

BAB II

TINJAUAN PUSTAKA

2.1 Kerangka Teori

2.2 2.1.1 Sistem

Istilah sistem terdiri dari dua bagian, yaitu struktur dan proses. Struktur adalah komponen dari sistem tersebut dan proses adalah prosedurnya. Kedua pendekatan tersebut hanya mengambil satu aspek dari sistem saja untuk menjelaskan dari sudut pandangan aspek tersebut. Untuk bisa mengetahui lebih dalam lagi mengenai sistem, berikut ini beberapa pendapat para ahli mengenai pengertian sistem, yaitu:

a. Pengertian sistem

1. Sistem adalah kumpulan dari komponen saling berhubungan satu sama lainnya membentuk satu kesatuan untuk mencapai satu tujuan. (Jogiyanto, 2003 : 34)
2. Sistem adalah sekelompok elemen – elemen yang berintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. (Raymond Mcleod, 2001 : 9)

Dari pengertian diatas penulis dapat mengambil kesimpulan bahwa sistem adalah kumpulan dari bagian-bagian, unsur-unsur atau komponen yang saling berhubungan satu sama lainnya secara teratur dan merupakan kesatuan yang saling berhubungan untuk mencapai tujuan.

Sistem mencakup 3 (tiga) bagian kegiatan yaitu:

1. Memasukkan (input) yaitu bagian yang dimasukkan ke dalam sistem.
2. Pengolahan (proses) yaitu pengolahan yang akan mengubah masukan menjadi keluaran.
3. Keluaran (output) yaitu hasil yang di olah dan diklasifikasikan menjadi keluaran

b. Karakteristik sistem

Model umum sebuah sistem terdiri dari input, proses dan output. Hal ini merupakan sebuah konsep yang sangat sederhana mengingat sebuah sistem dapat mempunyai beberapa masukan dan keluaran sekaligus. Suatu sistem juga mempunyai karakteristik atau sifat-sifat, yaitu mempunyai:

1. Komponen Sistem (*Components*)

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

2. Batas Sistem (*Boundary*)

Merupakan daerah yang membatasi antara sistem dengan sistem lainnya itu dengan lingkungan luar. Batas sistem menunjukkan ruang lingkup dari sistem tersebut.

3. Lingkungan Luar Sistem (*Environnement*)

Lingkungan luar dari sistem adalah batas dari sistem yang mempengaruhi operasi sistem, lingkungan luar sistem ini dapat menguntungkan dapat juga merugikan sistem tersebut.

4. Penghubung sistem (*Interface*)

Merupakan suatu media penhubung antara satu subsistem dengan subsistem yang lain. Dengan melalui penghubung ini memungkinkan sumber data mengalir dari suatu subsistem ke subsistem yang lain dengan melalui penghubung.

5. Masukan sistem (*Input*)

Masukan disini dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). Perawatan yang dimaksud adalah energy yang dimasukkan supaya sistem tersebut dapat ber operasi, dan masukan sinyal adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh didalam sistem komputer, program adalah masukan perawatan (*maintenance input*) dan data adalah masukan sinyal (*signal input*) untuk dioalah menjadi informasi.

6. Keluaran sistem (*output*)

Adalah hasil dari data yang diolah dan diklasifikasikan menjadi keluaran yang berguna atau informasi

7. Pengoalah sistem (*proses*)

Adalah suatu bagian dari sistem yang berfungsi merubah masukan menjadi keluaran.

8. Sasaran sistem (*Objective*)

Suatu sistem harus mempunyai tujuan dan sasaran, karena kalau tidak maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan.

2.1.2 Informasi

Untuk memahami defenisi dari sebuah sistem informasi secara lengkap, sebaiknya kita harus mendefenisikan terlebih dahulu arti dari Sistem dan Informasi. Hal ini dikarenakan bahwa keduanya memiliki defenisi yang berbeda tapi juga memiliki satu tujuan yang sama. Sistem dapat didefenisikan sebagai kumpulan atau group dari subsistem/bagian/komponen apapun baik fisik ataupun non fisik yang saling berhubungan satu sama lain dan bekerja sama harmonis untuk mencapai satu tujuan. Sedangkan informasi sebagai hasil pengolahan data yang berarti dan bermanfaat. Sehingga dapat ditarik kesimpulan bahwa sistem informasi adalah kumpulan/goup dari sub sistem/ bagian/ komponen baik fisik atau non fisik yang saling berhubungan satu sama lain dan bekerja sama untuk mencapai hasil yaitu mengolah data menjadi informasi yang memiliki arti dan berguna.

Pengertian menurut Kadir (2014:9), Sistem informasi adalah “sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi, dan didistribusikan kepada pemakai”.

2.1.3 Pengertian Sistem Informasi

Sistem informasi adalah sistem yang diciptakan oleh para analisis dan manajer guna melaksanakan tugas khusus tertentu yang sangat esensial bagi organisasi (Scott, 2004). Menurut Robert A. Leitch, sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Jogiyanto, 2005).

Jadi, sistem informasi dapat diartikan sebagai sebuah sistem yang terintegrasi secara optimal dan berbasis komputer yang dapat menghimpun dan menyajikan berbagai jenis data yang akurat untuk berbagai macam kebutuhan.

2.1.4 Pengertian Sistem Pendukung Keputusan (SPK)

Sistem pendukung keputusan ialah proses pengambilan keputusan dibantu menggunakan komputer untuk membantu pengambil keputusan dengan menggunakan beberapa data dan model tertentu untuk menyelesaikan beberapa masalah yang tidak terstruktur. Keberadaan SPK pada perusahaan atau organisasi bukan untuk menggantikan tugas-tugas pengambil keputusan, tetapi merupakan sarana yang membantu bagi mereka dalam pengambilan keputusan. Dengan menggunakan data-data yang diolah menjadi informasi untuk mengambil keputusan dari masalah-masalah semi-terstruktur. Dalam implementasi SPK, hasil dari keputusan-keputusan dari sistem bukanlah hal yang menjadi patokan, pengambilan keputusan tetap berada pada pengambil

keputusan. Sistem hanya menghasilkan keluaran yang mengkalkulasi data-data sebagaimana pertimbangan seorang pengambil keputusan. Sehingga kerja pengambil keputusan dalam mempertimbangkan keputusan dapat dimudahkan (Wibowo, 2011). Sistem Pendukung Keputusan dirancang untuk mendukung seluruh tahap pengambilan keputusan mulai dari mengidentifikasi masalah, memilih data yang relevan, dan menentukan pendekatan yang digunakan dalam proses pengambilan keputusan sampai mengevaluasi pemilihan alternatif-alternatif yang ada (Fitriani, 2012).

2.1.5 Karakteristik dan Kemampuan Sistem Pendukung Keputusan

Karakteristik sistem pendukung keputusan menurut Wibowo (Wibowo, 2011):

1. Sistem Pendukung Keputusan dirancang untuk membantu pengambil keputusan dalam memecahkan masalah yang sifatnya semi terstruktur ataupun tidak terstruktur dengan menambahkan kebijaksanaan manusia dan informasi komputerisasi.
2. Dalam proses pengolahannya, sistem pendukung keputusan mengkombinasikan penggunaan model-model analisis dengan teknik pemasukan data konvensional serta fungsi-fungsi pencari/interogasi informasi.
3. Sistem Pendukung Keputusan, dirancang sedemikian rupa sehingga dapat digunakan/dioperasikan dengan mudah.
4. Sistem Pendukung Keputusan dirancang dengan menekankan pada aspek fleksibilitas serta kemampuan adaptasi yang tinggi.

2.1.6 Manfaat dan Keuntungan yang dapat diambil dari SPK

Dengan berbagai karakter khusus di atas, SPK dapat memberikan berbagai manfaat dan keuntungan. Manfaat yang dapat diambil dari SPK menurut Kadarsah dalam tulisan Utami (Utami, 2012) :

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

2.1.7 Metode Weighted Product (WP)

Weighted Product (WP) adalah keputusan analisis multi-kriteria dan merupakan metode pengambilan keputusan multi kriteria. Seperti semua metode-metode lainnya, WP adalah himpunan dari alternatif keputusan yang dijelaskan dalam istilah beberapa kriteria.

Metode perkalian atau metode WP ini berbeda dengan metode SAW dalam perlakuan awal terhadap hasil penilaian atribut keputusan. Dalam metode WP tidak diperlukan manipulasi matriks karena metode ini

mengalikan hasil penilaian setiap atribut . Hasil perkalian tersebut belum dibandingkan (dibagi) dengan nilai standar, dalam hal ini alternatif ideal sering digunakan sebagai nilai standar bobot, bobot untuk atribut manfaat berfungsi sebagai pangkat positif dalam proses perkalian antar atribut, sementara bobot berfungsi sebagai pangkat negatif.(Fahmi Basyaib,2006)

Langkah-langkah dalam menggunakan metode weighted product adalah:

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan.
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Menentukan bobot preferensi tiap kriteria.
4. Mengalihkan seluruh atribut bagi sebuah alternatif dengan bobot sebagai pangkat positif untuk atribut keuntungan dan bobot berpangkat negatif untuk atribut biaya.
5. Hasil perkalian tersebut dijumlahkan untuk menghasilkan nilai S untuk setiap alternatif.
6. Mencari nilai alternatif dengan melakukan langkah yang sama seperti pada satu, hanya saja menggunakan nilai tertinggi untuk setiap atribut tertinggi untuk setiap atribut manfaat dan nilai terendah untuk atribut biaya.
7. Membagi nilai S bagi setiap alternatif dengan nilai total S yang akan menghasilkan V.
8. Mencari nilai alternatif ideal, setelah metode yang digunakan sudah ditentukan, tahap selanjutnya adalah menentukan tools yang akan digunakan, dimana sistem pendukung keputusan ini dirancang sebagai web

desktop application dengan tools yang digunakan yaitu PHP, CSS dan MySQL.

Metode Weighted Product menggunakan perkalian untuk menghubungkan rating atribut, dimana rating setiap atribut harus dipangkatkan terlebih dahulu dengan bobot atribut yang bersangkutan. Proses ini sama halnya dengan proses normalisasi. (Kusumadewi, 2006) Pembobotan metode WP dihitung berdasarkan tingkat Nilai Preferensi.

Proses normalisasi bobot kriteria (W), $\sum W = 1$ adalah :

$$W_j = W_j / \sum W_j \dots\dots\dots(1)$$

Keterangan :

W_j : Bobot atribut

$\sum W_j$: Penjumlahan bobot atribut

Menghitung Vektor S

$$S_i = \prod_{j=1}^n X_{ij}^{w_j}$$

Atau

$$S = S_1^{w_j} * S_2^{w_j} * \dots * S_i$$

Keterangan :

S : menyatakan alternatif yang dianalogikan sebagai vektor S

x : menyatakan nilai kriteria

w : menyatakan bobot kriteria

i : menyatakan alternatif

j : menyatakan kriteria

n : menyatakan banyaknya kriteria

W_j adalah pangkat bernilai positif untuk atribut keuntungan, dan bernilai negatif untuk atribut biaya.

Menghitung Vektor V:

$$V_i = \frac{\prod_{j=1}^n X_{ij}^{w_j}}{\prod_{j=1}^n (X_j^*)}$$

Atau

$$V_1 = \frac{S_1}{S_1 + S_2 + S_3}$$

Keterangan :

V : menyatakan alternatif yang dianalogikan sebagai vektor V

x : menyatakan nilai kriteria

w : menyatakan bobot kriteria

i : menyatakan alternatif

j : menyatakan kriteria

n : menyatakan banyaknya kriteria

Dimana, v_1 preferensi alternatif ke – I dan merupakan perjumlahan hasil perkalian rating alternatif per atribut.

2.1.8 Penjelasan UML

UML merupakan singkatan dari “*Unified Modeling Language*” yaitu suatu metode permodelan secara visual untuk sarana perancangan sistem berorientasi objek, atau defenisi UML yaitu sebagai suatu bahasa yang sudah menjadi standar pada viusalisasi, perancangan dan juga pendokumentasian

sistem software. Saat ini UML sudah menjadi bahasa standar dalam penulisan blue print software.

Pada tahun 1994, Grady Booch dan James Rumbaugh sepakat bergabung untuk menggunakan metode pengembangan berorientasi objek dengan tujuan membuat proses standar tunggal untuk mengembangkan sistem berorientasi objek. Ivar Jacobson bergabung pada tahun 1995, dan mereka bertiga fokus membuat sebuah bahasa pemodelan objek standar sebagai ganti pendekatan atau metode berorientasi objek standar (pada saat menuliskan, mereka bertiga memasarkan metodologi pemodelan objek yang disebut *UnifiedProcess*). Berdasarkan kerja mereka dan hasil kerja lainnya pada industry, *Unified Modeling Process (UML)* versi 1.0 dirilis pada tahun 1997.

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, 2012:513). 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 Booch (2005), “UML merupakan suatu bahasa. Suatu bahasa terdiri dari kata-kata, dan memiliki aturan untuk menggabungkan kata-kata tersebut, sehingga tercipta komunikasi. Sebuah permodelan bahasa adalah suatu bahasa dimana kata-kata dan aturannya berfokus pada penggambaran sistem secara konseptual dan fisik. Sebuah permodelan bahasa seperti UML telah menjadi bahasa standar untuk merencanakan suatu aplikasi”.

Menurut Nugroho (2009:4), “UML (Unified Modeling Language) adalah Metodologi kolaborasi antara metoda-metoda Booch, OMT (Object Modeling Technique), serta OOSE (object Oriented Software Engineering) dan beberapa metoda 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” (OOP)”.

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 divisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah system pengembangan perangkat lunak berbasis objek (Object Oriented Programming)”.

2.1.9 Tujuan dari penggunaan UML

Beberapa tujuan dari penggunaan UML diantaranya:









1. Memberikan bahasa pemodelan yang bebas dari berbagai bahas pemrograman dan proses rekayasa.
2. Menyatukan 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.
4. 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 coding program atau bahkan membaca program dan menginterpretasikan kembali ke dalam bentuk diagram (reserve engineering).

2.1.10 Jenis-Jenis diagram UML dan beberapa contoh diagramnya

a. 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 (2011 : 20) “*Use Case* Diagram merupakan rangkaian tindakan yang dilakukan oleh sistem, aktor mewakili user atau sistem lain yang berinteraksi dengan sistem yang dimodelkan”.


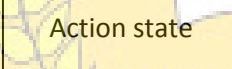

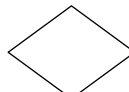
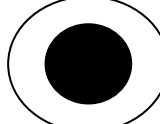
Tabel 2.1 Simbol Use Case Diagram

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

b. Activity Diagram

Diagram aktivitas adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. Diagram aktivitas mempunyai peran seperti halnya flowchart, akan tetapi perbedaannya dengan flowchart adalah diagram aktivitas bisa mendukung perilaku paralel sedangkan flowchart tidak bisa. Sebuah activity diagram memiliki : Satzinger (2011 : 23)

Tabel 2.2 Tipe Relasi pada Activity Diagram

Simbol	Penjelasan
	<i>Initial state</i> Mempresentasikan dimulainya alur kerja suatu sistem dalam <i>activity diagram</i> .
	<i>Action State</i> Sebuah <i>state</i> yang menggambarkan eksekusi dari aksi <i>atomic</i> .
	<i>Transition Between Activities</i> Mengidentifikasi bahwa suatu objek dari state pertama akan menampilkan aksi-aksi tertentu dan memasuki state kedua ketika peristiwa terjadi pergerakan dari aksi ke aksi lainnya.
	<i>Decision Point</i> Menentukan kapan alur dalam aktivitas menjadi bercabang.
	<i>Final State</i> Mempresentasikan bahwa telah diakhirinya alur suatu sistem dalam <i>activity diagram</i> .

c. Class Diagram

Diagram kelas atau Class diagram sangat membantu dalam visualisasi struktur kelas dari suatu sistem. Hal ini disebabkan karena class adalah deskripsi kelompok obyek-obyek dengan property, operasi dan relasi yang sama. Disamping itu diagram kelas bisa memberikan pandangan global atas sebuah sistem. Hal tersebut tercermin dari class-class yang ada dan relasinya satu dengan lainnya. Itulah sebabnya diagram kelas menjadi diagram yang paling populer di UML. Satzinger (2011 : 28)

1. Asosiasi adalah class-class yang berhubungan satu sama lain secara konseptual, yaitu menghubungkan dua kelas menjadi satu asosiasi.
2. Atribut adalah properti dari sebuah kelas. Atribut ini menjelaskan batas nilai yang mungkin ada pada obyek dari kelas. Sebuah kelas mungkin mempunyai nol atau lebih atribut.

Operasi adalah sesuatu yang bisa dilakukan oleh sebuah class atau yang anda (atau class yang lain) dapat lakukan untuk sebuah class. Sama halnya dengan atribut, kita bisa juga memberikan tambahan informasi untuk operasi dengan menambahkan parameter yang akan dilakukan oleh operasi dengan tanda kurung.

2.1.11 Visual Basic 2010

Visual Basic 2010 adalah salah satu bahasa pemrograman yang dikeluarkan dan dikembangkan oleh Microsoft. Metode pemrograman yang diterapkan dalam Visual Basic 2010 berorientasi kepada objek atau lebih

sering dikenal dengan istilah OOP (*Object Oriented Programming*) sehingga mempermudah pengembangan program.

Visual Basic 2010 merupakan program *event-driven*, artinya program menunggu pengguna melakukan sesuatu (“*event*”), seperti klik pada ikon, dan kemudian program akan merespons (“*driven*”). Karena penggunaannya mudah, Visual Basic memungkinkan programmer pemula untuk menciptakan aplikasi-aplikasi berbasis windows yang menarik.

2.1.12 Basis Data

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

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

Ada banyak cara untuk mengorganisasikan skema atau memodelkan struktur basisdata yang biasa disebut dengan model basisdata 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.

Basisdata 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 disbanding 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.

2.1.13 Flowchart

Pengertian *Flowchart* (Diagram Alur) Menurut Indrajani (2011:22), *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. *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 menggambaranya.


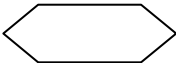






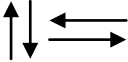
d. *Program Flowchart*

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 utnuk 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 2.3 Simbol-Simbol Flowchart

SIMBOL	KETERANGAN
	Terminal Point Symbol. digunakan untuk menunjukkan awal dan akhir dari suatu proses.
	Preparation Symbol. Simbol Persiapan digunakan untuk memberi nilai awal Suatu besaran atau <i>variabel</i> (harga awal).
	Process Symbol. Simbol Proses atau Pengolahan digunakan untuk mewakili suatu proses, seperti pengolahan aritmatika atau pemindahan data.
	Predefined Process Symbol. Simbol Proses Terdefinisi digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain atau untuk proses yang detilnya dijelaskan terpisah, misalnya berbentuk subroutine.
	Decision Symbol. Simbol Keputusan digunakan untuk mewakili operasi perbandingan logika atau suatu penyeleksian kondisi di dalam program.
	Input/Output Symbol. Simbol Input/Output digunakan untuk menyatakan dan mewakili data masukan atau keluaran.
	Connector Symbol. Simbol Penghubung digunakan untuk menunjukkan sambungan dari bagan alir yang terputus di halaman yang masih sama.
	Off-page Connector. Simbol Penghubung Halaman lain digunakan untuk menunjukkan hubungan arus proses yang terputus dengan sambungannya ada di halaman yang lain.
	Flow Lines Symbol. Simbol Garis Alir digunakan untuk menunjukkan aliran atau arus dari proses.

2.2 Kerangka Berpikir

Kerangka pemikiran dibuat berdasarkan penelitian yang dilakukan oleh penulis.

