

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Sistem**

##### **2.1.1 Pengertian sistem**

Pada dasarnya sistem adalah suatu kerangka dari prosedur-prosedur yang saling berhubungan, disusun sesuai dengan skema yang menyeluruh untuk melaksanakan suatu kegiatan atau fungsi utama dari perusahaan yang dihasilkan oleh suatu proses tertentu. Sistem bertujuan menyediakan informasi untuk membantu mengambil keputusan manajemen operasi perusahaan dari hari ke hari serta menyediakan informasi yang layak untuk pihak di luar perusahaan. Pengertian sistem yang dikemukakan oleh para ahli adalah sebagai berikut :

Menurut Susanto, dalam bukunya dengan judul Sistem Informasi Akutansi : “Sistem adalah kumpulan/group dari sub sistem/bagian/komponen apapun baik fisik ataupun non fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu.” (Susanto 2013)

Menurut Romney dan Steinbart : “Sistem adalah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan. Sebagian sistem terdiri dari sub sistem yang lebih kecil yang mendukung sistem yang lebih besar.” (Romney, Marshall & Steinbart, 2015)

Menurut Mulyadi : “Sistem adalah suatu jaringan prosedur yang dibuat menurut pola yang terpadu untuk melaksanakan kegiatan pokok perusahaan.” (Mulyadi, 2016)

### 2.1.2 Karakteristik Sistem

Suatu sistem mempunyai ciri-ciri karakteristik yang terdapat pada sekumpulan elemen yang harus dipahami dalam mengidentifikasi pembuatan sistem. Adapun karakteristik sistem yang dimaksud (Hutahean, 2015) adalah sebagai berikut :

#### 1. Komponen

Sistem terdiri dari sejumlah komponen yang saling berinteraksi dan bekerjasama untuk membentuk satu kesatuan. Komponen sistem dapat berupa sub sistem atau bagian-bagian dari sistem.

#### 2. Batasan sistem (*boundary*)

Daerah yang membatasi antara suatu sistem dengan sistem lainnya atau dengan lingkungan luar dinamakan dengan batasan sistem. Batasan sistem ini memungkinkan sistem dipandang sebagai satu kesatuan dan juga menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

#### 3. Lingkungan luar sistem (*environment*)

Apapun yang berada di luar batas dari sistem dan mempengaruhi sistem tersebut dinamakan dengan lingkungan luar sistem. Lingkungan luar yang bersifat menguntungkan wajib dipelihara dan yang merugikan harus dikendalikan agar tidak mengganggu kelangsungan sistem.

#### 4. Penghubung sistem (*interface*)

Media penghubung diperlukan untuk mengalirkan sumber-sumber daya dari sub sistem ke sub sistem lainnya dinamakan dengan penghubung sistem.

### 5. Masukkan sistem (*input*)

Energi yang dimasukkan ke dalam sistem dinamakan dengan masukan sistem (*input*) dapat berupa perawatan dan masukan sinyal. Perawatan ini berfungsi agar sistem dapat beroperasi dan masukan sinyal adalah energi yang diproses untuk menghasilkan keluaran (*output*).

### 6. Keluaran sistem (*output*)

Hasil dari energi yang telah diolah dan diklasifikasikan menjadi keluaran yang berguna dinamakan dengan keluaran sistem (*output*). Informasi merupakan contoh keluaran sistem.

### 7. Pengolah sistem

Untuk mengolah masukan menjadi keluaran diperlukan suatu pengolah yang dinamakan dengan pengolah sistem.

### 8. Sasaran sistem

Sistem pasti memiliki tujuan atau sasaran yang sangat menentukan *input* yang dibutuhkan oleh sistem dan keluaran yang dihasilkan.

## 2.1.3 Tujuan dan Manfaat Sistem

Adapun tujuan sistem menurut Azhar Susanto, dalam bukunya Sistem Informasi Akutansi adalah sebagai berikut : “Tujuan sistem merupakan target atau sasaran akhir yang ingin dicapai oleh suatu sistem. Agar supaya target tersebut bisa tercapai, maka target atau sasaran tersebut harus diketahui terlebih dahulu ciri-ciri atau kriterianya. Upaya mencapai suatu sasaran tanpa mengetahui ciri-ciri atau kriteria dari sasaran tersebut kemungkinan besar sasaran tersebut tidak akan pernah tercapai. Ciri-ciri atau kriteria dapat juga digunakan sebagai

tolak ukur dalam menilai suatu keberhasilan suatu sistem dan menjadi dasar dilakukannya suatu pengendalian” (Susanto, 2013).

Menurut (Krismiaji, 2015), tujuan umum penyusunan sistem informasi adalah :

1. Untuk memperbaiki informasi yang diberikan oleh sistem dalam kualitas, ketepatan waktu atau struktur dari informasi tersebut.
2. Untuk memperbaiki pengendalian akuntansi dan pengecekan intern, yang berarti memperbaiki daya andal informasi dan menyediakan catatan yang lengkap sebagai pertanggungjawaban dalam melindungi harta perusahaan.
3. Untuk menurunkan biaya dalam menyelenggarakan catatan akuntansi. Dari pengertian diatas dapat ditarik kesimpulan bahwa sistem merupakan kumpulan suatu komponen sistem yang saling berhubungan satu dengan yang lain untuk mencapai tujuan suatu kegiatan pokok perusahaan.

Menurut (Krismiaji, 2015), manfaat sistem informasi antara lain :

1. Meningkatkan aksesibilitas data yang tersaji secara tepat waktu dan akurat bagi para pemakai, tanpa mengharuskan adanya prantara sistem informasi.
2. Menjamin terjadinya kualitas dan keterampilan dalam memanfaatkan sistem informasi secara kritis.
3. Mengembangkan proses perencanaan yang efektif.
4. Mengidentifikasi kebutuhan-kebutuhan akan keterampilan pendukung sistem informasi.
5. Menetapkan investasi yang akan diarahkan pada sistem informasi.

6. Mengantisipasi dan memahami konsekuensi-konsekuensi ekonomis dari sistem informasi dan teknologi baru.
7. Memperbaiki produktivitas dalam aplikasi pengembangan dan pemeliharaan sistem.

## **2.2 Sistem Pendukung Keputusan**

### **2.2.1 Definisi Sistem Pendukung Keputusan**

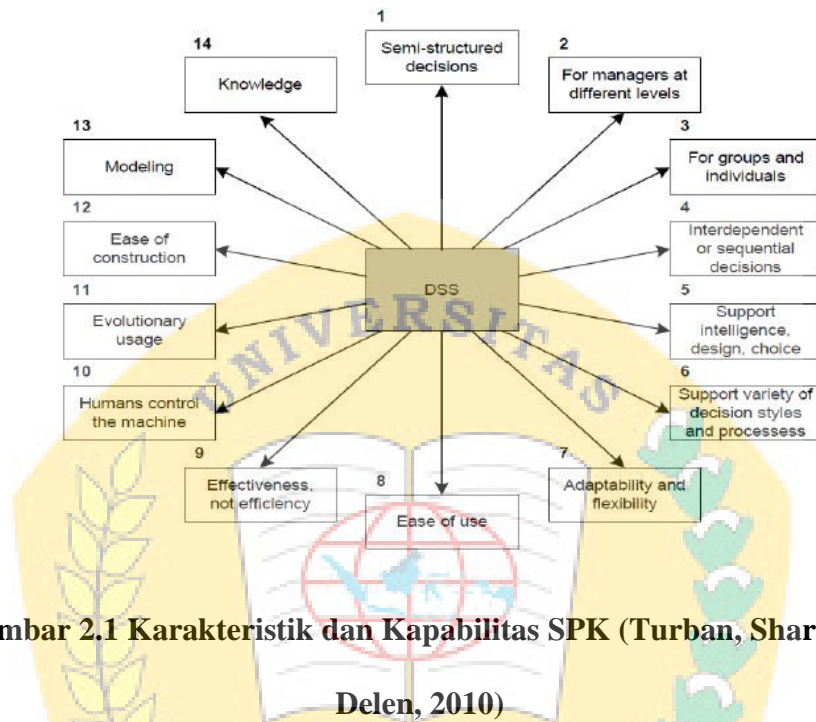
Konsep Sistem Pendukung Keputusan (SPK) pertama kali diungkapkan pada awal tahun 1970-an oleh Michael S. Scott Morton dengan istilah *Management Decision System*. Sistem tersebut adalah suatu sistem yang berbasis komputer yang ditunjukkan untuk membantu pengambil keputusan dengan memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur (Dewanto, 2015).

Menurut Turban, Sistem Pendukung Keputusan (SPK) merupakan sistem informasi yang berbasis komputer yang fleksibel, interaktif dan dapat diadaptasi, yang dikembangkan untuk mendukung solusi untuk masalah manajemen spesifik yang tidak terstruktur. Sistem Pendukung Keputusan menggunakan data, memberikan antarmuka pengguna yang mudah dan dapat menggabungkan pemikiran pengambilan keputusan (Turban, Sharda & Delen, 2010).

Dari definisi ahli diatas, dapat disimpulkan bahwa sistem pendukung keputusan merupakan sistem informasi yang mendukung manajemen level menengah dalam mengambil keputusan semi terstruktur dengan menggunakan pemodelan analitis dan data yang ada.

## 2.2.2 Karakteristik dan Kapabilitas Sistem Pendukung Keputusan

Karakteristik dan Kapabilitas SPK menurut Turban, Sharda & Delen, adalah sebagai berikut :



**Gambar 2.1 Karakteristik dan Kapabilitas SPK (Turban, Sharda & Delen, 2010)**

1. SPK menyediakan dukungan bagi pengambil keputusan terutama pada situasi terstruktur dan tidak terstruktur dengan memadukan pertimbangan manusia dan informasi terkomputerisasi.
2. Dukungan untuk semua level manajerial, mulai dari eksekutif puncak sampai manajer lapangan.
3. Dukungan untuk individu dan kelompok. Masalah yang kurang terstruktur sering memerlukan keterlibatan individu dari departemen dan tingkat organisasional yang berbeda atau bahkan dari organisasi lain.

4. Dukungan untuk keputusan independen dan atau sekuensial. Keputusan dapat dibuat satu kali, beberapa kali atau berulang (dalam interval yang sama).
5. Dukungan pada semua fase proses pengambilan keputusan : intelegensi, desain, pilihan dan implementasi.
6. Dukungan diberbagai proses dan gaya pengambilan keputusan.
7. SPK selalu dapat beradaptasi sepanjang waktu. Pengambilan keputusan harus reaktif, dapat menghadapi perubahan kondisi secara tepat dan dapat mengadaptasikan SPK untuk memenuhi perubahan tersebut.
8. SPK mudah untuk digunakan. Pengguna harus merasa nyaman dengan sistem. *User-friendly*, dukungan grafis yang baik dan antarmuka bahasa yang sesuai dengan bahasa manusia dapat meningkatkan efektivitas SPK.
9. Peningkatan terhadap efektivitas dari pengambilan keputusan (akurasi, *timeless*, kualitas) ketimbang pada efisiensinya (biaya membuat keputusan, termasuk biaya penggunaan komputer).
10. Pengambil keputusan memiliki kontrol penuh terhadap semua langkah proses pengambilan keputusan dalam memecahkan suatu masalah. SPK ditujukan untuk mendukung bukan menggantikan pengambil keputusan.
11. Pengguna akhir dapat mengembangkan dan memodifikasi sistem sendiri. Sistem yang lebih besar dapat dibangun dengan bantuan ahli sistem informasi. Perangkat lunak OLAP dalam kaitannya dengan data *warehouse* membolehkan pengguna untuk membangun SPK yang cukup besar dan kompleks.

12. Biasanya model-model digunakan untuk menganalisa situasi pengambilan keputusan.
13. Akses disediakan untuk berbagai sumber data, format dan tipe mulai dari sistem informasi geografis (GIS) sampai sistem berorientasi objek.
14. Dapat dilakukan sebagai *stand-alone tool* yang digunakan oleh seorang pengambil keputusan pada satu lokasi atau didistribusikan pada suatu organisasi keseluruhan dan beberapa organisasi terkait.

### 2.2.3 Tahap-tahap Pembuatan Keputusan

Jika mengamati proses keluarnya suatu keputusan, maka dapat dicatat tahap-tahap penting dalam proses tersebut, yaitu :

1. Pemaparan masalah. Pada tahap ini, pengambil keputusan akan mendengarkan dan memperhatikan dengan seksama pemaparan persoalan yang dihadapi oleh perusahaannya.
2. Pengumpulan data dan informasi. Pada tahap ini, pengambil keputusan membutuhkan data dan informasi selengkap mungkin.
3. Pengamatan terhadap lingkungan. Pada tahap ini, pengambil keputusan akan mengamati keadaan lingkungan yang akan menerima keputusan yang akan di ambilnya.
4. Perancangan keputusan. Pada tahap ini, pengambil keputusan mencari dan menemukan alternatif-alternatif keputusan, mengembangkan dan menganalisa dampak masing-masing alternatif.



5. Penentuan keputusan. Pada tahap ini, pengambil keputusan akan melakukan seleksi terhadap alternatif-alternatif keputusan dengan hasil analisisnya masing-masing (Hutagalung, 2017).

#### **2.2.4 Manfaat Sistem Pendukung Keputusan**

1. Sistem Pendukung Keputusan memperluas kemampuan pengambil keputusan dalam memproses data/informasi bagi pemakainya.
2. Sistem Pendukung Keputusan membantu pengambil keputusan untuk memecahkan masalah terutama berbagai masalah yang sangat kompleks dan tidak terstruktur.
3. Sistem Pendukung Keputusan dapat menghasilkan solusi dengan lebih cepat serta hasilnya dapat diandalkan.
4. Walaupun suatu Sistem Pendukung Keputusan, mungkin saja tidak mampu memecahkan masalah yang dihadapi oleh pengambil keputusan, namun Sistem Pendukung Keputusan dapat menjadi stimulan bagi pengambil keputusan dalam memahami persoalannya, karena mampu menyajikan berbagai alternatif pemecahan (Utami, 2012).

#### **2.3 Simple Additive Weighting Method (SAW)**

Menurut Nofriansyah : mengemukakan bahwa “Metode *Simple Additive Weighting* sering dikenal dengan istilah metode penjumlahan terbobot” Nofriansyah (2014).

Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut. Metode SAW

membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat dibandingkan dengan semua rating alternatif yang ada.

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\text{Maxi } x_{ij}} & , \text{ Jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{x_{ij}}{\text{Maxi } x_{ij}} & , \text{ Jika } j \text{ adalah atribut biaya (cost)} \end{cases}$$

Dimana  $r_{ij}$  = rating kinerja ternormalisasi dari alternative  $A_i$  pada atribut  $C_j$

$x_{ij}$  = nilai yang diperoleh dari hasil *fuzzy*

$i = 1, 2, \dots, m$  dan

$j = 1, 2, \dots, n$ .

Nilai prefensi untuk setiap alternatif ( $v_i$ ) diberikan sebagai :

$$V_i = \sum_{j=1}^n w_j r_{ij}$$

Dimana  $v_i$  = hasil perkalian antara hasil *fuzzy* dan bobot

$w_j$  = bobot untuk setiap kriteria  $C_j$

$r_{ij}$  = rating kinerja ternormalisasi dari alternative  $A_i$  pada atribut  $C_j$

$i = 1, 2, \dots, m$  dan

$j = 1, 2, \dots, n$ .

Nilai  $v_i$  yang lebih besar mengindikasikan bahwa alternatif  $A_i$  lebih terpilih (Kusumadewi, 2006).

### 2.3.1 Langkah penyelesaian

Dalam penelitian ini menggunakan model FMADM (*Fuzzy Multiple Attribute Decision Making*) metode SAW. FMADM adalah suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu. Inti dari FMADM adalah menentukan nilai bobot untuk setiap

atribut, kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternatif yang sudah diberikan.

Pada dasarnya, ada beberapa pendekatan untuk mencari nilai bobot atribut, yaitu pendekatan subyektif, pendekatan obyektif dan pendekatan integrasi antara subyektif dan obyektif. Masing-masing pendekatan memiliki kelebihan dan kelemahan. Pada pendekatan subyektif, nilai bobot ditentukan berdasarkan subyektifitas dari para pengambil keputusan, sehingga beberapa faktor dalam proses perankingan alternatif bisa ditentukan secara bebas. Sedangkan pada pendekatan obyektif, nilai bobot dihitung secara matematis sehingga mengabaikan subyektifitas dari pengambil keputusan (Kusumadewi, 2006).

Adapun langkah-langkahnya adalah:

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu  $C_i$ .
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan kriteria  $C_i$ , kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi  $R$ .
4. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi  $R$  dengan vektor bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif ( $A_i$ ) sebagai solusi (Kusumadewi, 2006).

### 2.3.2 Kelebihan Metode SAW

Kelebihan dari metode *Simple Additive Weighting* dibanding dengan model pengambilan keputusan lainnya terletak pada kemampuannya untuk melakukan penilaian secara lebih tepat karena didasarkan pada nilai kriteria dan bobot preferensi yang sudah ditentukan, selain itu *Simple Additive Weighting* juga dapat menyeleksi alternatif terbaik dari sejumlah alternatif yang ada karena adanya proses perankingan setelah menentukan bobot untuk setiap atribut (Kusumadewi, 2006).

### 2.4 Bahasa Pemrograman .NET

Secara garis besar, ada 4 bahasa pemrograman yang paling sering digunakan oleh *programmer-programmer* di seluruh dunia yang di dukung oleh teknologi *.NET Framework*, yaitu :

1. *Visual Basic .NET*

Bahasa *Visual Basic .NET* merupakan bahasa generasi terbaru dari bahasa *Visual Basic* yang diciptakan oleh Microsoft. *Visual Basic .NET* telah murni menetapkan konsep *Object Oriented Programming* didalamnya. Sedangkan *Visual Basic Classic* belum sepenuhnya mendukung *Object Oriented Programming*. Bahasa *Visual Basic .NET* banyak digunakan para Programmer yang sebelumnya menggunakan bahasa *Visual Basic Classic*.

2. *Visual C# .NET*

Bahasa C# merupakan bahasa pemrograman yang diciptakan oleh Microsoft. Bahasa ini memiliki sintaks yang sangat mirip dengan bahasa C

atau *Java*. Sehingga para *programmer* yang memiliki latar belakang pemrograman *Java* atau *C*, lebih memilih menggunakan bahasa *C#* dibandingkan dengan bahasa *Visual Basic .NET*. *C#* merupakan bahasa yang simpel, elegan, dan mendukung penuh konsep *Object Oriented*, bahasa *C#* menduduki posisi ke-6 di dunia.

### 3. *Visual C++ .NET*

Bahasa *C++ .NET* merupakan bahasa generasi berikutnya dari bahasa *C++*. Bahasa *C++ .NET* dapat menggunakan semua fasilitas yang ada di dalam *.NET Framework*. Menurut [msdn.microsoft.com](http://msdn.microsoft.com), bahasa *C++ .NET* merupakan bahasa yang terbaik untuk menciptakan aplikasi yang memiliki performa tinggi untuk sistem operasi *Windows* dan juga untuk aplikasi *Web*.

#### 2.4.1 *Visual basic .NET*

Menurut Alexander F.K. Sibero, *visual basic .NET* adalah bahasa pemrograman yang dikembangkan oleh perusahaan Microsoft. *Visual basic .NET* merupakan pengembangan dari versi sebelumnya, yaitu *visual basic 6.0*. Perbedaan mendasar antara *visual basic .NET* dengan versi-versi sebelumnya adalah kemampuan OOP (*Object Oriented Programming*) yang telah ditanamkan pada *visual basic .NET* (Sibero, 2010).

Menurut Didik Dwi Prasetyo dalam (Santoso, 2017) dalam bukunya “Pemrograman Aplikasi *Database* dengan *Visual Basic .Net* dan *MS Access*” menuliskan bahwa : *Visual Basic .Net* merupakan salah satu bahasa pemrograman yang bisa digunakan untuk membangun aplikasi-aplikasi *.Net* di *platform*

*Microsoft .Net* (Prasetyo, 2005).

## **2.5 XAMPP**

### **2.5.1 Pengertian XAMPP**

Menurut Nugroho, “XAMPP adalah paket program web lengkap yang dapat Anda pakai untuk belajar pemrograman web, khususnya *PHP* dan *MySQL*” (Nugroho, 2013). Menurut Buana, “XAMPP adalah perangkat lunak *open source* yang diunggah secara gratis dan bisa dijalankan di semua semua operasi seperti *windows, linux, solaris, dan mac*” (Buana, 2014).

## **2.6 Basis Data**

Menurut Indrajani : “Basis data adalah kumpulan data yang saling berhubungan secara logis dan didesain untuk mendapatkan data yang dibutuhkan oleh suatu organisasi” (Indrajani, 2015).

Basis data merupakan kumpulan informasi yang disimpan didalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil *query* basis data disebut sistem manajemen basis data (DBMS).

Istilah “basis data” berawal dari ilmu komputer, meskipun kemudian artinya semakin luas dengan memasukkan hal-hal diluar bidang elektronika. Konsep dasar dari basis data adalah kumpulan dari catatan-catatan atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan yang disebut dengan *skema*. *Skema* menggambarkan objek yang diwakili suatu basis data, dan hubungan diantara objek tersebut.

Ada banyak cara untuk mengorganisasikan *skema* atau memodelkan struktur basis data yang biasa disebut dengan model basis data atau model rasional, yang menurut istilah layman mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom. Dalam model ini, hubungan antar tabel mewakili dengan nilai yang sama antar tabel.

Basis data *MySQL* merupakan sebuah *database server* yang *free*, artinya bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya. *MySQL* pertama kali dirintis oleh seorang *programmer database* bernama Michael Widenius. Kelebihan dari *database MySQL* selain berperan sebagai *database server* juga dapat berperan sebagai *client* sehingga sering disebut *database client/server* dengan kemampuan yang dapat berjalan baik di OS (*Operating Sistem*) manapun, dengan *platform Windows* maupun *Linux*. Selain itu *database* ini memiliki beberapa kelebihan dibanding *database* lain, diantaranya adalah :

- a. *MySQL* sebagai *Database Management System (DBMS)*
- b. *MySQL* sebagai *Relation Database Management System (RDBMS)*
- c. *MySQL* adalah sebuah *software database* yang *open source*, artinya program ini bersifat *free* atau bebas digunakan oleh siapa saja tanpa harus membeli dan membayar lisensi kepada pembuatnya (Nugroho, 2009).

## **2.7 Flowchart**

Menurut Indrajani, *Flowchart* merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program,. Biasanya mempengaruhi

penyelesaian masalah yang khususnya perlu dipelajari dan dievaluasi lebih lanjut (Indrajani, 2011).

*Flowchart* di bedakan menjadi 5 jenis *flowchart*, antara lain *system flowchart*, *document flowchart*, *schematic flowchart*, *program flowchart*, *process flowchart*. Masing-masing jenis *flowchart* akan dijelaskan berikut ini :

a. *System Flowchart*

*System Flowchart* dapat didefinisikan sebagai bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan-urutan dari prosedur-prosedur yang ada di dalam sistem. Bagan alir sistem menunjukkan apa yang dikerjakan di sistem.

b. *Document Flowchart*

Bagan alir dokumen (*document flowchart*) atau disebut juga bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya.

c. *Schematic Flowchart*

Bagan alir skematik (*schematic flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur didalam sistem. Perbedaanya adalah, bagan alir skematik selain menggunakan symbol-simbol bagan alir sistem, juga menggunakan gambar-gambar computer dan peralatan lainnya yang digunakan. Maksud penggunaa gambar-gambar ini adalah untuk memudahkan komunikasi kepada orang yang kurang paham dengan smbol-simbol bagan alir.



Penggunaan gambar-gambar ini memudahkan untuk dipahami, tetapi sulit dan lama menggambarinya.

d. *Program Flowchart*

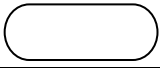
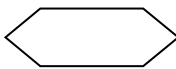
Bagan alir program (*program flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dibuat dari derivikasi bagan alir sistem.

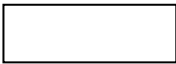

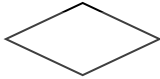



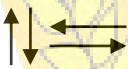
Bagan alir program dapat terdiri dari dua macam, yaitu bagan alir logika program (*program logic flowhart*) dan bagan alir program computer terinci (*detailed computer program flowchart*). Bagan alir logika program digunakan untuk menggambarkan tiap-tiap langkah di dalam program computer secara logika. Bagan alat logika program ini dipersiapkan oleh analis sistem. Gambar berikut menunjukkan bagan alir logika program. Bagan alir program computer terinci (*detailed computer program flowchart*) digunakan untuk menggambarkan instruksi-instruksi program computer secara terinci. Bagan alir ini dipersiapkan oleh pemogram.

e. *Process Flowchart*

Bagan alir proses (*process flowchart*) merupakan bagan alir yang banyak digunakan di teknik industry. Bagan alir ini juga berguna bagi analis sistem untuk menggambarkan proses dalam suatu prosedur.

**Tabel 2.1. Simbol-Simbol Flowchart**

SIMBOL	KETERANGAN
	<b>Terminal Point Symbol.</b> digunakan untuk menunjukkan awal dan akhir dari suatu proses.
	<b>Preparation Symbol. Simbol Persiapan</b> digunakan untuk memberi nilai awal Suatu besaran atau <i>variabel</i> ( harga awal ).

	<b>Process Symbol.</b> Simbol <b>Proses</b> atau <b>Pengolahan</b> digunakan untuk mewakili suatu proses, seperti pengolahan aritmatika atau pemindahan data.
	<b>Predefined Process Symbol.</b> Simbol <b>Proses Terdefenisi</b> digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain atau untuk proses yang detilnya dijelaskan terpisah, misalnya berbentuk subroutine.
	<b>Decision Symbol.</b> Simbol <b>Keputusan</b> digunakan untuk mewakili operasi perbandingan logika atau suatu penyeleksian kondisi di dalam program.
	<b>Input/Output Symbol.</b> Simbol <b>Input/Output</b> digunakan untuk menyatakan dan mewakili data masukan atau keluaran.
	<b>Connector Symbol.</b> Simbol <b>Penghubung</b> digunakan untuk menunjukkan sambungan dari bagan alir yang terputus di halaman yang masih sama.
	<b>Off-page Connector.</b> Simbol <b>Penghubung Halaman lain</b> digunakan untuk menunjukkan hubungan arus proses yang terputus dengan sambungannya ada di halaman yang lain.
	<b>Flow Lines Symbol.</b> Simbol <b>Garis Alir</b> digunakan untuk menunjukkan aliran atau arus dari proses.

## 2.8 UML

Pada tahun 1994, Grady Booch dan James Rumbaugh sepakat bergabung untuk menggunakan metode pengembangan berorientasi objek dan tujuan membuat proses standart tunggal untuk mengembangkan sistem berorientasi objek. Ivar Jacobson bergabung pada tahun 1995, dan mereka bertiga fokus membuat sebuah bahasa pemodelan objek standart sebagai ganti dari pendekatan atau metode berorientasi objek standart (pada saat menuliskan, mereka bertiga memasarkan metodologi pemodelan objek yang disebut *Unified Process*).

Berdasar kerja mereka dan hasil kerja lainnya pada industry, *Unified Modelling Process* (UML) versi 1.0 dirilis pada tahun 1997.

UML tidak menentukan metode untuk sistem-sistem pengembangan, hanya catatan yang saat ini telah diterima luas sebagai standart untuk pemodelan objek *Object Management Group* (OMG), badan standart industry, mengadopsi UML pada bulan November 1997 dan terus bekerja meningkatkannya berdasarkan kebutuhan industry.

UML (*Unified Modelling Language*) merupakan kosakata umum berbasis objek dan diagram teknik yang cukup efektif untuk memodelkan setiap proyek pengembangan sistem mulai tahap analisis sampai tahap desain dan implementasi (Dennis, 2012). UML dipublikasikan untuk maksud tertentu, biasanya antara lain untuk:

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

UML digunakan untuk menggambarkan perancangan awal dari sistem yang akan dibangun. UML memiliki banyak jenis permodelan, tetapi hanya beberapa yang digunakan saja yang akan dibahas.

Menurut Nugroho, “UML (*Unified Modelling Language*) adalah Metodologi kolaborasi antara metoda-metoda Booch, OMT (*Object Modelling Technique*), serta OOSE (*Object Oriented Software Engineering*) dan beberapa

metode lainnya, merupakan metodologi yang paling sering digunakan saat ini untuk analisa dan perancangan sistem dengan metodologi berorientasi objek mengadaptasi maraknya penggunaan bahasa pemrograman berorientasi objek” (Nugroho, 2009).

Berdasarkan pendapat yang dikemukakan diatas dapat ditarik kesimpulan bahwa “*Unified Modelling Language (UML)* adalah sebuah bahasa pemodelan secara visual membantu untuk menangkap struktur dan kelakuan objek dan mempermudah penggambaran interaksi untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah system pengembangan perangkat lunak berbasis objek (*Object Oriented Programming*)”.

### **2.8.1 Tujuan dari penggunaan UML**

Beberapa tujuan dari penggunaan UML diantaranya:




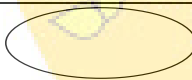
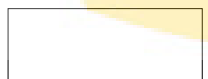
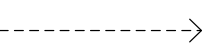
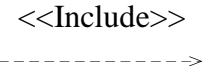
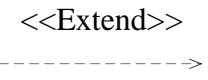
1. Memberikan bahasa pemodelan yang bebas dari berbagai bahas pemrograman dan proses rekayasa.
2. Menyatakan praktek-praktek terbaik yang terdapat dalam pemodelan.
3. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.

UML bisa juga berfungsi sebagai sebuah (*blue print*) cetak biru karena sangat lengkap dan detail. Dengan cetak biru ini maka akan bisa diketahui informasi secara detail tentang koding program atau bahkan membaca program dan menginterpretasikan kembali ke dalam bentuk diagram (*reverse engineering*).

### 2.8.2 Use Case Diagram

*Use case* diagram adalah rangkaian/uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. Berikut pengertian *Use case* diagram menurut Satzinger, “*Use Case* Diagram merupakan rangkaian tindakan yang dilakukan oleh sistem, aktor mewakili user atau sistem lain yang berinteraksi dengan sistem yang dimodelkan” (Satzinger, 2011).

**Tabel 2.2 Simbol Use Case Diagram**

Symbol	Penjelasan
	<i>Actor</i> Menspesifikasikan seperangkat peranan yang user sistem dapat diperankan ketika berinteraksi dengan use case.
	<i>Association</i> Menggambarkan interaksi antara actor dan use case
	<i>Generalization</i> Relasi antar use case, dimana salah satunya dalam bentuk yang lebih umum dari yang lain.
	<i>Use Case</i> Sebuah deskripsi dari seperangkat aksi-aksi berurutan yang ditampilkan pada sebuah sistem.
	<i>System</i> Tempat seluruh aktivitas-aktivitas sistem yang sedang berjalan.
	<i>Dependancy</i> Untuk menggambarkan ketergantungan sebuah use case dengan use case lainnya.
	<i>Include</i> Menggambarkan bahwa keseluruhan dari sebuah use case merupakan fungsionalitas use case lainnya.
	<i>Extend</i> Menggambarkan hubungan antara use case dimana sebuah use case merupakan fungsionalitas use case lainnya apabila kondisi tertentu dipenuhi.