Tabel 2.1 Bagan *Use Case Diagram*

Simbol	Keterangan
ACTOR 	Mendefinisikan entitas diluar sistem yang memakai sistem.
USE CASE 	Gambaran fungsinalitas dari suatu sistem, sehingga pengguna mengerti kegunaan sistem yang akan dibangun
RELASI 	Menceritakan hubungan antara aktor dan <i>use case</i> sehingga diagram dapat di pahami

2) *Activity Diagram*

Diagram aktifitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis, atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan di sini bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh sistem. (Rosa AS. & M. Salahuddin, 2015) Penjelasan sambil dari bagan *activity diagram* dapat dilihat pada tabel 2.2

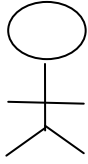
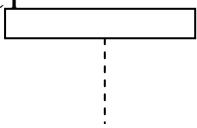

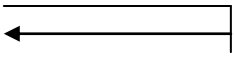
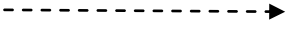
Tabel 2.2 Bagan *Activity Diagram*

Simbol	Keterangan
PARTITION 	Memperlihatkan dimana aliran berawal
RELASI 	Merupakan langkah atau aksi-aksi yang terjadi
Desicion 	Memperlihatkan dimana keputusan perlu diambil selama terjadi selama terjadi aliran kerja
FORK 	Merupakan percabangan dari sebuah aksi-aksi yang terjadi
FINAL NODE 	Memperlihatkan dimana aliran itu berakhir

3) *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *Use-Case* dengan mendeskripsikan waktu hidup objek. Oleh karena itu untuk menggambarkan diagram sekuen harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. *Sequence diagram* menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus berasosiasi dengan *use case*. *Sequence Diagram* memperlihatkan tahap demi tahap apa yang sebenarnya terjadi untuk menghasilkan sesuatu didalam *use case*. (Rosa A.S & M.Shalahudin, 2015) Penjelasan simbol bagan *sequence diagram* dapat dilihat pada tabel 2.3.


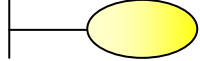
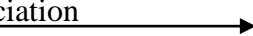

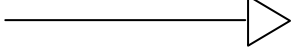
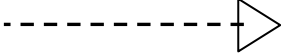
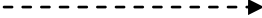
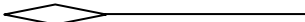

Tabel 2.3 Bagan *Sequence Diagram*

Simbol	Keterangan
ACTOR 	Prilaku didalam <i>sequence Diagram</i>
OBJECT 	Menambahkan Objek baru pada diagram
OBJECT MESSEGE 	Menggambarkan pesan (<i>message</i>) antar dua objek
MESSEGE TO SELF 	Menggambarkan pesan (<i>message</i>) yang menuju dirinya sendiri
RETURN MESSEGE 	Menggambarkan pengembalian dari pemanggilan prosedur

4) *Class Diagram*

Class Diagram atau diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif. (Rosa A.S & M.Shalahudin, 2013) Penjelasan simbol *class diagram* dapat dilihat pada tabel 2.5

Tabel 2.4 Bagan *Class Diagram*

Simbol	Keterangan
Class 	Menambahkan kelas baru pada diagram
Interface 	Menambahkan kelas antarmuka (<i>interface</i>) pada diagram
Association 	Menggambar relasi asosiasi
Association class 	Menghubungkan (<i>association class</i>) suatu relasi Asosiasi
Generalization 	Menggambarkan relasi generalisasi
Realize 	Menggambarkan relasi realisasi
Return message 	Menggambarkan pengembalian dari pemanggilan prosedur
Aggregation 	Menggambarkan relasi agregasi
Actor 	Menggambarkan aktor pada diagram Kelas

5) Relasi Antar Tabel

Model data relasional menggunakan sekumpulan tabel berdimensi dua (yang bisa disebut relasi tabel) dengan masing-masing tabel tersusun atas sejumlah baris dan kolom. (Abdul Kadir, 2014)

Beberapa jenis relasi antar tabel, yaitu sebagai berikut :

1) *1-1*

Hubungan antara lain *file* pertama dan *file* kedua adalah satu berbanding satu.

Contoh tabel_probadi dan tabel_pekerjaan

2) *1-M*

Hubungan antara *file* pertama dan *file* kedua adalah satu berbanding banyak atau dapat pula dibalik menjadi banyak lawan satu. Contoh tabel_pekerjaan dan tabel_bagian.

6) Kamus Data (KD)

Kamus data adalah suatu daftar elemen yang pada sistem perangkat lunak sehingga masukan dan keluaran dapat dipahami (memiliki standar cara penulisan). (Rosa AS & M. Salahuddin, 2015) Pembentukan kamus data didasarkan pada alur data yang terdapat pada DFD.

Format Kamus Data			
Nama database :			
Nama tabel :			
Primary Key :			
Foreign Key :			
Nama Field	Type	Size	Keterangan

Gambar 2.6 *Format* Kamus Data

7) Pengkodean

Pengkodean adalah suatu kode digunakan untuk tujuan mengklasifikasikan data, memasukkan data ke dalam komputer dan untuk mengambil bermacam-macam informasi yang berhubungan dengannya. Ada beberapa macam tipe dari kode yang digunakan didalam sistem informasi, diantaranya adalah:

1) Kode (*Memonic Code*)

Kode mnemonik digunakan untuk tujuan supaya mudah diingat. Kode mnemonic dibuat dengan dasar singkatan atau mengambil sebagian karakter dari item yang akan diwakili dengan kode ini. Misal "P" untuk Pria, "W" untuk Wanita, "JKT" untuk Jakarta. Umumnya kode *mnemonic* menggunakan huruf. Akan tetapi dapat juga menggunakan gabungan huruf dan angka, Dengan kode ini membaca satu data akan menjadi lebih mudah, contoh kode sederhana B01-JKT-COD yang berarti = kode barang - lokasi kirim - metode pembayaran. Kebaikan dari kode ini adalah mudah diingat dan kelemahannya adalah kode dapat terlalu panjang. Kode Urut (*Sequential Code*)

2) Kode Urut

Kode urut merupakan kode yang nilainya urut antara satu kode dengan kode berikutnya.

Contoh :

001 Kas.

002 Piutang dagang

003 Persediaan produk selesai

004 Persediaan produk dalam proses

2.5 Teori Perangkat Lunak

Berikut ini teori-teori perangkat lunak pengembangan aplikasi yang digunakan dalam penelitian ini adalah sebagai berikut:

2.5.1 *MySql*

MySql (bisa di baca Mai-es-ki-el) adalah suatu perangkat lunak database relasi (*Relational Database Managemen System* atau *RDBMS*), seperti halnya *ORACLE* dan *SQL Server*. *MySql* merupakan salah satu database yang populer dan mendunia. *MySql* bekerja dengan menggunakan *SQL Language* (*Structure Query*

Language) itu artinya *MySql* merupakan standar pengguna database di dunia untuk mengolah data. (Agus Saputra, 2013)

Menurut Bunafit Nugroho (2012), *Database MySQL* adalah *program database server* berbasis *DOS* yang perintah dasarnya adalah *Structured Query Language (SQL)* untuk memanipulasi data atau manajemen data di dalam database.

Pengembangan *MySql* dimulai pada tahun 1979 dengan tool database *UNREG* yang dibuat oleh Michael “*Monty*” Widenius untuk perusahaan *TcX* di Swedia. Kemudian pada tahun 1994 mulai mencari server *SQL* untuk mengembangkan *aplikasi web*. Mereka menguji beberapa *server* komersial namun semuanya masih terlalu lambat untuk tabel-tabel *TcX* yang besar.

Tahun 1995 David Axmark dari Detoro HB mencoba menekan *TcX* untuk merelease *MySql* di internet selain itu juga membuat dokumentasi *MySql* yang di-build untuk GNU konfigurasi utility. *MySql* 3.11.1 dipublikasikan di dunia pada tahun 1996 dan didistribusikan untuk *Linux* dan *Solaris*. Sekarang ini *MySql* bekerja untuk banyak *platform* serta tersedia *source codenya*. Beberapa kelebihan yang dimiliki MySQL adalah sebagai berikut:

- a. Bersifat *open source*
- b. Menggunakan bahasa *SQL (structured query language)*
- c. Kecepatan dalam pengolahan data
- d. Mudah untuk dipelajari
- e. Dapat dijalankan sebagai sistem operasi
- f. Dapat digunakan oleh beberapa user dalam waktu bersamaan.

Berikut adalah logo database *MySQL* :



Gambar 2.7 Logo *MySQL*

Perintah yang umum dilakukan dalam MySQL adalah sebagai berikut:

- a. *Select* untuk menampilkan data
- b. *Insert* untuk menambahkan data baru
- c. *Update* untuk mengubah data yang sudah ada
- d. *Delete* untuk menghapus data

Tipe data di *MySQL* terbagi menjadi beberapa kelompok, yaitu *numerik*, *string*, *date-and-time*, dan kelompok himpunan (*set* dan *enum*). Masing-masing *tipe field* memiliki batasan lebar dan ukurannya.

1. Tipe Numerik

Tipe data numerik digunakan untuk menyimpan data numeric (angka). Ciri utama data numeric adalah suatu data yang memungkinkan untuk dikenai operasi aritmatika seperti penambahan, pengurangan, perkalian dan pembagian. Berikut ini tipe field (kolom) di MySQL yang termasuk ke dalam kelompok tipe numerik:

a. *Tinyint*

Penggunaan : digunakan untuk menyimpan data bilangan bulat positif dan negatif.

Jangkauan :-128 s/d 127

Ukuran : 1 byte (8 bit).

b. *Smallint*

Penggunaan : digunakan untuk menyimpan data bilangan bulat positif dan negatif.

Jangkauan : -32.768 s/d 32.767

Ukuran : 2 byte (16 bit).

c. *Mediumint*

Penggunaan : digunakan untuk menyimpan data bilangan bulat positif dan negatif.

Jangkauan : -8.388.608 s/d 8.388.607

Ukuran : 3 byte (24 bit)

d. *Int*

Penggunaan : digunakan untuk menyimpan data bilangan bulat positif dan negatif.

Jangkauan : -2.147.483.648 s/d 2.147.483.647

Ukuran : 4 byte (32 bit).

e. *Bigint*

Penggunaan : digunakan untuk menyimpan data bilangan bulat positif dan negatif.

Jangkauan : $\pm 9,22 \times 10^{18}$

Ukuran : 8 byte (64 bit)

f. *Float*

Digunakan untuk menyimpan data bilangan pecahan positif dan negatif presisi tunggal.

Jangkauan : $3.402823466E+38$ s/d $-1.175494351E-38$, 0, dan $1.175494351E-38$ s/d $3.402823466E+38$.

Ukuran : 4 byte (32 bit).

g. *Double*

Penggunaan : digunakan untuk menyimpan data bilangan pecahan positif dan negatif presisi ganda.

Jangkauan : $-1.79...E+308$ s/d $-2.22...E-308$, 0, dan $2.22...E-308$ s/d $1.79...E+308$.

Ukuran : 8 byte (64 bit).

h. *Real*

Penggunaan : Merupakan sinonim dari DOUBLE.

i. *Decimal*

Penggunaan : digunakan untuk menyimpan data bilangan pecahan positif dan negatif.

Jangkauan : $-1.79...E+308$ s/d $-2.22...E-308$, 0, dan $2.22...E-308$ s/d $1.79...E+308$.

Ukuran : 8 byte (64 bit).

j. *Numeric*

Penggunaan : Merupakan sinonim dari DECIMAL.

2. Tipe Date dan Time

Tipe data date dan time digunakan untuk menyimpan data tanggal dan waktu.

Berikut ini tipe field (kolom) di MySQL yang termasuk ke dalam kelompok tipe date dan time:

a. *Date*

Penggunaan : digunakan untuk menyimpan data tanggal.

Jangkauan : 1000-01-01 s/d 9999-12-31 (YYYY-MM-DD)

Ukuran : 3 byte.

b. *Time*

Penggunaan : digunakan untuk menyimpan data waktu.

Jangkauan : -838:59:59 s/d +838:59:59 (HH:MM:SS)

Ukuran : 3 byte.

b. *Datetime*

Penggunaan : digunakan untuk menyimpan data tanggal dan waktu.

Jangkauan : '1000-01-01 00:00:00' s/d '9999-12-31 23:59:59'

Ukuran : 8 byte.

d. *Year*

Penggunaan : digunakan untuk menyimpan data tahun dari tanggal.

Jangkauan : 1900 s/d 2155

Ukuran : 1 byte.

3. Tipe String (Text)

Tipe data string digunakan untuk menyimpan data string (text). Ciri utama data string adalah suatu data yang memungkinkan untuk dikenai operasi aritmatika seperti penambahan, pengurangan, perkalian dan pembagian. Berikut ini tipe field (kolom) di MySQL yang termasuk ke dalam kelompok tipe string:

a. *Char*

Penggunaan : digunakan untuk menyimpan data string ukuran tetap.

Jangkauan : 0 s/d 255 karakter

b. *Varchar*

Penggunaan : digunakan untuk menyimpan data string ukuran dinamis.

Jangkauan : 0 s/d 255 karakter (versi 4.1), 0 s/d 65.535 (versi 5.0.3)

c. *Tinytext*

Penggunaan : digunakan untuk menyimpan data text.

Jangkauan : 0 s/d 255 karakter (versi 4.1), 0 s/d 65.535 (versi 5.0.3)

d. *text*

Penggunaan : digunakan untuk menyimpan data text.

Jangkauan : 0 s/d 65.535 (216 - 1) karakter

e. *Mediumtext*

Penggunaan : digunakan untuk menyimpan data text.

Jangkauan : 0 s/d 224 - 1 karakter

f. *Longtext*

Penggunaan : digunakan untuk menyimpan data text.

Jangkauan : 0 s/d 232 - 1 karakter

4. Tipe **BLOB (Biner)**

Tipe data blob digunakan untuk menyimpan data biner. Tipe ini biasanya digunakan untuk menyimpan kode-kode biner dari suatu file atau object. BLOB merupakan singkatan dari Binary Large Object. Berikut ini tipe field (kolom) di MySQL yang termasuk ke dalam kelompok tipe blob:

a. *Bit* (sejak versi 5.0.3)

Penggunaan : digunakan untuk menyimpan data biner.

Jangkauan : 64 digit biner

b. *Tinyblob*

Penggunaan : digunakan untuk menyimpan data biner.

Jangkauan : 255 byte

c. *Blob*

Penggunaan : digunakan untuk menyimpan data biner.

Jangkauan : 216 - 1 byte

d. *Mediumblob*

Penggunaan : digunakan untuk menyimpan data biner.

Jangkauan : 224 - 1 byte

e. *Longblob*

Penggunaan : digunakan untuk menyimpan data biner.

Jangkauan : 232 - 1 byte

5. Tipe Data yang Lain

Selain tipe data di atas, MySQL juga menyediakan tipe data yang lain. Tipe data di MySQL mungkin akan terus bertambah seiring dengan perkembangan versi MySQL. Berikut ini beberapa tipe data tambahan MySQL:

a. *Enum*

Penggunaan : Enumerasi (kumpulan data).

Jangkauan : Sampai dengan 65535 string.

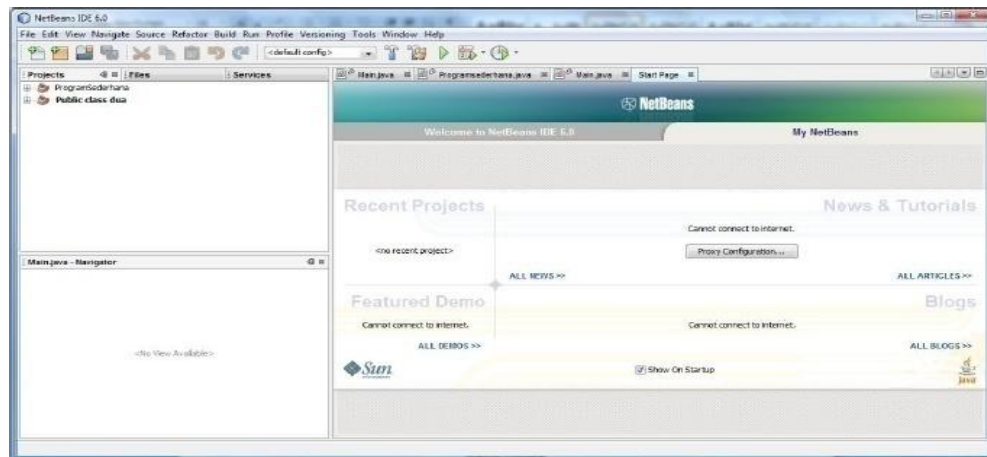
b. *Set*

Penggunaan : Combination (himpunan data).

Jangkauan : Sampai dengan 255 string anggota.


2.5.2 *Java (Netbeans IDE)*

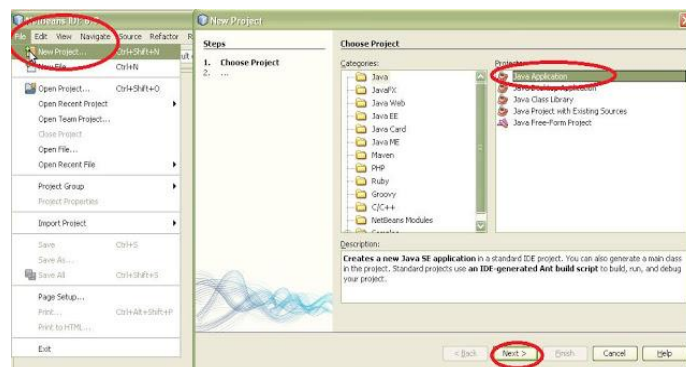
Java dibuat pada tahun 1995 di *sun Microsystem*. Kini *java* dibawah perusahaan oracle. *Java* merupakan bahasa pemrograman berorientasi objek dan serbaguna. Kode *java* di kompilasi dalam format yang disebut *bytecode* yang dapat dijalankan disemua komputer yang sudah dilengkapi dengan program *java interpreter* dan *java virtual mechine*. (Abdul Kadir, 2014). *Netbeans* merupakan salah satu proyek *open source* yang terdiri dari *netbeans IDE* dan *netbeans Platform* dan menyediakan paket lengkap dalam pemrograman dari pemrograman standar (Aplikasi *Desktop*), pemrograman *enterprise*, dan pemrograman perangkat *mobile*. *Netbeans IDE* merupakan produk yang digunakan untuk melakukan pemrograman baik menulis kode, mengkompilasi, mencari kesalahan, dan mendistribusikan program. *Netbeans Platform* adalah sebuah modul yang merupakan karangan awal/fondasi dalam membangun aplikasi *desktop* yang besar. (Wahana Komputer, 2010) Berikut *Java / Netbeans IDE* yang ditunjukkan pada Gambar 2.8;



Gambar 2.8 Ruang Lingkup Java / Netbeans IDE

Langkah-langkah dalam membuat projek di netbeans akan dijelaskan seperti berikut:

1. Langkah pertama buka Netbeans IDE, kemudian Klik File  *New Project*. Lalu pada *Categori* pilih Java, dan pada kolom *Projects* pilih *Java Application*



Gambar 2.9 Tampilan Pilihan *New Project*

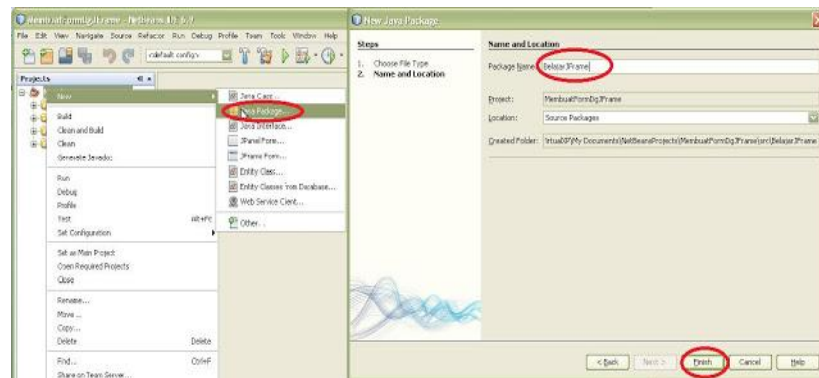
Keterangan :

Project Name : isikan nama project.

Project Location : merupakan tempat dimana yang akan meletakkan *project*, ubah atau biarkan secara default.

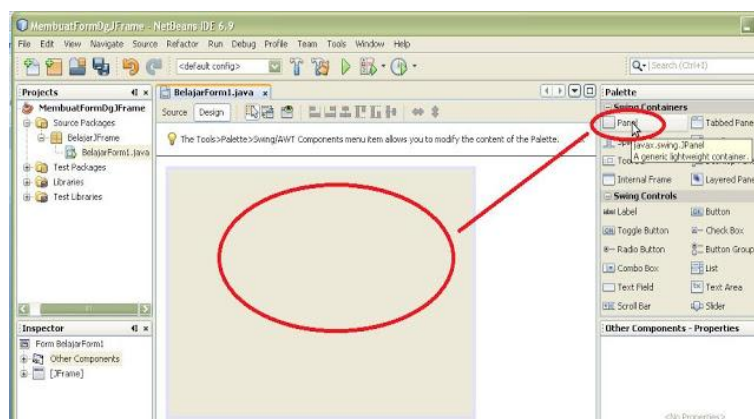
2. Selanjutnya buat package(folder) di dalam project yang sudah dibuat, hal ini di maksudkan agar penyimpanan file-file .java dapat dikelompokkan berdasarkan folder yang dikehendaki. Untuk membuat *package*, langkahnya sebagai berikut:

Klik kanan project yang sudah dibuat ,, *New,, Java Package*. Kemudian berikan nama *package(folder)* sesuai keinginan user.



Gambar 2.10 Pembuatan *Package*

3. Selanjutnya adalah membuat JFrame Form, dengan langkah – langkah sebagai berikut : Klik kanan package yang sudah dibuat klik *New* *JFrame Form*. Kemudian berikan nama pada kolom Class Name, misal : BelajarForm1. Form yang sudah dibuat tidak akan berfungsi apa – apa jika tidak terdapat komponen di dalamnya, untuk memasukkan komponen ke dalam form dapat dilakukan dengan cara : Klik komponen yang ingin di masukkan, kemudian klik ke atasnya form, contoh jika ingin menambahkan panel kedalam form, maka dapat di lakukan dengan cara : Klik *Panel* :



Gambar 2.11 Pembuatan *Panel*

Geserlah panel untuk memperoleh panjang dan lebar sesuai kebutuhan, agar panel tidak hilang oleh kasat mata dan bisa diperhatikan di jendela inspector : perhatikan gambar di atas, ketika kehilangan pandangan tentang posisi panel yang sudah diletakkan di atas form bisa klik komponen tersebut pada jendela inspector. Karena komponen JPanel (panel) di design secara default tidak kasat mata, agar selalu nampak tambahkanlah border, dengan cara klik kanan panel (yang berada di atas form) yang akan di munculkan pada properties. Pilihlah border, maka akan muncul pilihan berbagai macam border, pilihlah salah satu yang menurut cocok dengan aplikasi.

2.5.3 Ireports

IReports adalah sebuah *tool* yang digunakan untuk membuat design laporan pada *JasperReports*. (Ir.Yuniar Supardi, 2013). Artinya, dengan fasilitas ini *report* dapat dibuat dengan *drag and drop* bahkan secara otomatis *wizard*. Berikut tampilan gambar pada *IReport* yang ada di gambar 2.12:



Gambar 2.12 Tampilan *Jendela Ireport*

2.6 Struktur Organisasi SMA Perintis I Bandar Lampung



Gambar 2.13 Struktur Organisasi SMA Perintis I Bandar Lampung