

BAB II

LANDASAN TEORI

2.1 Sistem Pendukung Keputusan (SPK)

Menurut Alter dalam buku yang ditulis oleh Kusriani (2007), sistem pendukung keputusan atau biasa disebut *Decision Support System* (DSS) merupakan sistem informasi interaktif yang menyediakan informasi pemodelan dan pemanipulasian data. Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi yang terstruktur dan situasi yang tidak terstruktur, dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat.

Konsep sistem pendukung keputusan pertama kali diperkenalkan oleh Michael S. Scott Morton pada awal tahun 1970an dengan istilah *Management Decision System*. Sistem tersebut adalah sistem berbasis komputer yang ditujukan untuk membantu pengambilan keputusan dengan memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur (Kurniasih, 2013).

DSS biasanya digunakan untuk mendukung solusi atas suatu masalah atau untuk mengevaluasi suatu peluang. DSS yang seperti itu disebut aplikasi DSS. Aplikasi DSS digunakan dalam pengambilan keputusan dengan menggunakan *Computer Based Information Systems* (CBIS) yang fleksibel, interaktif, dan dapat diadaptasi,

yang dikembangkan untuk mendukung solusi atas masalah manajemen spesifik yang tidak terstruktur. Aplikasi DSS ini menggunakan data, memberikan antarmuka pengguna yang mudah, dan dapat menggabungkan pemikiran pengambil keputusan. DSS tidak dimaksudkan untuk mengotomatisasikan pengambil keputusan, tetapi memberikan perangkat interaktif yang memungkinkan pengambil keputusan untuk melakukan berbagai analisis menggunakan model-model yang tersedia (Kusrini, 2007).

Menurut Turban dalam buku Kusrini (2007) tujuan dari DSS adalah sebagai berikut:

1. Membantu manajer dalam pengambilan keputusan atas masalah yang terstruktur.
2. Memberikan dukungan atas pertimbangan manajer dan bukannya dimaksudkan untuk menggantikan fungsi manajer.
3. Meningkatkan efektivitas keputusan yang diambil manajer lebih daripada perbaikan efisiensinya.
4. Kecepatan komputasi. Komputer memungkinkan para pengambil keputusan untuk melakukan banyak komputasi secara cepat dengan biaya yang rendah.
5. Peningkatan produktivitas.
6. Dukungan kualitas. Komputer bisa meningkatkan kualitas keputusan yang dibuat.
7. Berdaya saing.
8. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan.

Menurut Kusrini dalam bukunya (Kusrini, 2007), DSS dibagi menjadi tiga ditinjau dari tingkat teknologinya, yaitu:

1. SPK Spesifik

SPK spesifik bertujuan membantu memecahkan suatu masalah dengan karakteristik tertentu. Misalnya, SPK penentuan harga satuan barang.

2. Pembangkit SPK

Suatu *software* yang khusus digunakan untuk membangun dan mengembangkan SPK. Pembangkit SPK akan memudahkan perancang dalam membangun SPK spesifik.

3. Perlengkapan SPK

Berupa *software* dan *hardware* yang digunakan untuk mendukung pembangunan SPK spesifik maupun pembangkit SPK.

Selanjutnya, Kusrini juga menjelaskan, berdasarkan keterstrukturannya, keputusan yang diambil untuk menyelesaikan suatu masalah dibagi menjadi tiga, yaitu (Kusrini, 2007):

1. Keputusan Terstruktur (*Structured Decision*)

Keputusan terstruktur adalah keputusan yang dilakukan secara berulang-ulang dan bersifat rutin. Keputusan ini biasanya dilakukan pada manajemen tingkat bawah.

2. Keputusan Semiterstruktur (*Semistructured Decision*)

Keputusan semiterstruktur adalah keputusan yang memiliki dua sifat. Sebagian keputusan bisa ditangani oleh komputer dan yang lain tetap harus

dilakukan oleh pengambil keputusan. Keputusan semacam ini biasanya dilakukan untuk manajemen tingkat menengah.

3. Keputusan tak Terstruktur (*Unstructured Decision*)

Keputusan tak terstruktur adalah keputusan yang penanganannya rumit karena tidak terjadi berulang-ulang atau tidak selalu terjadi. Keputusan tersebut menuntut pengalaman dan berbagai sumber yang bersifat eksternal.

Keputusan ini umumnya digunakan oleh manajemen tingkat atas.

Menurut Turban dalam buku Kusri (2007), karakteristik yang diharapkan ada di DSS, antara lain:

1. Dukungan kepada pengambil keputusan, terutama pada situasi semiterstruktur dan tak terstruktur, dengan menyertakan penilaian manusia dan informasi terkomputerisasi.
2. Dukungan untuk semua tingkat manajerial, dari eksekutif puncak sampai manajer lini.
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.
5. Dukungan di semua fase proses pengambilan keputusan inteligensi, desain, pilihan, dan implementasi.
6. Dukungan di berbagai proses dan gaya pengambilan keputusan.

7. Adaptivitas sepanjang waktu. Pengambil keputusan seharusnya reaktif, bisa menghadapi perubahan kondisi secara cepat, dan mengadaptasi DSS untuk memenuhi kebutuhan tersebut.
8. Peningkatan efektivitas pengambilan keputusan (akurasi, *timelines*, kualitas) ketimbang pada efisiensinya (biaya pengambilan keputusan).
9. Kontrol penuh oleh pengambil keputusan terhadap semua langkah proses pengambilan keputusan dalam memecahkan suatu masalah.
10. Pengguna akhir bisa mengembangkan dan memodifikasi sendiri sistem sederhana.
11. Dapat digunakan sebagai alat *stand alone* oleh seorang pengambil keputusan pada satu lokasi atau didistribusikan di suatu organisasi secara keseluruhan dan di beberapa organisasi sepanjang rantai persediaan.

2.2 *Analitycal Hierarchy Process (AHP)*

Metode *Analitycal Hierarchy Process (AHP)* merupakan teknik penyelesaian masalah dalam suatu pengambilan keputusan yang dapat membantu mengatasi masalah kerumitan, seperti beragamnya kriteria dari suatu permasalahan. AHP diperkenalkan oleh Thomas L. Saaty pada periode 1971-1975. AHP tidak hanya digunakan untuk menentukan prioritas pilihan-pilihan banyak kriteria, tetapi penerapannya telah meluas sebagai metode alternatif untuk menyelesaikan bermacam-macam masalah, seperti memilih portofolio, analisis manfaat biaya, penentuan prioritas, dan lain-lain (Mulyono, 1996).

Pada dasarnya AHP digunakan untuk menemukan skala rasio baik dari perbandingan pasangan yang diskrit maupun kontinu. Perbandingan tersebut diambil dari ukuran aktual atau dari suatu skala dasar yang mencerminkan kekuatan perasaan dan preferensi relatif. AHP memiliki perhatian khusus tentang penyimpangan dari konsistensi, pengukuran, dan pada ketergantungan di dalam dan di antara kelompok elemen strukturnya (Mulyono, 1996).

Kelebihan metode AHP dibandingkan dengan metode pengambilan keputusan yang lain terletak pada kemampuannya untuk memecahkan masalah yang multiobjektif dengan multikriteria. Kelebihan metode AHP ini lebih disebabkan oleh fleksibilitasnya yang tinggi terutama dalam pembuatan hirarki yang membuat metode AHP dapat menangkap beberapa tujuan dan beberapa kriteria sekaligus dalam sebuah model atau sebuah hirarki (Hidayat, 2014).

2.2.1 Dasar-Dasar AHP

Menurut Mulyono dalam bukunya (Mulyono, 1996), dalam menyelesaikan persoalan dengan AHP terdapat beberapa prinsip yang harus dipahami, yaitu:

1. *Decomposition*

Apabila persoalan telah didefinisikan, maka perlu dilakukan *decomposition* yaitu memecah persoalan yang utuh menjadi unsur-unsurnya. Agar hasilnya akurat, maka dilakukan pemecahan kembali dari unsur-unsurnya sampai tidak mungkin dilakukan pemecahan lebih lanjut. Proses ini dinamakan hirarki (*hierarchy*).

2. *Comperative Judgment*

Pada proses ini dibuat penilaian tentang kepentingan relatif dua elemen pada suatu tingkat tertentu dalam kaitannya dengan tingkat di atasnya. Hasil penilaian kepentingan relatif dua elemen yang dibandingkan kemudian dilakukan normalisasi. Penilaian ini merupakan inti dari AHP, karena akan berpengaruh terhadap prioritas elemen-elemen. Hasil dari penilaian disajikan dalam bentuk matriks *pairwise comparison*.

Agar diperoleh skala yang bermanfaat saat membandingkan dua elemen, maka harus dipahami secara menyeluruh terlebih dahulu tentang elemen-elemen yang dibandingkan dan relevansinya terhadap kriteria atau tujuan yang dipelajari. Dalam penyusunan skala kepentingan ini, digunakan patokan skala dasar yang ditampilkan dalam Tabel 2.1.

Tabel 2.1 Skala Kepentingan Relatif Thomas L. Saaty (Mulyono, 1996)

Tingkat Kepentingan	Definisi
1	Sama pentingnya dibanding yang lain
3	Sedikit pentingnya dibanding yang lain
5	Kuat pentingnya dibanding yang lain
7	Sangat kuat pentingnya dibanding yang lain
9	Mutlak pentingnya dibanding yang lain
2,4,6,8	Nilai di antara dua nilai pertimbangan yang berdekatan
<i>Reciprocal</i>	Jika elemen i memiliki salah satu angka di atas ketika dibandingkan dengan elemen j, maka j memiliki nilai kebalikannya ketika dibanding dengan elemen i.

Dalam penilaian kepentingan relatif dua elemen berlaku aksioma *reciprocal*, artinya jika elemen i dinilai 3 kali lebih penting dibanding j, maka elemen j harus sama dengan 1/3 kali pentingnya dibanding elemen i. jika terdapat n elemen, maka akan diperoleh matriks *pairwise comparison* berukuran $n \times n$.

Banyaknya penilaian yang diperlukan dalam menyusun matriks ini adalah $n(n - 1)/2$ karena matriksnya *reciprocal* dan elemen-elemen diagonal sama dengan 1.

3. *Synthesis of Priority*

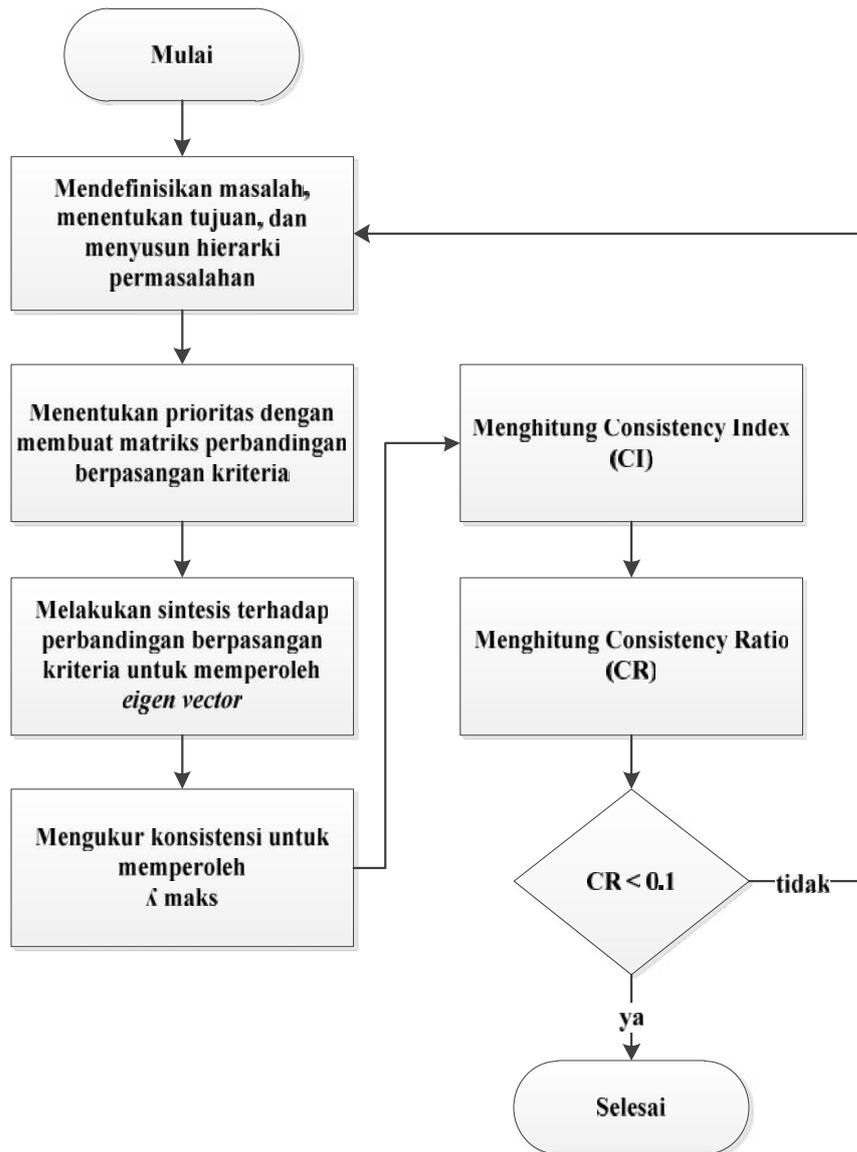
Dari setiap matriks *pairwise comparison* kemudian dicari *eigen vector*-nya. *Eigen vector* merupakan hasil perbandingan matriks berupa *local priority* atau nilai prioritas dari masing-masing elemen. Karena matriks *pairwise comparison* terdapat pada setiap tingkat, maka untuk mendapatkan *global priority* harus dilakukan sintesa di antara *local priority*. Prosedur melakukan sintesa berbeda menurut bentuk hirarki. Pengurutan elemen-elemen menurut kepentingan relatif melalui prosedur sintesa dinamakan *priority setting*.

4. *Logical Consistency*

Konsistensi memiliki dua makna, yaitu pertama, bahwa obyek-obyek yang serupa dapat dikelompokkan sesuai dengan keseragaman dan relevansi. Kedua, menyangkut tingkat hubungan antara obyek-obyek yang didasarkan pada kriteria tertentu. *Logical Consistency* dibagi menjadi dua, yaitu *Consistency Index* (CI) sebagai indikator kekonsistenan dan *Consistency Ratio* (CR) sebagai batas toleransi ketidak konsistenan dari nilai yang dimasukkan.

2.2.2 Langkah-Langkah Metode AHP

Menurut Kusriani (2007), prosedur atau langkah-langkah dalam metode AHP dapat dijelaskan dalam Gambar 2.1.



Gambar 2.1 Langkah-langkah Metode AHP (Kusrini, 2007)

Dalam memeriksa konsistensi hirarki, jika nilai *Consistency Ratio* (CR) lebih dari 10%, maka penilaian data *judgment* harus diperbaiki. Namun jika kurang atau sama dengan 0.1, maka hasil perhitungan bisa dinyatakan benar. Daftar *Consistency Random Index* (RI) dapat dilihat pada Tabel 2.2.

Tabel 2.2 Daftar Consistency Random Index (Kusrini, 2007)

Ukuran Matriks	Nilai RI
1.2	0.00
3	0.58
4	0.90
5	1.12
6	1.24
7	1.32
8	1.41
9	1.45
10	1.49
11	1.51
12	1.48
13	1.56
14	1.57
15	1.59

2.3 Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)

Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) adalah salah satu metode pengambilan keputusan yang pertama kali diperkenalkan oleh Yoon dan Hwang (1981). TOPSIS menggunakan prinsip bahwa alternatif yang terpilih harus mempunyai jarak terdekat dari solusi ideal positif dan mempunyai jarak terjauh dari solusi ideal negatif dari sudut pandang geometris dengan menggunakan jarak *Euclidean* untuk menentukan kedekatan relatif dari suatu alternatif dengan solusi optimal. Solusi ideal positif didefinisikan sebagai jumlah dari seluruh nilai terbaik yang dapat dicapai untuk setiap atribut, sedangkan solusi ideal negatif terdiri dari seluruh nilai terburuk yang dicapai untuk setiap atribut (Kurniasih, 2013).

TOPSIS telah menjadi teknik pengambilan keputusan yang menguntungkan untuk memecahkan masalah multikriteria. Hal ini dikarenakan dua alasan, yaitu (Singh, 2012):

1. Konsep dari TOPSIS masuk akal dan mudah dimengerti.
2. Dibandingkan metode pendukung keputusan lainnya, seperti AHP, TOPSIS hanya menggunakan sedikit proses komputasi, sehingga metode ini mudah untuk diterapkan.

Metode TOPSIS membantu menetapkan prioritas pada parameter yang harus dipertimbangkan dengan mengurangi keputusan yang kompleks menjadi serangkaian perbandingan *one-to-one*, sehingga diperoleh hasil yang terbaik dan ideal (Singh, 2012).

2.3.1 Langkah-langkah Metode TOPSIS

Menurut Sari (2013), langkah-langkah metode TOPSIS secara umum adalah sebagai berikut,

1. Membuat matriks keputusan yang ternormalisasi. Elemen r_{ij} merupakan hasil dari normalisasi *decision matrix* R dengan metode *Euclidian length of a vector*, dengan rumus sebagai berikut,

$$r_{ij} = \frac{X_{ij}}{\sqrt{\sum_{i=1}^n X_{ij}^2}}$$

dengan $i=1,2,\dots,m$; dan $j=1,2,\dots,n$

dimana :

r_{ij} = matriks ternormalisasi [i][j]

x_{ij} = matriks keputusan [i][j]

2. Membuat matriks keputusan yang ternormalisasi terbobot. Dengan bobot $W = (w_1, w_2, \dots, w_n)$, maka normalisasi bobot matrix V adalah:

$$V = \begin{bmatrix} w_{11}r_{11} & \cdots & w_{1n}r_{1n} \\ \vdots & \ddots & \vdots \\ w_{m1}r_{m1} & \cdots & w_{nm}r_{nm} \end{bmatrix}$$

3. Menentukan matriks solusi ideal positif dan matriks solusi ideal negatif. Solusi ideal positif A^+ dan solusi ideal negatif A^- dapat ditentukan berdasarkan rating bobot ternormalisasi (y_{ij}) sebagai :

$$y_{ij} = w_i \times r_{ij} ; \text{ dengan } i=1,2,\dots,m; \text{ dan } j=1,2,\dots,n$$

$$A^+ = (y_1^+, y_2^+, \dots, y_n^+);$$

$$A^- = (y_1^-, y_2^-, \dots, y_n^-);$$

dimana :

y_{ij} = matriks ternormalisasi terbobot [i][j]

w_i = vektor bobot[i] dari proses AHP

r_{ij} = matriks ternormalisasi [i][j]

4. Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif dan matriks solusi ideal negatif.

Jarak alternatif untuk solusi ideal positif adalah,

$$Di^+ = \sqrt{\sum_{i=1}^n (y_i^+ - y_{ij})^2} ; \quad i=1,2,\dots,m$$

dimana :

D_i^+ = jarak alternatif A_i dengan solusi ideal positif

y_i^+ = solusi ideal positif [i]

y_{ij} = matriks normalisasi terbobot [i][j]

Jarak alternatif untuk solusi ideal negatif adalah,

$$D_i^- = \sqrt{\sum_{j=1}^n (y_{ij} - y_i^-)^2} ; i=1,2,\dots,m$$

dimana :

D_i^- = jarak alternatif A_i dengan solusi ideal negatif

y_i^- = solusi ideal positif [i]

y_{ij} = matriks normalisasi terbobot [i][j]

5. Menentukan nilai preferensi untuk setiap alternatif.

$$V_i = \frac{D_i^-}{D_i^- + D_i^+} ; i = 1, 2, \dots, m$$

dimana :

V_i = kedekatan tiap alternatif terhadap solusi ideal

D_i^+ = jarak alternatif A_i dengan solusi ideal positif

D_i^- = jarak alternatif A_i dengan solusi ideal negatif

Nilai V_i yang lebih besar menunjukkan bahwa alternatif A_i lebih dipilih.

2.4 Metode Pengembangan Perangkat Lunak

2.4.1 Analisis Sistem

Dalam tahap analisis ini pengembang akan bertemu dengan pengguna untuk mengetahui kebutuhan awal yang diharapkan oleh pengguna. Kemudian

pengembang dengan tim akan mengidentifikasi kebutuhan sistem bagi pengguna tadi secara lebih rinci (Al Fatta, 2007).

Abdul Kadir Menjelaskan bahwa analisis sistem mencakup studi kelayakan dan analisis kebutuhan, dengan penjelasan sebagai berikut (Kadir, 2003):

- Studi kelayakan digunakan untuk menentukan kemungkinan keberhasilan solusi yang diusulkan. Analisis sistem melaksanakan penyelidikan awal terhadap masalah dan peluang bisnis yang disajikan dalam usulan proyek pengembangan sistem.
- Studi kebutuhan dilakukan untuk menghasilkan spesifikasi kebutuhan. Spesifikasi kebutuhan adalah spesifikasi yang rinci tentang hal-hal yang akan dilakukan sistem ketika diimplementasikan (Kadir, 2003).

2.4.2 Desain Sistem

Desain sistem merupakan tahap pembuatan atau perancangan desain sistem. Target akhir dari perancangan ini adalah menghasilkan rancangan yang memenuhi kebutuhan yang ditentukan selama tahapan analisis sistem. Hasil akhirnya berupa spesifikasi rancangan yang sangat rinci sehingga mudah diwujudkan pada saat pemrograman. Ada beberapa alat bantu yang digunakan dalam desain sistem yaitu DFD (*Data Flow Diagram*), Diagram Konteks (*Context Diagram*) dan ERD (*Entity Relationship Diagram*) (Kadir, 2003).

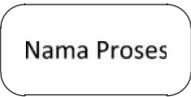
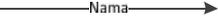
2.4.2.1 Context Diagram

Context diagram adalah DFD pertama dalam proses bisnis. Menunjukkan semua proses dalam satu proses tunggal. *Context diagram* juga menunjukkan semua entitas luar yang menerima informasi dari atau memberikan informasi ke sistem (Al Fatta, 2007).

2.4.2.2 Data Flow Diagram (DFD)

Ada dua jenis DFD, yaitu DFD logis dan DFD fisik. DFD logis menggambarkan proses tanpa menyoroti bagaimana mereka akan dilakukan, sedangkan DFD fisik menggambarkan proses model berikut implementasi pemrosesan informasinya (Al Fatta, 2007).

Tabel 2.3 Simbol Data Flow Diagram (Al Fatta, 2007)

Elemen Data Flow Diagram	Simbol Gene And Sarson	Simbol De Marco and Jourdan	Keterangan
Proses	 Nama Proses	 Nama Proses	Menunjukkan pemrosesan data/informasi yang terjadi di dalam sistem
Data Flow	 Nama	 Nama	Menunjukkan arah aliran dokumen antar bagian yang terkait pada suatu sistem
Data Store			Tempat menyimpan dokumen arsip
Entitas			Menunjukkan entitas atau bagian yang terlibat yang melakukan proses

2.4.2.3 Entity Relationship Diagram (ERD)

ERD adalah gambar atau diagram yang menunjukkan informasi dibuat, disimpan dan digunakan dalam sistem bisnis. Entitas biasanya menggambarkan jenis informasi yang sama. Dalam entitas digunakan untuk menghubungkan antar entitas yang sekaligus menunjukkan hubungan antar data (Al Fatta, 2007).

Tabel 2.4 Simbol ERD (Al Fatta, 2007)

Elemen ERD	Simbol	Keterangan
Entitas		Menunjukkan objek pada suatu sistem/menjelaskan entitas yang terlibat di dalamnya
<i>Relationship</i>		Menunjukkan hubungan antara dua entitas
Garis Penghubung		Menunjukkan aliran data
Atribut		Melambangkan atribut

2.4.2.4 Flowchart

Flowchart adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma. Terdapat dua macam *flowchart* yang menggambarkan proses dengan komputer, yaitu (Al-Bahra, 2005) :

1. System Flowchart

System flowchart merupakan bagan yang memperlihatkan urutan proses dalam sistem dengan menunjukkan alat media *input*, *output* serta jenis media penyimpanan dalam proses pengolahan data.

2. Program Flowchart

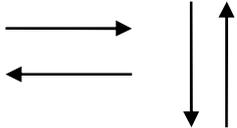
Program flowchart merupakan bagan yang memperlihatkan urutan instruksi yang digambarkan dengan simbol tertentu untuk memecahkan masalah dalam suatu program.

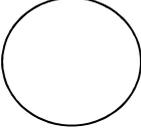
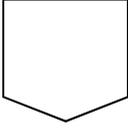
Flowchart disusun dengan simbol yang dipakai sebagai alat bantu menggambarkan proses di dalam program. Simbol-simbol yang digunakan dibagi menjadi tiga, yaitu (Al-Bahra, 2005):

1. Flow Direction Symbols (Simbol Penghubung/Alur)

Flow Direction Symbols merupakan simbol yang digunakan untuk menghubungkan antara simbol satu dengan simbol yang lain. Simbol ini disebut juga *connecting line*.

Tabel 2.5 Flow Direction Symbols (Al-Bahra, 2005)

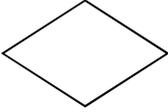
Simbol	Keterangan
	Simbol arus / <i>flow</i> , untuk menyatakan jalannya arus suatu proses.
	Simbol <i>communication link</i> , untuk menyatakan bahwa adanya transmisi suatu data/informasi dari satu lokasi ke lokasi lainnya.

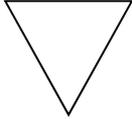
	<p>Simbol <i>connector</i>, untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman yang sama.</p>
	<p>Simbol <i>offline connector</i>, untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman yang berbeda.</p>

2. Processing Symbols (Simbol Proses)

Processing Symbols merupakan simbol yang menunjukkan jenis operasi pengolahan dalam suatu proses/prosedur.

Tabel 2.6 Processing Symbols (Al-Bahra, 2005)

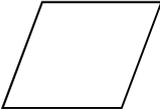
Simbol	Keterangan
	<p>Simbol <i>process</i>, untuk menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.</p>
	<p>Simbol <i>manual</i>, untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer (manual).</p>
	<p>Simbol <i>decision</i> / logika, untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, ya / tidak.</p>
	<p>Simbol <i>predefined process</i>, untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.</p>
	<p>Simbol <i>terminal</i>, untuk menyatakan permulaan atau akhir suatu program.</p>
	<p>Simbol <i>keying operation</i>, untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai <i>keyboard</i>.</p>

	Simbol <i>offline-storage</i> , untuk menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu.
	Simbol <i>manual input</i> , untuk memasukkan data secara manual dengan menggunakan <i>online keyboard</i> .

3. Input-Output Symbols (Simbol Input-Output)

Input-Output Symbols merupakan simbol yang menunjukkan jenis peralatan yang digunakan sebagai media *input* atau *output*.

Tabel 2.7 Input-Output Symbols (Al-Bahra, 2005)

Simbol	Keterangan
	Simbol <i>input/output</i> , untuk menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis peralatannya.
	Simbol <i>punched card</i> , untuk menyatakan <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu.
	Simbol <i>magnetic tape</i> , untuk menyatakan <i>input</i> berasal dari pita magnetis atau <i>output</i> disimpan ke pita magnetis.
	Simbol <i>disk storage</i> , untuk menyatakan <i>input</i> berasal dari disk atau <i>output</i> disimpan ke <i>disk</i> .
	Simbol <i>document</i> , untuk mencetak keluaran dalam bentuk dokumen (melalui printer).
	Simbol <i>display</i> , untuk mencetak keluaran dalam layar monitor.

2.4.3 Implementasi Sistem

Menurut Kadir dalam bukunya (Kadir, 2003), implementasi sistem dilakukan untuk memastikan apakah aplikasi yang dibuat telah sesuai kebutuhan sistem, desain, dan semua fungsi dapat berjalan dan dipergunakan dengan baik tanpa kesalahan atau eror. Pada implementasi sistem, aktifitas yang dilakukan antara lain:

- a. Pemrograman dan Pengujian
- b. Instalasi *hardware* dan *software*
- c. Pelatihan kepada pengguna
- d. Pembuatan dokumentasi
- e. Konversi

2.4.4 Pemeliharaan Sistem

Selama sistem beroperasi, pemeliharaan sistem tetap diperlukan karena beberapa alasan. Pertama, mungkin sistem masih menyisakan masalah-masalah yang tidak terdeteksi selama pengujian sistem. Kedua pemeliharaan diperlukan karena perubahan bisnis atau lingkungan atau adanya permintaan kebutuhan baru oleh pengguna. Ketiga, pemeliharaan juga bisa dipicu karena kinerja sistem yang menjadi menurun (Kadir, 2003).

2.4.5 Pengujian Sistem

Pengujian perangkat lunak merupakan proses eksekusi suatu program atau sistem dengan tujuan menemukan ada atau tidaknya kekurangan atau masalah pada sistem dengan melibatkan setiap aktivitas. Pada pengujian sistem ini dilakukan

evaluasi pada setiap atribut atau kemampuan suatu sistem sehingga diketahui apakah sistem yang dibuat sudah sesuai dengan kebutuhan pengguna (Zulkifli, 2013).

Black box Testing merupakan salah satu metode pengujian sistem, dimana pengujian yang dilakukan berdasarkan spesifikasi kebutuhan sistem dan tidak perlu memeriksa *coding*. Dengan *Black box Testing*, pengujian yang dilakukan hanya berdasarkan pandangan pengguna untuk mengetahui apakah fungsi yang dibutuhkan berjalan sesuai harapan atau tidak. Metode *Black box Testing* ini digunakan saat suatu sistem telah selesai dibuat. Keuntungan penggunaan metode ini adalah pengujian tidak memerlukan pengetahuan yang spesifik mengenai bahasa pemrograman yang digunakan dalam pembuatan sistem tersebut dan juga pengetahuan pada implementasinya (Nidhra dan Dondeti, 2012).

Pada pengujian dengan *Black box Testing*, ada beberapa teknik yang dapat digunakan, salah satunya adalah *Equivalence Class Partitioning*. Pengujian dengan *Equivalence Class Partitioning* didasarkan pada asumsi bahwa *input* dan *output* program dapat dibagi menjadi kelas dengan jumlah terbatas (valid dan non-valid) sehingga semua kasus yang sudah dipartisi ke dalam kelas-kelasnya akan diuji dengan perilaku yang sama (Nidhra dan Dondeti, 2012).