



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Asropudin (2013:19), *Computer* merupakan alat bantu pemrosesan data secara elektronik dan cara pemrosesan datanya berdasarkan urutan instruksi atau program yang tersimpan dalam memori masing-masing komputer.

Blissmer dalam Jogiyanto (2005:1), Komputer adalah suatu alat elektronik yang mampu melakukan beberapa tugas berikut: menerima *input*, memproses *input* tadi sesuai dengan programnya, menyimpan perintah-perintah dan hasil dari pengolahan, menyediakan *output* dalam bentuk informasi.

Sanders dalam Jogiyanto (2005:1), Komputer adalah sistem elektronik untuk memanipulasi data yang cepat dan tepat serta dirancang dan diorganisasikan supaya secara otomatis menerima dan menyimpan data *input*, memprosesnya, dan menghasilkan *output* dibawah pengawasan suatu langkah-langkah instruksi-instruksi program yang tersimpan di memori (*stored program*).

Dari beberapa definisi komputer diatas penulis menyimpulkan bahwa komputer adalah suatu alat elektronik yang mampu melakukan beberapa tugas berikut: menerima *input*, memproses *input* tadi sesuai dengan programnya, menyimpan perintah-perintah dan hasil dari pengolahan, menyediakan *output* dalam bentuk informasi yang bermanfaat dan dapat bekerja secara otomatis.

2.1.2 Pengertian Perangkat Lunak(*Software*)

Sukamto dan Shalahuddin (2014:2), *Software* adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*).

Menurut Sujatmiko (2012:210), Perangkat lunak (*software*) istilah umum untuk data yang diformat dan disimpan secara digital, termasuk program



komputer, dokumentasinya, dan berbagai informasi yang bisa dibaca dan ditulis oleh komputer. Dengan kata lain bagian sistem komputer yang tidak berwujud.

Sutanta (2005:20), Perangkat lunak merupakan serangkaian intruksi dengan aturan tertentu yang mengatur operasi perangkat keras.

Dari beberapa definisi diatas penulis menyimpulkan bahwa perangkat lunak (*software*) adalah serangkaian instruksi elektronik yang dapat mengoperasikan suatu pekerjaan sesuai dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*).

2.1.3 Pengertian Program

Menurut Sujatmiko (2012:223), “Program adalah serangkaian petunjuk berupa perintah-perintah yang disusun sedemikian rupa melaksanakan suatu tugas yang akan di kerjakan oleh komputer.

Menurut Sutanta (2005:21), “Program merupakan bahasa yang mudah dipahami oleh orang awam.”

Dari beberapa definisi diatas penulis menyimpulkan bahwa program adalah serangkaian petunjuk berupa perintah atau instruksi yang dioperasikan untuk melaksanakan suatu tugas di dalam komputer dan mudah dipahami oleh orang awam.

2.1.4 Pengertian Data

Hartono (2013:15), “Data adalah hasil pengukuran dan pencatatan terhadap fakta tentang sesuatu, keadaan, tindakan, atau kejadian.”

Asropudin (2013:22) menjelaskan, “Data adalah kumpulan dari angka-angka maupun karakter-karakter yang tidak memiliki arti.”

Sujatmiko (2012:76), Data adalah kumpulan dari angka-angka maupun karakter-karakter yang tidak memiliki arti. Data dapat diolah sehingga menghasilkan informasi.



Dari beberapa definisi diatas dapat disimpulkan bahwa data adalah sekumpulan fakta yang harus diolah lebih lanjut untuk menghasilkan suatu informasi.

2.1.5 Pengertian *Database*

Wahana Komputer (2010:24), *Database* atau basis data adalah sekumpulan data yang memiliki hubungan secara logika dan diatur dengan susunan tertentu serta disimpan dalam media penyimpanan komputer.

Sukanto dan Shalahuddin (2014:43), “Basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan.”

Sujatmiko (2012:76), *Database* (Basis data) adalah representasi kumpulan fakta yang saling berhubungan disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redundansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan.

Wahana Komputer (2010:24), Dalam *database* ada sebutan-sebutan untuk satuan data yaitu:

1. Karakter

Karakter adalah satuan data terkecil. Data terdiri atas susunan karakter yang pada akhirnya mewakili data yang memiliki arti dari sebuah kata.

2. *Field*

Field adalah kumpulan dari karakter yang mewakili fakta tertentu misalnya seperti nama mahasiswa, tanggal lahir, dan lain-lain. Dalam dunia perancangan *database*, *field* juga disebut atribut. Bila dipandang dari sudut pemrograman berorientasi obyek maka sebuah *field* akan memiliki dua properti utama yaitu properti *name* dan properti *type*. Properti *name* atau nama adalah properti dari *field* yang berisi nama *field* yang mewakili data sejenis yang disimpannya. Sedangkan properti *type* adalah properti yang mengatur tipe data dari data yang akan ditampungnya. Misalnya nama *field*-nya adalah nama mahasiswa maka tipe datanya adalah *char*, bila nama *field*-nya adalah tanggal lahir maka tipe datanya adalah *date*.



3. *Record*

Record adalah kumpulan dari *field*. Pada *record* dapat menemukan banyak sekali informasi penting dengan cara mengkombinasikan *field-field* yang ada.

4. Tabel

Tabel adalah sekumpulan dari *record-record* yang memiliki kesamaan *entity* dalam dunia nyata. Kumpulan dari table adalah *database*, wujud fisik sebuah *database* dalam komputer adalah sebuah *file* yang di dalamnya terdapat berbagai tingkatan data.

2.1.6 Jaringan Komputer

Sofana (2013:3), Jaringan komputer atau *computer networks* adalah suatu himpunan interkoneksi sejumlah komputer *autonomous*. Dalam bahasa yang populer dapat dijelaskan bahwa jaringan komputer adalah kumpulan beberapa komputer (dan perangkat lain seperti *router*, *switch*, dan sebagainya) yang saling terhubung satu sama lain melalui media perantara. Media perantara ini bisa berupa media kabel ataupun media tanpa kabel (nirkabel). Informasi juga berupa data akan mengalir dari satu komputer ke perangkat yang lain, sehingga masing-masing komputer yang terhubung tersebut bisa saling bertukar data atau berbagi perangkat.

Sujatmiko (2012:145), Jaringan komputer adalah sekumpulan komputer individu (*personal computer*) yang dihubungkan dengan menggunakan *protocol transmission control protocol/ internet protocol* (TCP/IP).

Sofana (2013:7), Berdasarkan pola operasi atau fungsinya, jaringan komputer dapat dibagi menjadi 2 jenis, yaitu:

1. *Client-Server*

Client-Server adalah jaringan komputer yang mengharuskan salah satu (atau lebih) komputer difungsikan sebagai *server* atau *central.Server* melayani komputer lain yang disebut *client*. Layanan yang diberikan bisa berupa akses *Web*, *e-mail*, *file*, atau yang lain. *Client server* banyak dijumpai pada jaringan



Internet. Namun LAN atau jaringan lain pun bisa mengimplementasikan *client-server*. Hal ini sangat bergantung pada kebutuhan masing-masing.

2. *Peer to Peer*

Peer to peer adalah jaringan komputer dimana setiap komputer bisa menjadi *server* sekaligus *client*. Jadi tidak ada komputer yang “lebih utama” dibandingkan komputer lain. Setiap komputer dapat menerima dan memberikan akses dari/ ke komputer lain. *Peer to peer* banyak diimplementasikan pada LAN. Walaupun dapat juga diimplementasikan pada MAN, WAN atau *internet*, namun hal ini kurang lazim. Salah satu alasannya adalah masalah manajemen dan *security* pada jaringan *peer to peer* manakala pengguna komputer sudah sangat banyak.

2.2 Teori Judul

2.2.1 Pengertian Aplikasi

Asropudin (2013:7), “Aplikasi dibuat untuk mengerjakan atau menyelesaikan masalah-masalah khusus.”

Sujatmiko (2013:23), *Application* adalah program komputer yang dibuat oleh suatu perusahaan komputer untuk membantu manusia dalam mengerjakan tugas-tugas tertentu, misalnya Ms. Word, Ms.Excel dan lain-lain.

Dari beberapa definisi diatas dapat disimpulkan bahwa aplikasi adalah perangkat lunak yang digunakan untuk mengerjakan masalah-masalah khusus.

2.2.2 Pengertian Rekam Medis

Direktorat Jenderal Bina Pelayanan Medik (2006:11), Rekam medis diartikan sebagai keterangan baik yang tertulis maupun yang terekam tentang identitas, anamnesa, pemeriksaan fisik, laboratorium, diagnosa serta segala pelayanan dan tindakan medis yang diberikan kepada pasien, dan pengobatan baik yang dirawat inap, rawat jalan maupun yang mendapatkan pelayanan gawat darurat.

Direktorat Jenderal Bina Pelayanan Medik (2006:11), Sesuai dengan penjelasan Pasal 46 ayat (1) UU No.29 Tahun 2004 tentang praktik kedokteran



disebutkan bahwa, yang dimaksud Rekam Medis adalah berkas yang berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan, tindakan dan pelayanan lain yang telah diberikan kepada pasien. Dan yang dimaksud dengan petugas adalah dokter atau dokter gigi atau tenaga kesehatan lain yang memberikan pelayanan kesehatan secara langsung kepada pasien.

Dari beberapa definisi diatas dapat disimpulkan bahwa rekam medis adalah berkas yang berisikan catatan dan dokumen identitas, anamnesa, pemeriksaan fisik, laboratorium, diagnosa serta segala pelayanan dan tindakan medis yang diberikan kepada pasien, dan pengobatan baik yang dirawat inap, rawat jalan maupun yang mendapatkan pelayanan gawat darurat.

Direktorat Jenderal Bina Pelayanan Medik (2006:13), Tujuan rekam medis adalah menunjang tercapainya tertib administrasi dalam rangka upaya peningkatan pelayanan kesehatan di rumah sakit. Tanpa didukung suatu sistem pengelolaan rekam medis yang baik dan benar, tidak akan tercipta tertib administrasi rumah sakit sebagaimana yang diharapkan. Sedangkan tertib administrasi merupakan salah satu faktor yang menentukan di dalam upaya pelayanan kesehatan di rumah sakit.

2.2.3 Pengertian Pasien Rawat Jalan

Rahayu (2014:403), Pasien rawat jalan adalah pasien yang hanya memperoleh pelayanan kesehatan tertentu dan tidak melakukan rawat inap atau opname.

2.2.4 Pengertian Pasien Rawat Inap

Rahayu (2014:403), Pasien rawat inap adalah pasien yang memperoleh pelayanan tinggal atau dirawat inap pada suatu unit kesehatan tertentu; pasien yang dirawat inap dirumah sakit.

2.2.5 Pengertian Rumah Sakit Ibu dan Anak



Rahayu (2014:475), Rumah Sakit Ibu dan Anak adalah rumah sakit yang khusus melayani pemeriksaan ibu-ibu hamil, ibu-ibu yang akan melahirkan, dan menangani kesehatan balita.

2.2.6 Pengertian Rumah Sakit Ibu dan Anak Tiara Fatrin Palembang

Rumah Sakit Ibu dan Anak Tiara Fatrin Palembang merupakan salah satu rumah sakit swasta yang bergerak dibidang jasa pelayanan kebidanan dan anak-anak.

2.2.7 Pengertian *Client-Server*

Sufana (2013:3), *Client-Server* adalah jaringan komputer yang mengharuskan salah satu (atau lebih) komputer difungsikan sebagai *server* atau *central*. *Server* melayani komputer lain yang disebut *client*. Layanan yang diberikan bisa berupa akses *Web*, *e-mail*, *file*, atau yang lain. *Client server* banyak dijumpai pada jaringan *Internet*. Namun LAN atau jaringan lain pun bisa mengimplementasikan *client-server*. Hal ini sangat bergantung pada kebutuhan masing-masing.

Sujatmiko (2012:66), *Client-server* adalah sebuah paradigma dalam teknologi informasi yang merujuk kepada cara untuk mendistribusikan aplikasi ke dalam dua pihak, yaitu pihak klien dan pihak *server*. Dalam model *client/ server*, sebuah aplikasi dibagi menjadi dua bagian yang terpisah, tapi masih merupakan sebuah kesatuan yakni komponen klien dan komponen *server*. Komponen klien juga sering disebut sebagai *front-end*, sementara komponen *server* disebut sebagai *back-end*.

Dari beberapa definisi diatas dapat disimpulkan bahwa *client-server* adalah salah satu jaringan komputer yang mengharuskan salah satu komputer berfungsi sebagai *server* atau *central* yang melayani komputer lain yang disebut *client*. Layanan yang diberikan bisa berupa akses *Web*, *e-mail*, *file*, atau yang lain.



2.2.8 Pengertian Aplikasi Rekam Medis Pasien Rawat Jalan dan Rawat Inap pada Rumah Sakit Ibu dan Anak Tiara Fatrin Palembang berarsitektur *client-server*

Merupakan suatu perangkat lunak yang dikembangkan untuk difungsikan secara khusus dalam mengelolah data pasien baik itu pasien rawat inap maupun rawat jalan serta mampu membuat laporan bulanan jumlah pasien yang melakukan rawat inap dan rawat jalan di Rumah Sakit Ibu dan Anak Tiara Fatrin Palembang, sehingga pengolahan data pasien dapat dilakukan secara terkomputerisasi dan terhubung komputer satu dengan komputer lain dengan menggunakan jaringan *client-server*.

2.3 Teori Khusus

2.3.1 Pemograman Berorientasi Objek

2.3.1.1 Pengertian Pemograman Berorientasi Objek

Menurut Sukamto dan Shalahuddin (2014:100), “Berorientasi Objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya.”

2.3.1.2 Konsep Dasar Berorientasi Objek

Beberapa konsep dasar yang harus dipahami tentang metodologi berorientasi objek.

1. Kelas (*Class*)

Kelas adalah kumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statik dan himpunan objek yang sama yang mungkin lahir atau diciptakan dan kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (metode), hubungan (*relationship*).

2. Objek (*Object*)

Objek adalah abstraksi dan sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, struktur, status atau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu



menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan.

3. Metode (*Method*)

Operasi atau metode atau *method* pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek.

4. Atribut (*Attribute*)

Atribut dari sebuah kelas adalah *variable* global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dipunyai secara individual oleh sebuah objek, misalnya berat, jenis dan sebagainya.

5. Abstraksi (*Abstraction*)

Prinsip untuk mempresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. Perwarisan (*Inheritance*)

Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dari dirinya.

7. Antarmuka (*Interface*)

Antarmuka atau *interface* sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa ini. Deklarasi metode pada sebuah *interface* dapat diimplementasikan oleh kelas lain,.

8. *Reusability*

Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut. Misalkan dalam sebuah aplikasi peminjaman buku diperlukan kelas anggota,



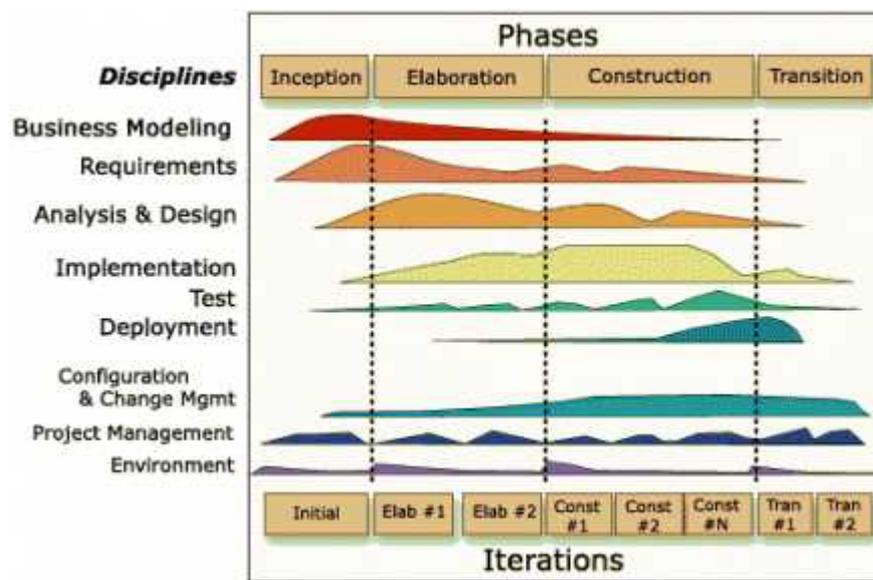
maka ketika membuat aplikasi penyewaan VCD, kelas anggota ini bisa digunakan kembali dengan sedikit perubahan untuk aplikasi penyewaan VCD tanpa harus membuat dari awal kembali.

9. Generalisasi dan Spesialisasi

Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus. Misalnya kelas yang lebih umum (generalisasi) adalah kendaraan darat dan kelas khususnya (spesialisasi) adalah mobil, motor, dan kereta.

2.3.2 Metodologi RUP (*Rational Unified Process*)

Sukanto dan Shalahudin (2014:125), RUP (*Rational Unified Process*) adalah pengembangan perangkat lunak yang dilakukan berulang-ulang, fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*). RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang baik (*well defined*) dan penstrukturan yang baik (*well structured*). RUP menyediakan pendefinisian struktur yang baik untuk alur hidup proyek perangkat lunak.



Gambar 2.1. Arsitektur Rational Unified Process



2.3.2.1 Penerapan Tahap Metodologi Pengembangan Perangkat Lunak dengan RUP

Dalam *Rational Unified Process* terdapat empat tahap pengembangan perangkat lunak yaitu:

1. *Inception*

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan dan mendefinisikan kebutuhan akan sistem yang akan dibuat. Tahap yang dibutuhkan pada tahap ini antara lain memahami ruang lingkup dari proyek (termasuk biaya, waktu, kebutuhan, resiko dan lain sebagainya).

Hasil yang diharapkan pada tahap ini adalah memenuhi *Lifecycle Objective Milestone* dengan kriteria sebagai berikut:

- a. Umpan balik dari pendefinisian ruang lingkup, perkiraan biaya, perkiraan jadwal.
- b. Kebutuhan dimengerti dengan pasti dan sejalan dengan kasus primer yang dibutuhkan.
- c. Kredibilitas dari perkiraan biaya, perkiraan jadwal, penentuan skala prioritas, resiko, dan proses pengembangan.
- d. Ruang lingkup purwarupa (*prototype*) yang akan dikembangkan.
- e. Membangun garis dasar dengan membandingkan perencanaan actual dengan perencanaan yang direncanakan.

2. *Elaboration*

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada sistem *prototype*.

Hasil yang diharapkan pada tahap ini adalah memenuhi *Lifecycle Objective Milestone* dengan kriteria sebagai berikut:

- a. Model kasus yang digunakan (*use case*) dimana kasus dan aktor yang terlibat telah diidentifikasi dan sebagian besar kasus harus



dikembangkan. Model *use case* harus dikembangkan. Model *use case* harus 80 persen lengkap dibuat.

- b. Deskripsi dari arsitektur perangkat lunak dari proses pengembangan sistem perangkat lunak telah dibuat.
- c. Rancangan arsitektur yang dapat diimplementasikan dan mengimplementasikan *use case*.
- d. Kasus bisnis atau proses bisnis dan daftar risiko yang sudah mengalami perbaikan telah dibuat.
- e. Rencana pengembangan untuk seluruh proyek telah dibuat.
- f. *Prototype* yang dapat didemonstrasikan untuk mengurangi setiap resiko teknis yang diidentifikasi.

3. *Construction*

Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program.

4. *Transition*

Tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh *user*. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *initial operational capability milestone* atau batas/tonggak kemampuan operasional awal. Aktivitas pada tahap ini termasuk pada pelatihan *user*, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan *user*.

2.3.3 *Unified Modelling Language (UML)*

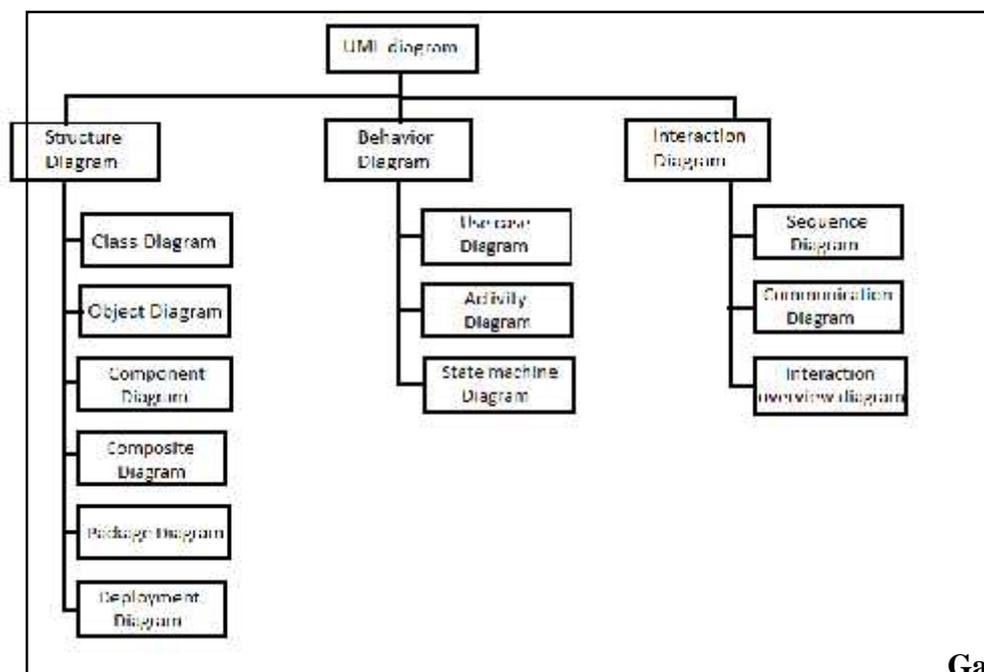
Menurut Sukamto dan Shalahuddin (2014:133), “*UML (Undefined Modelling Language)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.”



Menurut Widodo dan Herlawati (2011:6), “UML singkatan dari *Unified Modeling Language* yang berarti bahasa permodelan standar.”

2.3.3.1 Macam-macam Diagram UML

Sukamto dan Shalahuddin (2014:140), Pada UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini:



Gamb

ar 2.2 Diagram UML

Sumber: Sukamto dan Shalahudin (2014:140)

Berikut ini penjelasan singkat dari pembagian kategori tersebut.

a. *Structure Diagram*

Yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

b. *Behavior Diagram*

Yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

c. *Interaction Diagram*



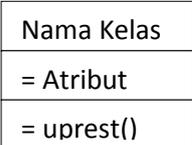
Yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

2.3.4 *Class Diagram* (Diagram Kelas)

Sukamto dan Shalahudin (2014:141), *Class Diagram* atau diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Tabel 2.1 Simbol-simbol Diagram *Class*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem.
Antarmuka/ <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
Asosiasi/ <i>Association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah/ <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus).



Kebergantungan/ <i>dependency</i>	Relasi antarkelas dengan makna kebergantungan antarkelas.
Agregasi/ <i>aggregation</i> 	Relasi antarkelas dengan makna semua bagian (<i>whole-part</i>).

Sumber: Sukamto dan Shalahudin (2014:146)

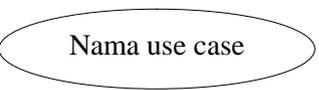
2.3.5 Use Case Diagram

Sukamto dan Shalahudin (2014:155), *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara kasar, *user case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

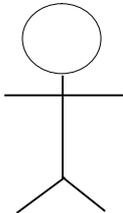
Widodo dan Herlawati (2011:10), Diagram *use case* merupakan digram yang bersifat statis. Diagram ini memperlihatkan himpunan *use case* dan aktor-aktor. Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.

Berikut adalah simbol-simbol yang ada pada diagram *use case*:

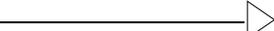
Tabel 2.2 Diagram *use case*

Simbol	Deskripsi
<i>Use Case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar peran antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i> .



<p>Aktor/<i>Actor</i></p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan kata benda di awal frase nama aktor.</p>
<p>Asosiasi/<i>association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>

Lanjutan Tabel 2.2 Diagram *use case*

Simbol	Deskripsi
<p>Ekstensi/<i>extend</i></p> <p><<extend>></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.</p>
<p>Generalisasi/<i>generalisation</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>



<p>Menggunakan/ <i>include/ uses</i></p> <p style="text-align: center;"><<extend>></p> <p style="text-align: center;">—————→</p> <p style="text-align: center;"><<uses>></p> <p style="text-align: center;">—————▷</p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p>
--	---

Sumber: Sukamto dan Shalahudin (2014:156)

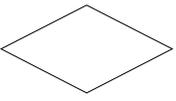
2.3.6 Activity Diagram (Diagram Aktivitas)

Sukamto dan Shalahudin (2014:161), *Activity diagram* atau diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

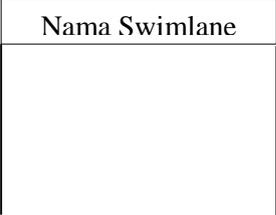
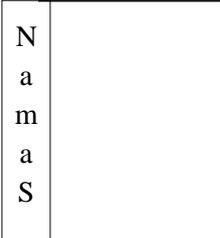
Widodo dan Herlawati (2011:11), Diagram Aktivitas (*Activity Diagram*) adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem.

Berikut simbol-simbol yang sering digunakan dalam *activity diagram*:

Tabel 2.3 Simbol-simbol *Activity Diagram*

No.	Simbol	Keterangan
1.	Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3.	Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.



4.	Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5.	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
6.	Swimlane  atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber: Sukamto dan Shalahudin (2014:162)

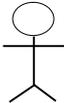
2.3.7 Sequence Diagram (Diagram Sekuensial)

Sukamto dan Shalahudin (2014:165), *Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.

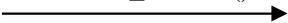
Tabel 2.4 Simbol-simbol Sequence Diagram

No.	Simbol	Keterangan
-----	--------	------------

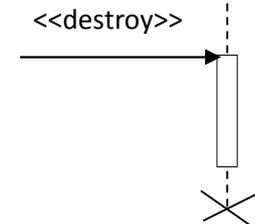


1.	<p>Aktor</p>  <p>Nama aktor Atau</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">Nama aktor</div> <p>Tanpa waktu aktif</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2.	<p>Garis hidup / <i>lifeline</i></p> 	Menyatakan kehidupan suatu objek.
3.	<p>Objek</p> 	Menyatakan objek yang berinteraksi dengan pesan.
4.	<p>Waktu aktif</p> 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
5.	<p>Pesan tipe <i>create</i></p> <p><<create>></p> 	Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat.

Lanjutan Tabel 2.4 Simbol-simbol *Sequence Diagram*

No.	Simbol	Keterangan
6.	<p>Pesan tipe <i>call</i></p> <p>1: nama_metode()</p> 	Menyatakan suatu objek memanggil operasi/ metode yang ada pada objek lain atau dirinya sendiri.



7.	Pesan tipe <i>send</i> 1: masukan 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek dikirim.
8.	Pesan tipe <i>return</i> 1: keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9.	Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

Sumber: Sukamto dan Shalahudin (2014:165)

2.3.8. Kamus data

Sukamto dan Shalahuddin (2014:73), Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan).

2.4 Teori Program

2.4.1 Pemrograman *Visual Basic* 2012

2.4.1.1 Pengertian Pemrograman *Visual Basic* 2012



Wahana Komputer (2013:2), *Visual Basic* merupakan bahasa pemrograman yang paling populer dalam dunia komputer. Beraneka ragam program dapat dibuat dengan aplikasi ini.

2.4.1.2 Visual Studio Toolbox

Wahana Komputer (2013:3), *Toolbox* dapat diartikan sebagai kotak alat. Didalam *toolbox* terdapat banyak ikon di mana tiap ikon tersebut melambangkan objek yang disebut kontrol. Kontrol pada *toolbox* bisa digunakan untuk membuat *instance* atau kejadian baru di *form* yang digunakan selama pemrograman.



Gambar 2.3 *Toolbox All Windows Form*

Sumber: Wahana Komputer (2013:4)

Penjelasan fungsi-fungsi dari kontrol *toolbox* yang ada di *Visual Basic* 2012 seperti berikut:

Tabel 2.6 Kontrol *Toolbox Visual Basic* 2012

Nama	Keterangan
<i>Pointer</i>	Untuk memilih <i>pointermouse</i> atau melepaskan dari memilih objek-objek <i>toolbox</i> .

Lanjutan Tabel 2.6 Kontrol *Toolbox Visual Basic* 2012

Nama	Keterangan
------	------------



<i>Background Worker</i>	Mengeksekusi operator tertentu di <i>thread</i> lain.
<i>Binding Navigator</i>	Menyediakan antarmuka untuk navigasi dan manipulasi data yang dikaitkan dengan <i>form</i> .
<i>Binding Source</i>	Mengkapsulasikan sumber data untuk sebuah <i>form</i> dan menyediakan fasilitas <i>update</i> , navigasi, <i>filter</i> , dan <i>sorting</i> .
<i>Button</i>	Untuk menerima klik dari pengguna. Jika pengguna mengkliknya maka kode tertentu bisa dieksekusi.
<i>Check Box</i>	Untuk menerima tanda centang dari pengguna. Jika pengguna mencentangnya maka item di kotak tersebut otomatis bisa dipakai untuk program.
<i>Checked List Box</i>	Menampilkan item-item seperti <i>listbox</i> biasa, namun dengan <i>checkbox</i> disebelah kirinya sehingga satu <i>listbox</i> bisa menerima dua item atau lebih.
<i>Combo Box</i>	Menampilkan item-item namun dalam bentuk hanya satu yang terlihat. Jika pengguna mengklik <i>combobox</i> barulah semua item didalamnya akan terlihat.
<i>Context Menu Strip</i>	Menampilkan menu <i>shortcut</i> jika pengguna mengklik kanan pada item.
<i>Data Grid View</i>	Menampilkan data dalam bentuk table yang bisa dimodifikasi.
<i>Data Set</i>	Merupakan representasi data dari sumber data di memori.

Lanjutan Tabel 2.6 Kontrol *Toolbox Visual Basic 2012*

Nama	Keterangan
------	------------



<i>Date Time Picker</i>	Memungkinkan pengguna menampilkan tanggal dan memilihnya. Selanjutnya tanggal dan waktu dipilih dapat dimodifikasi menggunakan format tertentu.
<i>Directory Entry</i>	Mengenkapsulasi node atau objek di hirarki <i>Active Directory</i> (AD) di <i>Windows Server</i> .
<i>Directory Searcher</i>	Melakukan <i>query</i> terhadap <i>active directory</i> .
<i>Domain Up Down</i>	Menampilkan nilai <i>string</i> yang ditampilkan pengguna dari daftar item dengan cara mengklik tombol up dan down.
<i>Error Provider</i>	Menentukan dukungan untuk inisialisasi transaksi.
<i>Event Log</i>	Menyediakan interaksi dengan <i>eventlog</i> dari windows.
<i>File System Watcher</i>	Memonitor perubahan sistem file dan mengeksekusi event ketika ada perubahan direktori atau file.
<i>Group Box</i>	Mengelompokkan control dalam satu kotak untuk mempermudah pengaturan dan manajemen antarmuka.
<i>Image List</i>	Menampilkan daftar gambar.
<i>Label</i>	Untuk menampilkan teks dimana nantinya isi didalamnya bisa diberi

Lanjutan Tabel 2.6 Kontrol *Toolbox Visual Basic 2012*

Nama	Keterangan
------	------------

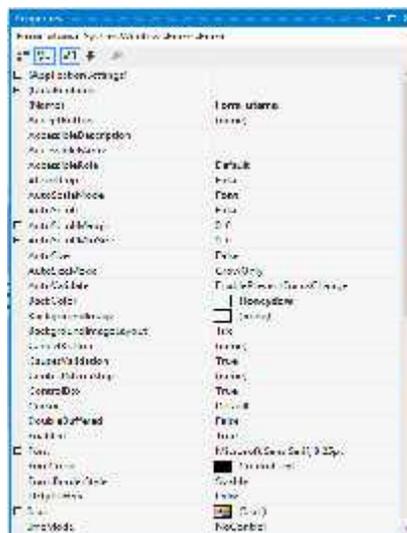


	masukan oleh pengguna seperti halnya <i>textbox</i> .
<i>List Box</i>	Menampilkan item-item dalam bentuk kotak di mana semua item ditampilkan di dalam daftar.
<i>List View</i>	Menampilkan item-item dalam satu dari lima tampilan yang berbeda.
<i>Picture Box</i>	Menampilkan gambar yang tampilannya bisa diatur sedemikian rupa.
<i>Radio Button</i>	Memungkinkan pengguna untuk memilih satu opsi dari beberapa pilihan yang dikaitkan dengan tombol radio lainnya.
<i>Text Box</i>	Memungkinkan pengguna memasukan teks yang bisa lebih dari satu baris dan menyediakan kemampuan pengeditan dan <i>masking</i> kata sandi.

Sumber: Wahana Komputer (2013:4)

2.4.1.3 Panel *Properties*

Panel *properties* menampilkan *properties* dari kontrol yang dipilih. Berikut ini merupakan contoh tampilan jendela *properties* untuk kontrol *form* dan arti-arti dari nilai *properties*-nya:





Gambar 2.4.Panel *properties*

Sumber: Wahana Komputer (2013: 16)

Tabel 2.7 Kontrol *Properties Visual Basic 2012*

Nama	Keterangan
<i>Auto Scroll</i>	Mengindikasikan apakah <i>scrollbar</i> otomatis diaktifkan jika isi kontrol lebih besar dari area yang terlihat.
<i>Auto Size</i>	Menentukan apakah kontrol akan mengatur ulang ukuran dirinya sendiri secara otomatis sesuai dengan kontennya.
<i>Back Color</i>	Menentukan warna <i>background</i> dari komponen.
<i>Background Image</i>	Menentukan gambar <i>background</i> untuk kontrol.
<i>Control Box</i>	Menentukan apakah <i>form</i> memiliki kontrol/ sistem <i>menubox</i> atau tidak.
<i>Enabled</i>	Menentukan apakah kontrol dalam posisi aktif atau tidak.

Lanjutan Tabel 2.7 Kontrol *Properties Visual Basic 2012*

Nama	Keterangan
<i>Font</i>	Menentukan jenis dan ukuran <i>font</i> yang akan ditampilkan di kontrol.
<i>Fore Color</i>	Menentukan warna <i>foreground</i> / depan dari komponen yang digunakan untuk menampilkan teks dikomponen tersebut.
<i>Icon</i>	Mengindikasikan <i>icon</i> untuk sebuah <i>form</i> . <i>Icon</i> ini akan ditampilkan



	dibagian <i>menu box</i> sistem dan ketika <i>form</i> sedang di <i>minimize</i> .
<i>Locked</i>	Menentukan apakah kontrol bisa dipindahkan atau di <i>resize</i> .
<i>Maximize Box</i>	Menentukan apakah <i>form</i> memiliki tombol <i>maximize</i> dibagian kanan atas di <i>captionbar</i> .
<i>Maximum Size</i>	Menentukan ukuran maksimal <i>form</i> yang bisa di <i>resize</i> .
<i>Minimize Box</i>	Menentukan apakah <i>form</i> memiliki kotak <i>minimize</i> dibagian kanan atas dari <i>captionbar</i> .
<i>Minimum Size</i>	Menentukan ukuran minimal dari <i>form</i> yang bisa di <i>resize</i> .
<i>Opacity</i>	Menentukan persen ketidaktransparan dari kontrol, 0% artinya transparan dan 100% tidak transparan.

Lanjutan Tabel 2.7 Kontrol *Properties Visual Basic 2012*

Nama	Keterangan
<i>Show Icon</i>	Menentukan apakah <i>icon</i> ditampilkan di <i>titlebar</i> di <i>form</i> .
<i>Size</i>	Menentukan ukuran kontrol dalam satuan piksel.
<i>Text</i>	Menentukan teks yang diasosiasikan untuk kontrol tersebut.
<i>Window State</i>	Menentukan kondisi awal jendela dari <i>form</i> .



Sumber: Wahana Komputer (2013:17)

2.4.2 *SQL Server 2008*

2.4.2.1 Pengertian *SQL Server 2008*

Wahana Komputer (2010:2), *SQL Server 2008* adalah sebuah terobosan baru di Microsoft dalam bidang *database*. *SQL Server* adalah sebuah DBMS (*Database Management System*) yang dibuat oleh Microsoft untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan Oracle. *SQL Server 2008* dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa *SQL Server 2008* membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data.

2.4.2.2 Kebutuhan *Hardware SQL Server 2008*

Wahana Komputer (2010:2), Sebelum melakukan proses *download* dan instalasi *SQL Server 2008*, pahami terlebih dahulu *hardware* yang harus digunakan untuk menjalankan *SQL Server*. Adapun *hardware* yang diperlukan minimal adalah sebagai berikut:

- a. *Processor* > = 1 GHz.
- b. *Memory* > = 512 Mb.
- c. OS = Windows XP 32 bit, Windows 7.

Biarpun dapat diinstal pada sistem komputer dengan memori 512 MB, tetapi disarankan menggunakan memori 1 GB. Sedangkan untuk jaringannya diperlukan adalah:

- a. *Shared Memory*
- b. TCP/ IP
- c. *Named Pipes*
- d. VIA (*Virtual Interface Adapter*)

Sedangkan bila menggunakan Windows XP Profesional maka pastikan bahwa pada sistem anda telah terdapat .Net Framework 3.5 SP 1 dan juga *update*-nya, kemudian Windows Installer 4.5 dan Windows Power Shell.



2.4.2.3 Kunci dalam *SQL Server 2008*

Wahana Komputer (2013:220), Kunci primer merupakan atribut (satu kolom) atau kombinasi kolom yang nilainya unik (tidak ganda) dimana dengan kunci tersebut dapat memberi identitas pada sebuah *record* dalam satu entitas.

Ketika memilih kunci untuk entitas, sebuah kolom haruslah memiliki beberapa karakteristik:

1. Tiap *record* di entitas tersebut tidak boleh memilih nilai kosong (*null*) atau dengan kata lain nilainya harus ada (*not null*).
2. Nilainya harus unik dan tidak ada yang ganda.
3. Nilainya tidak berubah atau menjadi kosong ketika diperlukan.
4. Hanya ada satu kunci primer yang didefinisikan untuk tiap entitas.

Kadang diperlukan kombinasi lebih dari satu kolom sebagai identitas unik dari table. Untuk menerangkan istilah kunci primer yang terdiri atas lebih dari satu *field* ini disebut kunci komposit (*composite key*).

Foreignkey atau kunci asing merupakan atribut yang akan melengkapi relasi dengan cara mengidentifikasi entitas induknya. Kunci asing memiliki metode yang berguna untuk menjaga integritas data dan untuk melakukan navigasi antara beberapa table. Setiap relasi di table harus didukung oleh kunci asing.