

## **BAB II**

### **LANDASAN TEORI**

#### **II.1. Data**

##### **II.1.1 Pengertian Data**

Data adalah suatu hasil dari deskripsi tentang benda, kejadian, aktifitas dan transaksi yang tidak mempunyai makna atau tidak berpengaruh langsung kepada si pemakai. (Abdul Kadir : 2014 ; 44).

Data adalah fakta atau kenyataan yang menggambarkan suatu kejadian – kejadian yang mempunyai arti tersendiri. (Hamim Tohari : 2014 ; 7).

##### **II.1.2 Tipe Data**

Tipe data adalah suatu bentuk penggolongan jenis data berdasarkan kategori data, ukuran dan kegunaan data yang dapat ditampung oleh sebuah variabel dalam media bahasa pemrograman. (Muhammad Sadeli : 2010 ; 105).

Tipe data adalah jenis data yang digunakan dalam pembuatan program. Jenis data yang digunakan adalah Boolean, Byte, Char, Decimal, Double, Date, Integer, Long Single, String dan Short. (Hendrayudi : 2011 ; 14).

#### **II.2. Sistem**

Sistem adalah sekumpulan elemen – elemen yang saling berinteraksi untuk mencapai suatu tujuan tertentu. (Hamim Tohari : 2014 ; 2).

Sistem berasal dari Bahasa Latin (*Systema*) dan bahasa Yunani (*Sustema*) adalah suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi, di mana suatu model matematika seringkali bisa dibuat. (M. Sulaiman Silalahi ; 2013)

Sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Jika dalam sebuah sistem terdapat elemen yang tidak memberikan manfaat dalam mencapai suatu tujuan yang sama, maka elemen tersebut dapat dipastikan bukanlah bagian dari sistem. (Abdul Kadir : 2014 ; 61).

Sistem adalah suatu kesatuan usaha yang terdiri dari bagian – bagian yang saling berkaitan secara teratur dan berusaha untuk mencapai suatu tujuan di dalam suatu lingkungan yang kompleks. (Marimin : 2010 ; 1).

## **II.2.1 Karakteristik Sistem**

### **1. Komponen atau Elemen (*Components*)**

Suatu sistem yang terdiri dari komponen – komponen yang saling berinteraksi, yang artinya saling bekerja sama untuk membentuk suatu kesatuan.

### **2. Batas Sistem (*Boundary*)**

Batas sistem merupakan daerah yang membatasi antara sistem yang satu dengan sistem yang lainnya atau dengan lingkungan luarnya. Adanya batas sistem ini, maka sistem dapat membentuk suatu kesatuan, karena dengan batas

sistem ini, fungsi dan tugas dari subsistem satu dengan yang lainnya berbeda tetapi tetap saling berinteraksi. Batas sistem ini memungkinkan suatu sistem yang dipandang sebagai suatu kesatuan. Batas sistem merupakan ruang lingkup atau scope dari sistem atau subsistem itu sendiri.

### **3. Lingkungan Luar Sistem (*Environment*)**

Lingkungan luar dari sistem adalah segala sesuatu di luar batas sistem yang mempengaruhi operasi suatu sistem. Lingkungan luar sistem dapat bersifat menguntungkan atau merugikan. Lingkungan luar sistem yang bersifat menguntungkan harus dipelihara dan dijaga. Sedangkan lingkungan luar yang bersifat merugikan harus dihilangkan supaya tidak mengganggu operasi sistem.

### **4. Penghubung Sistem (*Interface*)**

Penghubung sistem merupakan suatu media (penghubung) antara satu subsistem anantara subsistem yang satu dengan yang lainnya yang membentuk satu kesatuan, sehingga sumber-sumber daya yang mengalir dari subsistem yang satu dengan subsistem yang lainnya .

### **5. Masukan (*Input*)**

Masukan adalah energy atau sesuatu yang dimasukkan ke dalam system yang dapat berupa masukan yaitu energi yang dimasukkan supaya system dapat beroperasi atau masukan sinyal yang merupakan energy yang diproses untuk menghasilkan suatu keluaran.

## **6. Keluaran (*Output*)**

Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna juga merupakan keluaran atau tujuan akhir dari sistem.

## **7. Pengolah (*Process*)**

Suatu sistem mempunyai bagian pengolah yang akan mengubah *input* menjadi *output*.

## **8. Sasaran (*Objective*)**

Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya. (Hamim Tohari : 2014 ; 2-3).

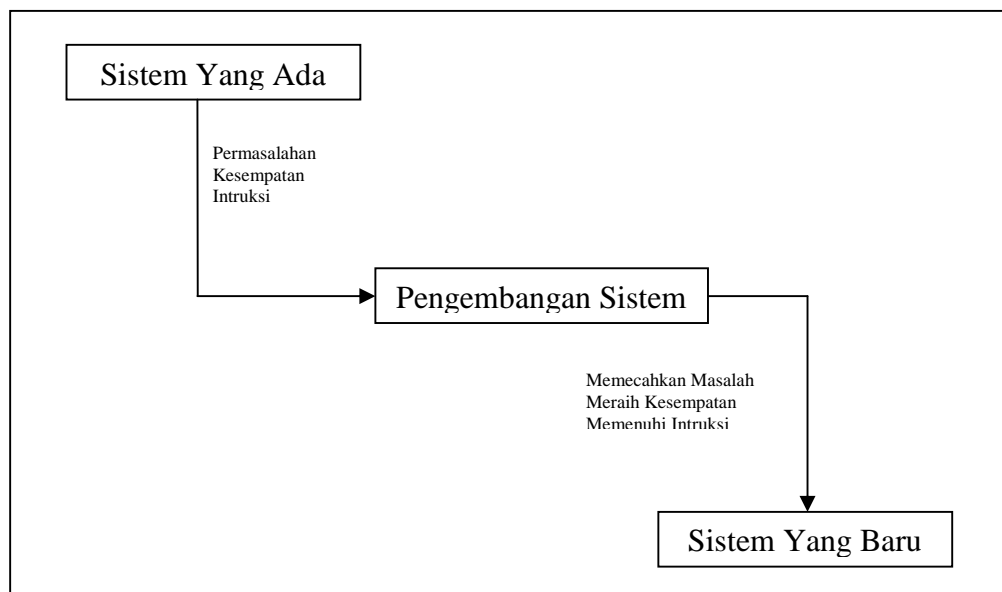
### **II.2.2 Pengembangan Sistem**

Pengembangan sistem (*System Development*) dapat berarti menyusun sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau pun memperbaiki sistem yang sudah ada.

Sistem yang lama perlu diperbaiki atau diganti disebabkan karena beberapa hal, antara lain ;

1. Adanya permasalahan – permasalahan (*Problems*) yang timbul di sistem yang lama. Permasalahan – Permasalahan yang timbul dapat berupa ;
  - a. Adanya ketidakberesan dalam sistem yang lama sehingga menyebabkan sistem tersebut tidak dapat beroperasi sesuai dengan yang diharapkan.

- b. Pertumbuhan organisasi yang menyebabkan harus disusunnya sistem yang baru. Pertumbuhan organisasi diantaranya adalah kebutuhan informasi yang semakin meningkat, perubahan prinsip akuntansi atau pun pengolahan data yang baru.
2. Untuk meraih kesempatan – kesempatan (*Opportunities*), Teknologi informasi telah berkembang dengan cepat. Perangkat keras komputer, perangkat lunak komputer dan teknologi komunikasi yang telah begitu cepat berkembang. Organisasi mulai merasakan bahwa teknologi informasi telah berkembang pesat sehingga dapat mendukung dalam proses pengambilan keputusan yang dilakukan oleh pihak manajemen.
  3. Adanya intruksi – intruksi (*Directives*), penyusunan sistem yang baru dapat juga terjadi karena adanya intruksi – intruksi dari atas pimpinan atau pun dari luar organisasi.



**Gambar II.1 Pengembangan Sistem (Informasi)**  
 (Sumber : Hamim Tohari : 2014 ; 5)

### II.2.3 Tahapan Pengembangan Sistem

Sistem yang baik adalah sistem yang selalu menyesuaikan dengan perubahan lingkungan disekitarnya. Sistem tersebut harus dinamis menuju pada keadaan yang lebih baik secara berkelanjutan.

Adapun tahapan – tahapan pengembangan sistem yang dinamis agar mendapatkan sistem yang baik, antara lain ;

#### 1. Perencanaan

Pada tahapan ini menyangkut studi kebutuhan – kebutuhan *user*, studi kelayakan baik secara teknis maupun teknologi serta penjadwalan pengembangan suatu proyek sistem informasi.

#### 2. Analisis

Pada tahapan ini kita berusaha mengenali segenap permasalahan yang muncul pada penggunaan, mengenali komponen – komponen sistem, dengan objek – objek , hubungan antar objek, dan sebagainya.

#### 3. Perancangan

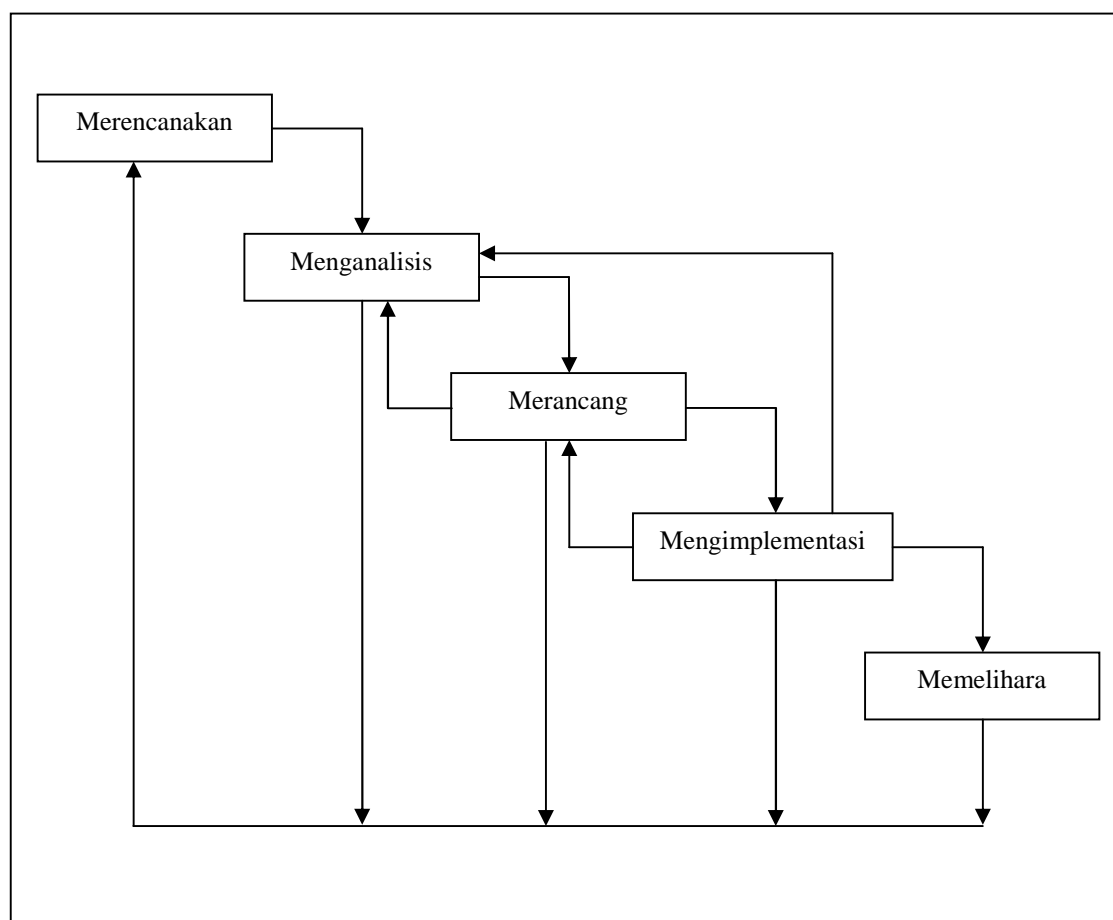
Pada tahapan ini kita mencoba mencari solusi permasalahan dari tahapan analisis.

#### 4. Implementasi

Pada tahapan ini mengimplementasikan perancangan sistem ke situasi yang nyata. Pada tahapan ini kita mulai dengan pemeliharaan perangkat keras, kemudian masuk kedalam tahapan penyusupan perangkat lunak dan aplikasi (*Pengkodean/Koding*) untuk mengukur tingkat kesesuaian sistem tersebut dengan *Pengguna* atau *User*.

## 5. Pemeliharaan atau Perawatan

Pada tahapan ini kita bisa mulai melakukan pengoprasian objek sistem, dan jika diperlukan dapat melakukan perbaikan – perbaikan kecil. Kemudian, jika waktu penggunaan sistem habis, maka kita akan masuk lagi pada tahap perencanaan.



**Gambar II.2. Kerangka Kerja Pengembangan Sistem (Informasi)**  
(Sumber : Hamim Tohari : 2014 ; 6)

## **II.2.4 Elemen Sistem**

Elemen-elemen yang membuat sebuah system yaitu:

### **1. Tujuan**

Setiap sistem memiliki tujuan (Goal), entah hanya satu atau mungkin banyak. Tujuan inilah yang menjadi pemotivasi yang mengarahkan sistem. Tanpa tujuan, sistem menjadi tak terarah dan tak terkendali. Tentu saja, tujuan antara satu sistem dengan sistem yang lain berbeda.

### **2. Masukan**

Masukan (input) sistem adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan yang diproses. Masukan dapat berupa hal-hal yang berwujud (tampak secara fisik) maupun yang tidak tampak. Contoh masukan yang berwujud adalah bahan mentah, sedangkan contoh yang tidak berwujud adalah informasi.

### **3. Proses**

Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna dan lebih bernilai, misalnya berupa informasi dan produk, tetapi juga bisa berupa hal-hal yang tidak berguna, misalnya saja sisa pembuangan atau limbah. Pada pabrik kimia, proses dapat berupa bahan mentah. Pada rumah sakit, proses dapat berupa aktivitas pembedahan pasien.

#### **4. Keluaran**

Keluaran (output) merupakan hasil dari pemrosesan. Pada sistem informasi, keluaran bisa berupa suatu informasi, saran, cetakan laporan, dan sebagainya.

#### **5. Batas**

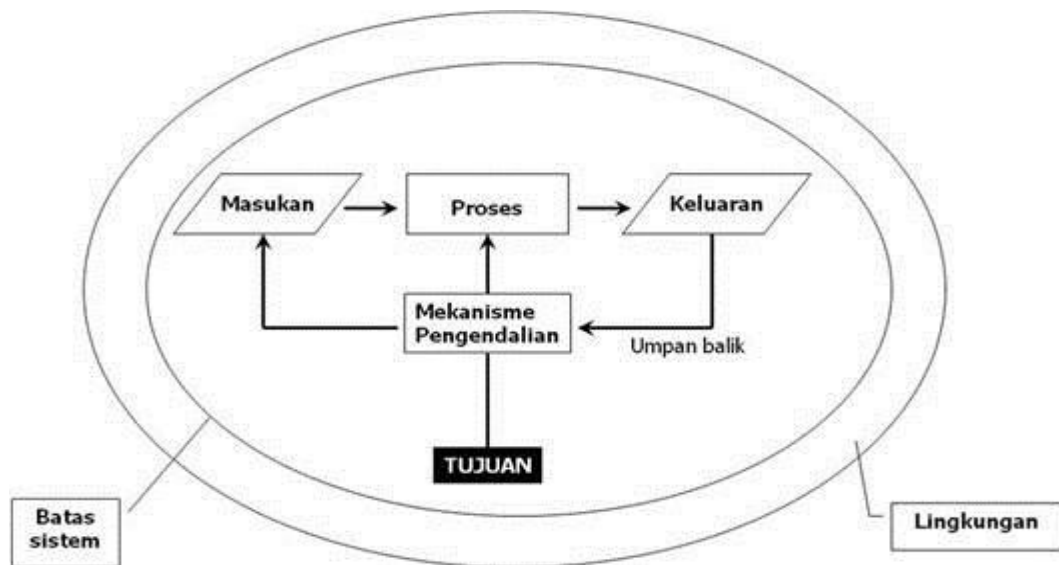
Yang disebut batas (boundary) sistem adalah pemisah antara sistem dan daerah di luar sistem (lingkungan). Batas sistem menentukan konfigurasi, ruang lingkup, atau kemampuan sistem. Sebagai contoh, tim sepakbola mempunyai aturan permainan dan keterbatasan kemampuan pemain. Pertumbuhan sebuah toko kelontong dipengaruhi oleh pembelian pelanggan, gerakan pesaing dan keterbatasan dana dari bank. Tentu saja batas sebuah sistem dapat dikurangi atau dimodifikasi sehingga akan mengubah perilaku sistem. Sebagai contoh, dengan menjual saham ke publik, sebuah perusahaan dapat mengurangi keterbatasan dana.

#### **6. Mekanisme Pengendalian dan Umpan Balik**

Mekanisme pengendalian (*Control Mechanism*) diwujudkan dengan menggunakan umpan balik (*Feedback*) yang mencuplik keluaran. Umpan balik ini digunakan untuk mengendalikan baik masukan maupun proses. Tujuannya adalah untuk mengatur agar sistem berjalan sesuai dengan tujuan.

#### **7. Lingkungan**

Lingkungan adalah segala sesuatu yang berada diluar sistem. Lingkungan bisa berpengaruh terhadap operasi sistem dalam arti bisa merugikan atau menguntungkan sistem itu sendiri. (Abdul Kadir : 2014 ; 62).



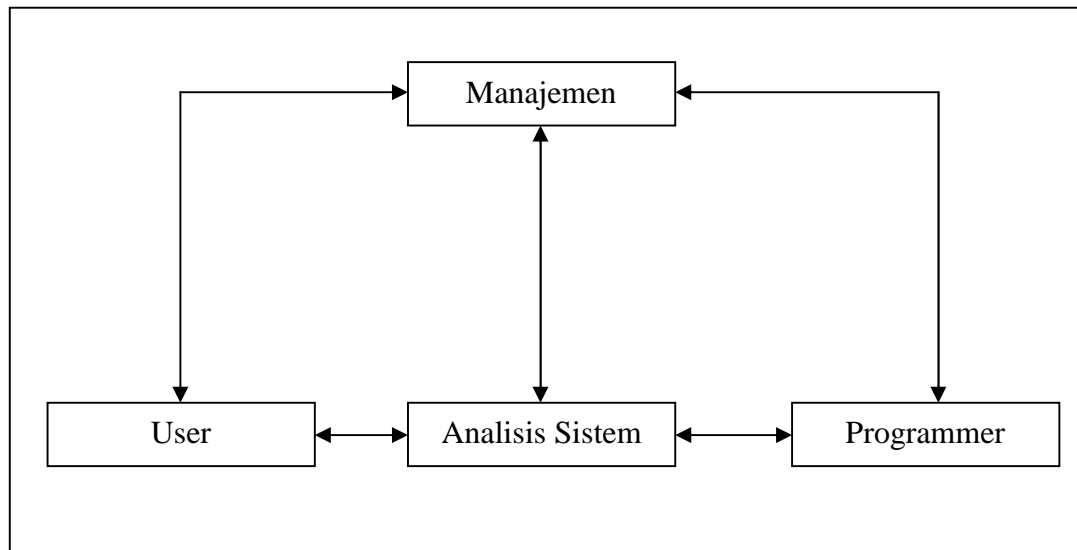
**Gambar II.3. Elemen – Elemen Sistem**  
(Sumber : Abdul Kadir : 2014 ; 62)

### II.2.5 Analisis Sistem

Suatu sistem akan dirancang oleh satu orang atau item. Orang yang merancang sistem ini disebut Analisis Sistem (*System Analyst*).

Terdapat beberapa definisi yang mendefinisikan sistem analis, diantaranya adalah ;

1. Seseorang yang menggunakan pengetahuan aplikasi komputer yang dimiliki untuk memecahkan masalah – masalah bisnis, dibawah petunjuk manajer sistem.
2. Seseorang yang bertanggungjawab untuk menterjemahkan kebutuhan – kebutuhan si pemakai sistem (*User*) ke dalam spesifikasi teknik yang diperlukan oleh Programmer dan diawasi oleh Manajemen.



**Gambar II.4. Ilustrasi Keberadaan Analisis Sistem  
(Sumber : Hamim Tohari : 2014 ; 11)**

## **II.3. Informasi**

### **II.3.1 Pengertian Informasi**

Informasi adalah data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut. Menurut Shannon dan Weaver (Kroenke, 1992) mendefinisikan informasi adalah “*jumlah ketidakpastian yang dikurangi ketika sebuah pesan diterima*”. (Abdul Kadir : 2014 ; 45).

Informasi adalah suatu aset penting bagi suatu institusi atau organisasi. Oleh karena itu, informasi harus berkualitas, dijaga dan dipelihara dengan baik. (Hamim Tohari : 2014 ; 7).

### **II.3.2 Kualitas Informasi**

Informasi harus berkualitas, dijaga, dan dipelihara dengan baik. Sedangkan kualitas dari suatu informasi tergantung dari tiga hal, yaitu ;

#### **1. Akurat.**

Kecocokan antara informasi dengan kejadian – kejadian atau objek – objek yang diwakilinya. Misalnya, laporan inventaris yang tidak akurat menyebutkan bahwa terdapat 15 unit barang yang tersisa di gudang . Kenyataannya, masih ada 51 unit barang di dalam gudang.

#### **2. Tepat Waktu.**

Informasi yang tidak tepat waktu akan menjadi informasi yang tidak berguna atau tidak dapat digunakan untuk membantu mengambil keputusan. Misalnya, informasi jadwal ujian seorang mahasiswa disampaikan setelah kegiatan ujian dilaksanakan atau diselenggarakan. Maka, informasi tersebut jadi tidak berguna lagi.

#### **3. Relevan.**

Informasi yang disajikan sebaiknya terkait dengan keputusan yang akan diambil oleh pengguna informasi tersebut. Misalnya, seorang manajer yang akan memberi kredit kepada pelanggan, bisa melihat laporan keuangan pelanggan tersebut karena laporan tersebut terkait dengan keputusan yang akan dibuat, yaitu memberikan atau tidak memberikan kredit kepada pelanggan tersebut. (Hamim Tohari : 2014 ; 8).

## **II.4 Sistem Informasi**

Sistem informasi merupakan suatu tatanan yang saling terkait antara unsur data, software, hardware, sumberdaya manusia dan kelembagaan serta aturan mainnya. Mengembangkan sistem informasi berarti mengembangkan seluruh unsur tersebut secara menyeluruh, tidak bisa dilakukan secara menyeluruh, tidak bisa dilakukan secara parsial atau sendiri-sendiri. (Renita Astri : 2013 ; 73).

Sistem informasi adalah suatu sistem yang dibuat oleh manusia yang terdiri dari komponen – komponen dalam organisasi untuk mencapai suatu tujuan yaitu menghasilkan informasi atau sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan atau untuk mengendalikan organisasi. (Angga Cahyo Saputro : 2012 ; 17).

Sistem Informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat managerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu laporan-laporan yang diperlukan. (Renita Astri : 2013 ;73).

### **II.4.1 Komponen – Komponen Sistem Informasi**

Sistem informasi memiliki beberapa komponen – komponen sistem informasi, diantaranya adalah sebagai berikut ;

### **1. Perangkat Keras (Hardware)**

Komponen Perangkat Keras (*Hardware*) ini mencakup piranti – piranti fisik seperti komputer dan printer.

### **2. Perangkat Lunak (Software)**

Komponen perangkat lunak (*Software*) atau program yaitu sekumpulan instruksi yang memungkinkan perangkat keras untuk memproses data.

### **3. Aturan (Prosedur)**

Komponen aturan (*Prosedur*) merupakan sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.

### **4. Orang (User)**

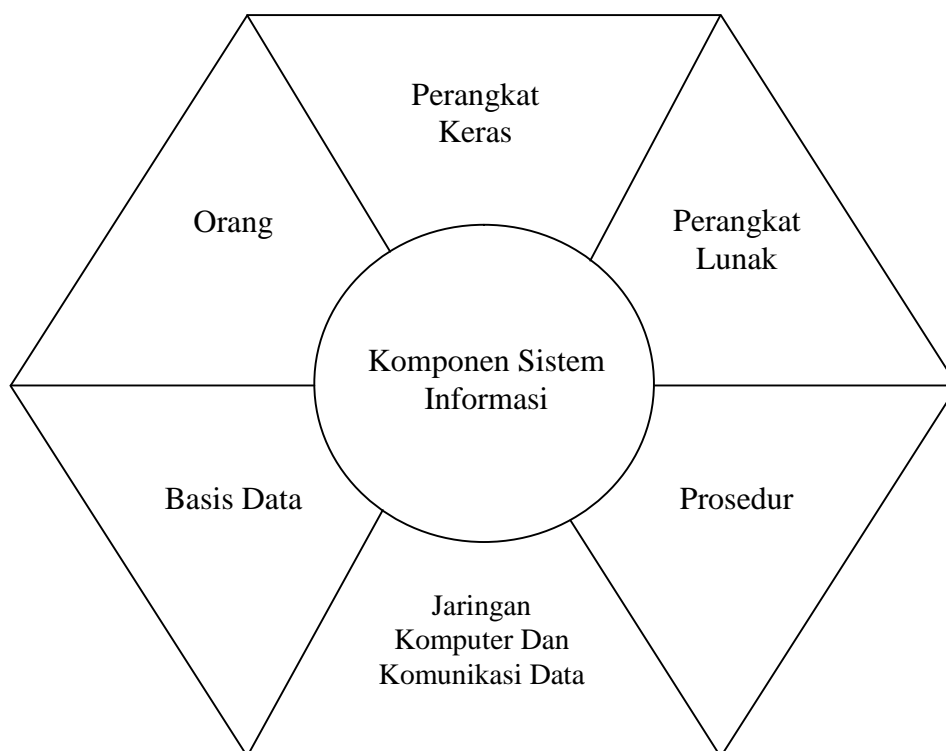
Komponen orang (*User*) yaitu semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan dan penggunaan keluaran sistem informasi.

### **5. Basis Data (Database)**

Komponen basis data (*Database*) yaitu sekumpulan tabel hubungan, dan lain – lain yang berkaitan dengan penyimpanan data.

### **6. Jaringan Komputer dan Komunikasi Data**

Komponen jaringan komputer dan komunikasi data yaitu sistem penghubung yang memungkinkan sumber (*Resources*) dipakai secara bersama atau diakses oleh sejumlah pemakai. (Abdul Kadir : 2014 ; 71-72).



**Gambar II.5. Komponen – Komponen Sistem Informasi**  
(Sumber : Abdul Kadir : 2014 ; 72)

#### **II.4.2 Blok Sistem Informasi**

Sistem informasi memiliki beberapa blok – blok sistem informasi, diantaranya adalah sebagai berikut ;

##### **1. Blok Masukan**

Blok masukan semua input yang mewakili data berupa dokumen-dokumen dasar yang masuk kedalam sistem informasi.

##### **2. Blok Model**

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

### **3. Blok Keluaran**

Blok keluaran berupa produk sistem informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai.

### **4. Blok Teknologi**

Blok teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian sistem secara keseluruhan.

### **5. Blok Basis Data**

Blok basis data merupakan kumpulan dari data yang saling berhubungan satu sama lainnya, tersimpan dalam perangkat keras komputer, dan digunakan perangkat lunak untuk memanipulasinya.

### **6. Blok Kendali**

Blok kendali perancangan pengendalian yang dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang merusak sistem dapat dicegah. (Renita Astri : 2013 ;73-74).

## **II.5. Aplikasi**

Aplikasi adalah penggunaan dalam suatu perangkat komputer, instruksi (*Instruction*) atau pernyataan (*Statement*) yang disusun hingga sedemikian rupa komputer dapat memproses masukan (*Input*) menjadi keluaran (*Output*). Aplikasi dapat digolongkan menjadi beberapa kelas, antara lain aplikasi perusahaan (*Enterprise*), aplikasi infrastruktur perusahaan, aplikasi

informasi kerja, aplikasi media dan hiburan, aplikasi pendidikan, aplikasi pengembangan media, aplikasi rekayasa produk. (Syamsu Rizal, Eko Retnadi, Andri Ikhwana, Vol. 10 No. 1 2013).

## **II.6. Keputusan**

### **II.6.1 Pengertian Keputusan**

Keputusan adalah suatu proses yang dilaksanakan seseorang berdasarkan pengetahuan dan informasi yang ada dengan harapan atau tujuan. Keputusan dapat diambil dari beberapa alternatif – alternatif yang ada. (*Marimin, Maghfiroh Nurul* : 2010 ; 16).

### **II.6.2 Jenis – Jenis Keputusan**

Keputusan–keputusan yang dibuat pada dasarnya dikelompokkan dalam dua jenis, antara lain (Herbert A. Simon ):

#### **1. Keputusan Terprogram**

Keputusan terprogram adalah keputusan yang dapat dispesifikasikan sebelumnya sebagai seperangkat aturan atau prosedur keputusan.

#### **2. Keputusan Tidak Terprogram**

Keputusan tidak terprogram adalah keputusan yang terjadi hanya satu kali atau berubah-ubah setiap saat ketika diperlukan.

Keputusan dalam suatu sistem keputusan terbuka (berada dalam suatu lingkungan yang rumit dan sebagian tak diketahui) adalah merupakan keputusan tidak terprogram karena tidak mungkin menspesifikasikan semua

faktor – faktornya sebelum melakukan pengambilan keputusan. (Dahria Muhammad, et al : Mei 2014).

## **II.7. Pengertian Pengambilan Keputusan**

Pengambilan keputusan merupakan proses pemilihan alternatif tindakan untuk mencapai tujuan atau sasaran tertentu. Pengambilan keputusan dilakukan dengan pendekatan sistematis terhadap permasalahan melalui proses pengumpulan data menjadi informasi serta ditambah dengan faktor–faktor yang perlu dipertimbangkan dalam pengambilan keputusan. (Dahria Muhammad, et al : Mei 2014).

## **II.8. Pengertian Sistem Pendukung Keputusan**

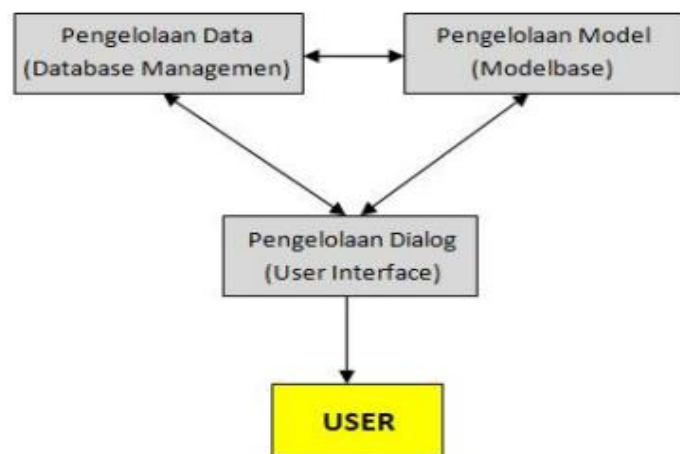
Sistem Pendukung Keputusan (*Decision Support System (DSS)*) hampir sama dengan sistem informasi manajemen tradisional karena keduanya tergantung pada basisdata sebagai sumber data. SPK menekankan pada fungsi pendukung pembuatan keputusan diseluruh tahap-tahapnya, sebagai pendamping keputusan aktual yang masih dibuat oleh wewenang eksekutif sebagai pembuat keputusan. (Pratiwi Heny, September 2014).

Sistem Pendukung Keputusan merupakan suatu sistem interaktif yang mendukung keputusan dalam proses pengambilan keputusan melalui alternatif – alternatif yang diperoleh dari hasil pengolahan data, informasi dan rancangan model. (Dahria Muhammad, et al : Mei 2014).

### II.8.1 Komponen – Komponen Dalam Sistem Pendukung Keputusan

Sistem pendukung keputusan memiliki beberapa komponen – komponen sistem, diantaranya adalah sebagai berikut ;

1. Subsistem Manajemen Basis Data
2. Subsistem Manajemen Model
3. Subsistem Dialog

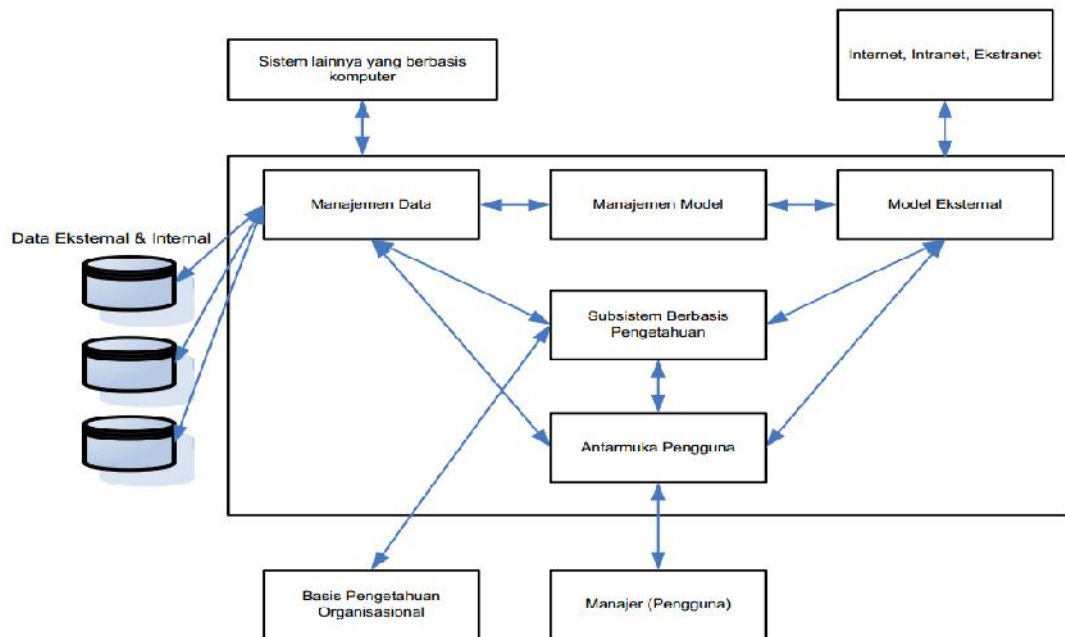


**Gambar II.6. Komponen – Komponen Dalam Sistem Pendukung Keputusan**  
(Sumber : Dahria Muhammad, et al: Mei 2014)

### II.8.2. Arsitektur *Decision Support System* (DSS)

*Decision Support System* (DSS) memiliki beberapa Arsitektur sistem, diantaranya adalah sebagai berikut ;

1. Subsistem Manajemen Basis Data
2. Subsistem Manajemen Model
3. Subsistem Antarmuka Pengguna
4. Manajemen Berbasis Pengetahuan



**Gambar II.7. Arsitektur rsitektur *Decision Support System* (DSS)  
(Sumber : Intan Oktariani, Jaidan Jauhari : April 2011)**

### 1. Subsistem Manajemen Data

Subsistem manajemen data memasukkan satu basisdata yang berisi data yang relevan untuk situasi dan dikelola oleh perangkat lunak yang disebut sistem manajemen basis data (DBMS / Data Base Manajemen System). Subsistem manajemen data bisa diinterkoneksi dengan data warehouse perusahaan, suatu repository untuk data perusahaan yang relevan dengan pengambilan keputusan.

### 2. Subsistem Manajemen Model

Subsistem manajemen model merupakan paket perangkat lunak yang memasukkan model keuangan, statistic, ilmu manajemen, atau model kuantitatif lain yang memberikan kapabilitas analitik dan manajemen perangkat lunak yang tepat. Bahasa-bahasa permodelan untuk membangun

model-model kustom juga dimasukkan. Perangkat lunak itu sering disebut Sistem Manajemen Basis Model (MBMS). Komponen tersebut dapat dikoneksikan ke penyimpanan korporat atau eksternal yang ada pada model.

Langkah-langkah permodelan dalam sistem pendukung keputusan, yaitu :

a. Fase Intelligence

Pada langkah ini, sasaran ditentukan dan dilakukan pencarian prosedur, pengumpulan data, identifikasi kepemilikan masalah, klasifikasi masalah, hingga akhirnya terbentuk sebuah pernyataan masalah. Kepemilikan masalah berkaitan dengan bagian apa yang akan dibangun oleh Sistem Pendukung Keputusan dan apa tugas dari bagian tersebut sehingga model tersebut bisa relevan dengan kebutuhan si pemilik masalah.

b. Perancangan (Design)

Pada tahapan ini, akan diformulasikan model yang akan digunakan dan Kriteria – Kriteria yang ditentukan. Setelah itu, dicari alternatif model yang bisa menyelesaikan permasalahan tersebut. Langkah selanjutnya adalah memprediksi keluaran yang mungkin. Kemudian ditentukan variabel-variabel model.

c. Pemilihan (Choice)

Setelah pada tahap desain ditentukan berbagai alternatif model beserta variabel-variabelnya, pada tahapan ini akan dilakukan pemilihan modelnya, termasuk solusi dari model tersebut. Selanjutnya, dilakukan analisis sensitivitas, yakni dengan mengganti beberapa variabel.

d. Membuat Sistem Pendukung Keputusan

Setelah menentukan modelnya, berikutnya adalah mengimplementasikannya dalam aplikasi Sistem Pendukung Keputusan.

**3. Subsistem Antarmuka Pengguna**

Pengguna berkomunikasi dengan memerintahkan sistem pendukung keputusan melalui subsistem tersebut. Pengguna adalah bagian yang dipertimbangkan dari sistem. Para peneliti menegaskan bahwa beberapa kontribusi unik dari sistem pendukung keputusan berasal dari interaksi yang intensif antara komputer dan pembuat keputusan.

**4. Subsistem Manajemen Berbasis Pengetahuan**

Subsistem tersebut mendukung semua subsistem lain atau bertindak langsung sebagai suatu komponen independen dan bersifat opsional. Selain memberikan intelegensi untuk memperbesar pengetahuan si pengambil keputusan, subsistem tersebut bisa diinterkoneksi dengan repository pengetahuan perusahaan (bagian dari sistem manajemen pengetahuan), yang kadang – kadang disebut basis pengetahuan organisasional. Berdasarkan penjelasan diatas, sistem pendukung keputusan harus mencakup 3 komponen utama dari DBMS, MBMS, dan antarmuka pengguna. Sedangkan subsistem manajemen berbasis pengetahuan adalah opsional, tetapi bisa memberikan banyak manfaat karena memberikan intelegensi bagi ketiga komponen utama tersebut. (Intan Oktariani, Jaidan Jauhari : April 2011)

## **II.9. Metode Multifactor Evaluation Process (MFEP)**

### **II.9.1 Pengertian Metode Multifactor Evaluation Process (MFEP)**

*Metode Multifactor Evaluation Process (MFEP)* ini merupakan metode kuantitatif yang menggunakan "*Weighting System*". Dan dalam pengambilan keputusan multi faktor, pengambil keputusan secara subyektif dan intuitif menimbang berbagai faktor yang mempunyai pengaruh penting terhadap alternatif pilihan mereka. Untuk keputusan yang berpengaruh secara strategis, lebih dianjurkan menggunakan sebuah pendekatan kuantitatif seperti MFEP. (Nitbani Debi Mariza, Nopember 2012).

Dalam metode MFEP ini pengambilan keputusan dilakukan dengan memberikan pertimbangan subyektif dan intuitif terhadap yang dianggap penting. Pertimbangan-pertimbangan tersebut berupa pemberian bobot (*weighting system*) atas multifactor yang terlibat dan dianggap penting tersebut. Langkah dalam metode MFEP ini yang pertama adalah menentukan faktor – faktor yang dianggap penting, yang selanjutnya membandingkan faktor-faktor tersebut sehingga diperoleh urutan faktor berdasarkan kepentingannya dari yang terpenting, kedua terpenting dan seterusnya. Proses pemilihan alternative terbaik menggunakan "*Weighting system*", dimana metode tersebut merupakan metode kuantitatif, disebut sebagai metode "*Multifactor Evaluation Process (MFEP)*". (Dahria Muhammad, et al: Mei 2014).

### II.9.2 Konsep Dasar Penggunaan Metode *Multifactor Evaluation Process* (MFEP)

Dibawah ini merupakan konsep dasar atau langkah – langkah proses perhitungan menggunakan metode *Multifactor Evaluation Process* (MFEP), yaitu:

1. Menentukan faktor dan bobot faktor dimana total pembobotan harus sama dengan 1 ( pembobotan = 1), yaitu factor weight.
2. Mengisikan nilai untuk setiap faktor yang mempengaruhi dalam pengambilan keputusan dari data-data yang akan diproses, nilai yang dimasukkan dalam proses pengambilan keputusan merupakan nilai objektif, yaitu sudah pasti factor evaluation yang nilainya antara 0 -1.
3. Proses perhitungan weight evaluation yang merupakan proses perhitungan bobot antara factor weight dan factor evaluation dengan serta penjumlahan seluruh hasil weight evaluations untuk memperoleh total hasil evaluasi.  
(Ahmad Khaidir, Volume : VI, Nomor: 3, April 2014).

### II.9.3 Algoritma Metode Metode *Multifactor Evaluation Process* (MFEP)

Dibawah ini merupakan Algoritma atau proses perhitungan menggunakan metode *Multifactor Evaluation Process* (MFEP), yaitu:

$$WE = FW \times E$$

$$WE = (FW \times E)$$

Keterangan :

WE = Weighted Evaluation

FW = Factor Weight

E = Evaluation

WE = Total Weighted Evaluation

Maka perhitungan perkalian antara nilai bobot weight dengan nilai bobot evaluation sesuai dengan evaluasi pihak sekolah pada setiap calon siswa baru.

(Ahmad Khaidir : 2014 ; 150).

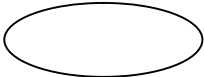
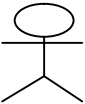
## **II.10. *Unified Modeling Language (UML)***

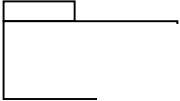
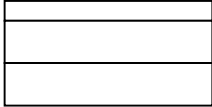
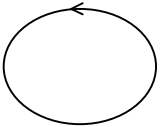
*Unified Modeling Language (UML)* adalah bahasa pemodelan untuk perangkat lunak yang berorientasi objek. Pemodelan digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah untuk dipelajari dan dipahami. (Syamsu Rizal, Eko Retnadi, Andri Ikhwana, Vol. 10 No. 1 2013).

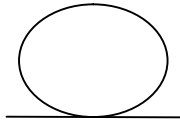


*Unified Modeling Language (UML)* adalah sistem arsitektur yang bekerja dalam OOAD (*Object-Oriented Analysis Design*) dengan satu bahasa yang konsisten untuk menentukan, visualisasi, mengkontruksi dan mendokumentasi artifact (sepotong informasi yang digunakan atau dihasilkan dalam suatu proses rekayasa software, dapat berupa model, deskripsi, atau software) yang terdapat dalam sistem software. (Angga Cahyo Saputro : 2012 ; 24).

*Unified Modeling Language* (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML juga merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. (Gellysa Urva, Helmi Fauzi Siregar : 2015).

**Tabel II.1. Daftar Simbol-simbol dalam *Unified Modelling Language* (UML)**

No.	Bentuk Simbol	Nama Simbol	Keterangan
1.		<i>Usecase</i>	Use Case menggambarkan fungsionalitas yang disediakan sistem sebagai unit – unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama Use Case. (Gellysa Urva, Helmi Fauzi Siregar : 2015).
2.		<i>Actor</i>	Aktor adalah abstraction dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian

			tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan use case, tetapi tidak memiliki control terhadap Use Case. (Gellysa Urva, Helmi Fauzi Siregar : 2015).
3.		<i>Package</i>	Package antara aktor dan use case, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data. (Gellysa Urva, Helmi Fauzi Siregar : 2015).
4.		<i>Class</i>	<i>Class</i> merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. (Gellysa Urva, Helmi Fauzi Siregar : 2015).
5.		<i>Control</i>	Control class, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan

			bisnis yang melibatkan berbagai objek. (Gellysa Urva, Helmi Fauzi Siregar : 2015).
6.		<i>Entity</i>	Entity Class, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
7.		<i>Activity</i>	Activites, menggambarkan suatu proses/kegiatan bisnis. (Gellysa Urva, Helmi Fauzi Siregar : 2015).
8.		<i>State</i>	State, pembagian activity diagram untuk menunjukkan siapa melakukan apa. (Gellysa Urva, Helmi Fauzi Siregar : 2015).

(Sumber : Gita Larasati Sumaja : 2013)

UML menyediakan jenis-jenis diagram yang dapat dikelompokkan berdasarkan sifatnya, yaitu:

1. *Use-Case Diagram* adalah suatu kumpulan urutan interaksi diantara user dengan sistem untuk mencapai suatu tujuan dimana use case ini menggambarkan kebutuhan fungsional suatu sistem tanpa menampilkan struktur internal sistem.

2. *Class Diagram* adalah Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.
3. *Activity Diagram* adalah Representasi secara grafis dari proses dan control flow dan berfungsi untuk memperlihatkan alur dari satu aktivitas ke aktivitas yang lain serta menggambarkan perilaku yang kompleks.
4. *Sequence Diagram* adalah Sequence diagram digunakan untuk menggambarkan event yang dilakukan aktoreksternal pada sistem atauinter sistem event dilihat dalam satu use case. (Gita Larasati Sumaja : 2013).

## **II.11. Use Case Diagram**

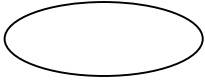
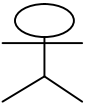

### **II.11.1 Pengertian Use Case Diagram**

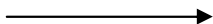
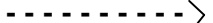

*Use Case Diagram* adalah rangkaian atau uraian sekelompok yang saling berhubungan atau terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah sektor. (Hamim Tohari : 2014 ; 7).

*Use Case Diagram* menjelaskan manfaat sistem jika dilihat menurut pandangan orang yang berbeda (*Actor*). Diagram ini menunjukkan *fungsionalitas* suatu sistem atau kelas dan bagaimana sistem berinterkasi dengan dunia luar. (Angga Cahyo Saputro, 2012).

*Use Case Diagram* menjelaskan manfaat sistem jika dilihat menurut pandangan orang yang berbeda (*Actor*). Diagram ini menunjukkan *fungsionalitas* suatu sistem atau kelas dan bagaimana sistem berinterkasi dengan dunia luar. (Angga Cahyo Saputro : 2012).

Tabel II.2. Daftar Simbol-simbol dalam *Use Case Diagram*

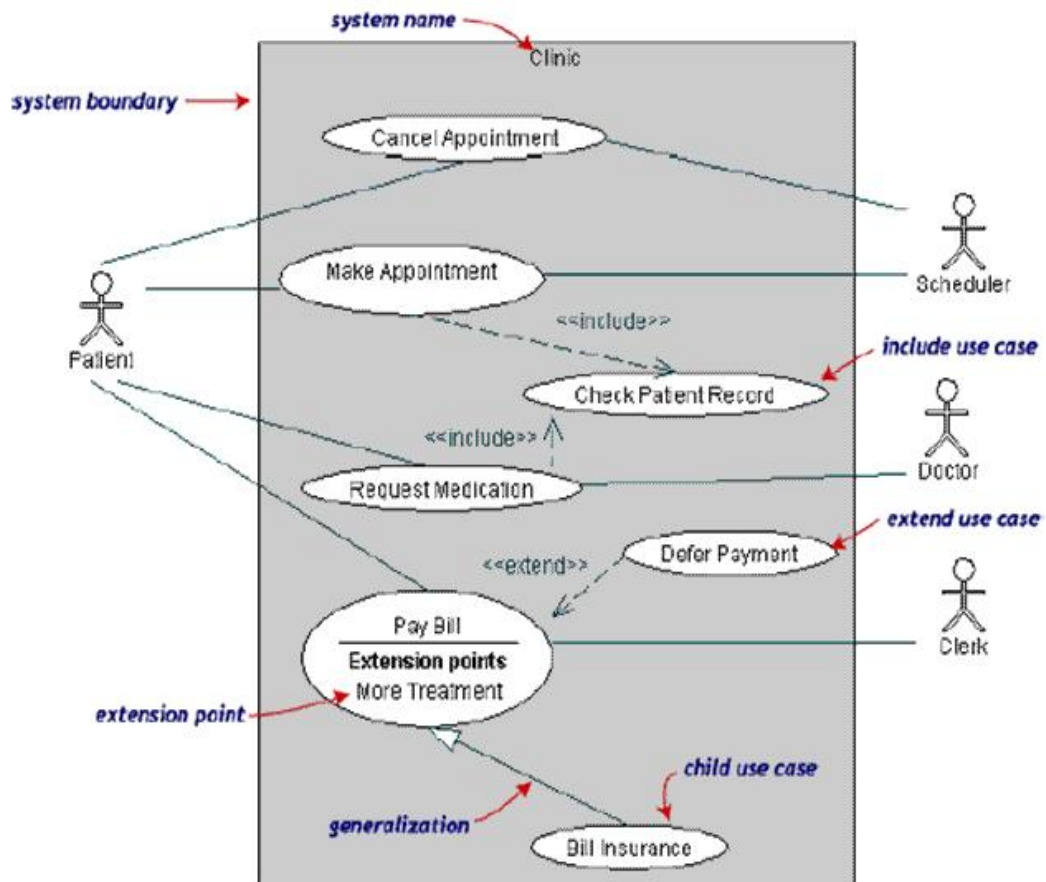
No.	Bentuk Simbol	Nama Simbol	Keterangan
1.		<i>Use Case</i>	Use Case menggambarkan fungsionalitas yang disediakan sistem sebagai unit – unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama Use Case.
2.		<i>Actor</i>	Aktor adalah abstraction dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan use case, tetapi tidak memiliki control terhadap Use Case.
3.		<i>Asosiasi</i>	Asosiasi antara aktor dan use case, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasi aliran data.

4.		<i>Asosiasi Antar Aktor</i>	Asosiasi antara aktor dan use case yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
5.		<i>Include</i>	Include merupakan di dalam use case lain (required) atau pemanggilan use case oleh use case lain, contohnya adalah pemanggilan sebuah fungsi program.
6.		<i>Extend</i>	Extend merupakan perluasan dari use case lain jika kondisi atau syarat terpenuhi.

(Sumber : Gellysa Urva, Helmi Fauzi Siregar : 2015)

### II.11.2 Fungsi *Use Case Diagram*

*Use Case Diagram* digunakan untuk membentuk tingkah laku benda dalam sebuah model serta direalisasikan oleh sebuah kolaborasi. (Hamim Tohari : 2014 ; 47).



**Gambar II.8. Contoh Use Case Diagram**  
(Sumber : Hamim Tohari : 2014 ; 49)

### II.11.3 Elemen-elemen *Use Case Diagram*

Use Case Diagram memiliki beberapa elemen-elemen *Use Case*, diantaranya adalah sebagai berikut ;

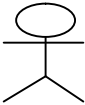
#### 1. Sistem

Sistem menyatakan batasan – batasan sistem dalam relasi dengan actor- actor yang menggunakannya (diluar sistem) dan fitur – fitur yang harus disajikan (dalam sistem).

## 2. Actor

Actor dapat berupa manusia, sistem atau device yang memiliki peranan dalam keberhasilan operasi dari sistem. Sistem yang digambarkan dengan icon yang memungkinkan bervariasi.

**Tabel II.3. Bentuk Simbol Dari Actor**

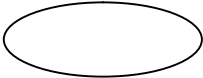
No.	Bentuk Simbol	Nama Simbol	Keterangan
1.		<i>Actor</i>	<p>Aktor adalah abstraction dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan use case, tetapi tidak memiliki control terhadap Use Case. (Gellysa Urva, Helmi Fauzi Siregar : 2015).</p>

(Sumber : Gellysa Urva, Helmi Fauzi Siregar : 2015)

### 3. Use Case

*Use Case* mengidentifikasi fitur kunci dari sistem. Tanpa fitur ini, maka sistem tidak akan memenuhi permintaan *User / Actor*. Setiap *Use Case* mengekspresikan *Goal* dari sistem yang harus dicapai.

**Tabel II.4. Bentuk Simbol Dari Use Case**


No.	Bentuk Simbol	Nama Simbol	Keterangan
1.		<i>Use Case</i>	<p>Use Case menggambarkan fungsionalitas yang disediakan sistem sebagai unit – unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama Use Case.</p>

(Sumber : Gellysa Urva, Helmi Fauzi Siregar : 2015)

### 4. Assosiation

Mengidentifikasi interaksi antara setiap *Actor* tertentu dengan setiap *Use Case* tertentu. Digambarkan dengan garis antara *Actor* terhadap *Use Case* yang bersangkutan.

**Tabel II.5. Bentuk Simbol Dari Asosiasi**

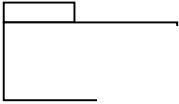
No.	Bentuk Simbol	Nama Simbol	Keterangan
1.		<i>Asosiasi</i>	Asosiasi antara aktor dan use case, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.

(Sumber : Gellysa Urva, Helmi Fauzi Siregar : 2015)

## 5. Stereotape

Memungkinkan perluasan *Unified Modeling Language* (UML) terhadap modifikasinya. Dimana *Unified Modeling Language* (UML) juga berperan sebagai *kualifier* pada suatu elemen – elemen model.

**Tabel II.6. Bentuk Simbol Dari Package**

No.	Bentuk Simbol	Nama Simbol	Keterangan
1.		<i>Package</i>	Package antara aktor dan use case, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data. (Gellysa Urva, Helmi Fauzi Siregar : 2015).

(Sumber : Gita Larasati Sumaja : 2013)

## 6. Dependency

Dependency juga sering disebut sebagai <<*include*>>.

**Tabel II.7. Bentuk Simbol Dari Include**

No.	Bentuk Simbol	Nama Simbol	Keterangan
1.	----->	<i>Include</i>	Include merupakan di dalam use case lain (required) atau pemanggilan use case oleh use case lain, contohnya adalah pemanggilan sebuah fungsi program.

(Sumber : Gellysa Urva, Helmi Fauzi Siregar : 2015)

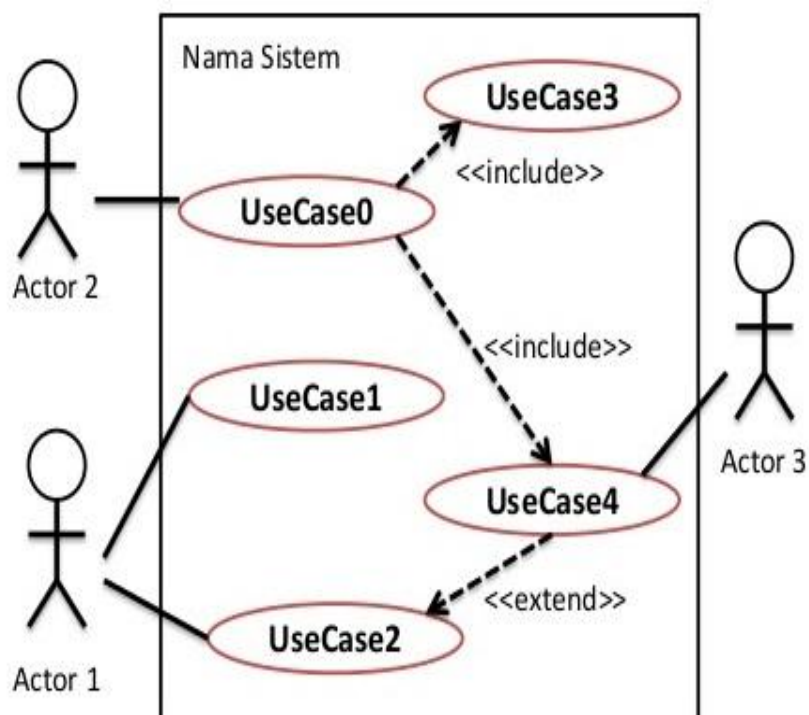
## 7. Generalization

Mengidentifikasi relasi antara Actor dan dua Use Case. Selanjutnya meng-inherit dan menambahkan atau override sifat dari yang lainnya. (Hamim Tohari : 2014 ; 52-54)

**Tabel II.8. Bentuk Simbol Dari Lifeline**

No.	Bentuk Simbol	Nama Simbol	Keterangan
1.	.....	<i>Lifeline</i>	Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang lifeline terdapat activation.

(Sumber : Gellysa Urva, Helmi Fauzi Siregar : 2015)






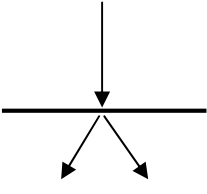
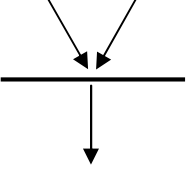
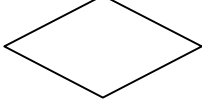
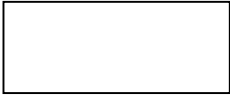
**Gambar II.9. Elemen-elemen *Use Case Diagram***  
(Sumber : Hamim Tohari : 2014 ; 52)

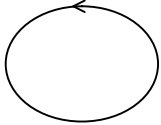

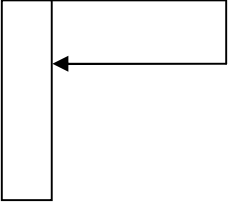

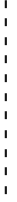
## II.12. *Activity Diagram*

*Activity diagram* digunakan untuk memodelkan alur kerja (*Workflow*) sebuah proses bisnis dan urutan aktivitas dalam suatu proses. Diagram ini sangat mirip dengan sebuah Flowchart karena kita dapat memodelkan sebuah alur kerja dari satu aktivitas ke aktivitas lainnya atau dari suatu aktivitas ke keadaan sesaat (*State*). (Angga Cahyo Saputro : 2012 ; 53).

*Activity Diagram* menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol – symbol yang digunakan dalam activity diagram yaitu : (Gellysa Urva, Helmi Fauzi Siregar : 2015).

Tabel II.9. Daftar Simbol-simbol dalam *Activiti Diagram*

No.	Bentuk Simbol	Nama Simbol	Keterangan
1.		<i>Start point</i>	Start point, diletakkan pada pojok kiri atas dan merupakan awal aktifitas
2.		<i>End point</i>	End point, akhir aktifitas.
3.		<i>Activites</i>	Activites, menggambarkan suatu proses/kegiatan bisnis
4.		<i>Fork</i>	Fork (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
5.		<i>Join</i>	Join (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
6.		<i>Decision Points</i>	Decision Points, menggambarkan pilihan untuk pengambilan keputusan, true, false.
7.		<i>Swimlane</i>	Swimlane, pembagian activity diagram untuk menunjukkan siapa melakukan apa.

8.		<i>Control Class</i>	Control class, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
9.		<i>Message</i>	Message, simbol mengirim pesan antar class.
10.		<i>Recursive</i>	Recursive, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
11.		<i>Activation</i>	Activation, activation mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
12.		<i>Lifeline</i>	Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang lifeline terdapat activation.

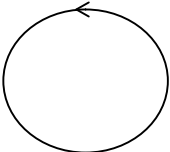
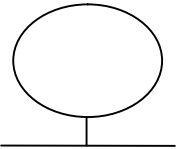
(Sumber : Gellysa Urva, Helmi Fauzi Siregar : 2015)

### II.13. *Sequence Diagram*

*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 seharusnya terjadi untuk menghasilkan sesuatu di dalam *Use Case*. (Angga Cahyo Saputro : 2012 ; 51).

*Sequence Diagram* menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam sequence diagram yaitu : (Gellysa Urva, Helmi Fauzi Siregar : 2015).

**Tabel II.10. Daftar Simbol-simbol dalam *Sequence Diagram***

No.	Bentuk Simbol	Nama Simbol	Keterangan
1.		<i>Entity Class</i>	Entity Class, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
2.		<i>Boundary Class</i>	Boundary Class, berisi kumpulan kelas yang menjadi interface atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan form cetak.

(Sumber : Gellysa Urva, Helmi Fauzi Siregar : 2015)

## II.14. *Class Diagram*

*Class diagram* membantu kita dalam visualisasi structure kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class diagram* memperlihatkan hubungan antarkelas dan penjelasan detail tiap-tiap kelas didalam model desain (Dalam Logical View) dari suatu sistem. (Angga Cahyo Saputro, 2012).

*Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan constraint yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (Class), Relasi, Associations, Generalization dan Aggregation, Atribut (Attributes), Operasi (Operations/Method), Visibility, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan multiplicity atau kardinaliti. (Gellysa Urva, Helmi Fauzi Siregar : 2015).

**Tabel II.11. Daftar Simbol-simbol dalam *Class Diagram***

No.	Bentuk Simbol	Keterangan
1.	1	Satu dan hanya satu
2.	0.. *	Boleh tidak ada atau 1 atau lebih
3.	1.. *	1 atau lebih
4.	0.. 1	Boleh tidak ada, maksimal 1

5.	n.. n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4
----	-------	---

(Sumber : Gellysa Urva, Helmi Fauzi Siregar : 2015)

## II.15. Microsoft Visual Basic 2010

Microsoft Visual Basic 2010 merupakan program aplikasi untuk membentuk program dari bahasa pemrograman klasik, legendaris dan tiada duanya yang paling banyak dipakai oleh jutaan programmer dan tercatat sebagai program yang paling disukai oleh mayoritas orang.(Edy Winarno ST, M.Eng. Ali Zaki, SmitDev Community : 2010).

Visual Basic 2010 terdapat di dalam paket software Microsoft Visual Basic 2010 yang dirilis pada bulan April 2010, didalam paket tersebut terdapat software lainnya seperti Visual C#, Visual C++, Visual F#, dan Web Development (ASP.Net), (XML Web Service), dan Visual Basic tentunya. (Muhammad Sadeli : 2011 ; 2).

## II.16. Basis Data

Basis data merupakan komponen terpenting dalam pembangunan SI, karena menjadi tempat untuk menampung dan mengorganisasikan seluruh data yang ada dalam sistem, sehingga dapat dieksplorasi untuk menyusun informasi-informasi dalam berbagai bentuk.

Basis data merupakan himpunan kelompok data yang saling berkaitan. Dengan basis data, pengguna dapat menyimpan data secara terorganisasi. Setelah data disimpan, informasi harus mudah diambil.

Prinsip utama basis data adalah pengaturan data dengan tujuan utama fleksibilitas dan kecepatan dalam pengambilan data kembali.

Bahasa basis data (database language) adalah suatu cara untuk berinteraksi atau berkomunikasi antara pemakai dengan basis data yang diatur dalam bahasa khusus yang ditetapkan oleh perusahaan.

### **II.16.1.Karakteristik Basis Data**

Dalam mempelajari sistem basis data, perlu diketahui beberapa karakteristik penting dari basis data, antara lain adalah ;

#### **1. Persistent.**

*Persistent* berarti bahwa data ditempatkan pada penyimpanan yang stabil seperti magnetic disk, tetapi tidak berarti bahwa data disimpan selama-lamanya. Data yang tidak lagi relevan atau tidak berguna akan dihilangkan.

#### **2. Shared**

*Shared* berarti bahwa basis data dapat memiliki banyak kegunaan dan pengguna. Banyak pengguna dapat menggunakan basis data pada saat yang bersamaan.

### 3. Interrelated

*Interrelated* berarti bahwa data yang disimpan sebagai unit yang berbeda dapat dihubungkan untuk menyediakan gambar keseluruhan. (Yanti Efendy : 2012).

## II.17. Database

Database merupakan suatu tempat untuk menampung data server. Dimana data tersebut nantinya akan diproses oleh program oleh *Perogrammer* dalam tahap pembuatan program. (M. Leo Agung : 2011).

Dalam suatu database terdapat 3 bagian bahasa pemrograman database, yaitu;

1. Data Definition Language (DDL),
2. Data Manipulation Language (DML), dan
3. Data Control Language (DCL). (Ivan Arifard Watung, Alicia A.E. Sinsuw, ST.MT, Sary D.E. Paturusi, ST ,M.Eng et,al : 2014).

### II.17.1 *Data Definition Language (DDL)*

*Data Definition Language (DDL)* merupakan singkatan dari Data Definition Language yang juga bagian dari structured query language (SQL). DDL berfungsi lebih ke dalam memanipulasi struktur dari database. DDL digunakan untuk membuat tabel atau menghapus tabel, membuat key atau indeks, membuat relasi Antartabel. (Ivan Arifard Watung, Alicia A.E. Sinsuw, ST.MT, Sary D.E. Paturusi, ST ,M.Eng et,al : 2014).

### **II.17.1.1 Fungsi Dari *Data Definition Language* (DDL)**

*Data Definition Language* (DDL) digunakan untuk mendefinisikan, mengubah, serta menghapus basis data dan objek-objek yang diperlukan dalam basis data, misalnya tabel, view, user, dan sebagainya. DDL biasanya digunakan oleh administrator basis data dalam pembuatan sebuah aplikasi basis data. Secara umum DDL yang digunakan adalah:

1. CREATE untuk membuat objek baru.
2. USE untuk menggunakan objek.
3. ALTER untuk mengubah objek yang sudah ada.
4. DROP untuk menghapus objek. (Adelia, Jimmy Setiawan : 2011 ; 115).

### **II.17.2 *Data Manipulation Language* (DML)**

*Data Manipulation Language* (DML) merupakan sekumpulan sintaks – sintaks bisa digunakan untuk melakukan proses insert, update atau delete ke dalam suatu database. (Ivan Arifard Watung, Alicia A.E. Sinsuw, ST.MT, Sary D.E. Paturusi, ST ,M.Eng et,al : 2014).

#### **II.17.2.1 Fungsi Dari *Data Manipulation Language* (DML)**

*Data Manipulation Language* (DML) digunakan untuk memanipulasi data yang ada dalam suatu tabel. Perintah-perintah yang umum dilakukan adalah:

1. SELECT untuk menampilkan data.
2. INSERT untuk menambahkan data baru.

3. UPDATE untuk mengubah data yang sudah ada.
4. DELETE untuk menghapus data. (Adelia, Jimmy Setiawan : 2011 ; 115).

### **II.17.3. Data Control Language (DCL)**

*Data Control Language (DCL)* merupakan kelompok perintah yang berisi untuk mengendalikan pangaksesan data DCL digunakan untuk menangani masalah keamanan dalam database server. (Ivan Arifard Watung, Alicia A.E. Sinsuw, ST.MT, Sary D.E. Paturusi, ST ,M.Eng et,al : 2014).

#### **II.17.3.1 Fungsi Dari Data Control Language (DCL)**

*Data Control Language (DCL)* digunakan untuk menangani masalah keamanan dalam database server. (Ivan Arifard Watung, Alicia A.E. Sinsuw, ST.MT, Sary D.E. Paturusi, ST ,M.Eng et,al : 2014).

### **II.18. Normalisasi**

Normalisasi merupakan suatu proses yang memudahkan desain struktur table secara benar sehingga query yang tak dapat ditanyakan tidak muncul. Disamping itu, normalisasi cenderung meminimumkan duplikasi data di dalam suatu basis data. Ini memiliki keunggulan dalam mengurangi ruang simpan yang dibutuhkan maupun mempercepat *query*. (Yanti Efendy : 2012).

## II.19. SQL Server

*SQL (Structured Query Language)* adalah sebuah bahasa yang dipergunakan untuk mengakses data dalam basis data relasional. Bahasa ini secara de facto merupakan bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua server basis data yang ada mendukung bahasa ini untuk melakukan manajemen datanya.

*SQL (Structured Query Language)* terdiri dari dua bahasa, yaitu *Data Definition Manipulation Language (DDL)*, Implementasi DDL dan DML Sistem Manajemen Basis Data (*SMBD*), namun secara umum implemen bahasa ini memiliki bentuk standar yang ditetapkan oleh *ANSI*. (Adelia, Jimmy Setiawan : 2011 ; 115).

Terdapat beberapa tipe data yang sering digunakan dalam pemrograman database SQL Server, antara lain adalah sebagai berikut ;

**Tabel II.12. Daftar Tipe Data**

No.	Tipe Data	Ukuran	Rentang Nilai
1.	Byte	1 byte	0 s/d +255
2.	Shortint	1 byte	-28 s/d +127
3.	Integer	2 byte	-32768 s/d 32767
4.	Word	2 bytes	0 s/d 65535
5.	Longint	4 bytes	2147483648 s/d 2147483647
6.	real	6 bytes	$2.9 \times 10^{-39}$ s/d $1.7 \times 10^{38}$
7.	single	4 bytes	$1.5 \times 10^{45}$ s/d $3.4 \times 10^{38}$
8.	double	8 bytes	$5.0 \times 10^{-324}$ s/d $1.7 \times 10^{308}$

			10308
<b>9.</b>	extended	10 bytes	$3.4 \times 10^{-4} \text{ s/d}$ $1.1 \times 10^4 \text{ s/d}$
<b>10.</b>	comp	8 bytes	$-9.2 \times 10^1 \text{ s/d}$ $9.2 \times 10^1 \text{ s/d}$

(Sumber : Th Arie Prabawati : 2015 ; 62 – 64)