

## BAB II

### LANDASAN TEORI

#### 2.1 Studi Kepustakaan

Di penyusunan proposal ini penulis menggunakan beberapa acuan kepustakaan yang akan digunakan sebagai bahan pedoman dan referensi.

Penelitian yang dilakukan oleh (Frieyadi, 2016) mengenai penerapan metode *Simple Additive Weighting (SAW)* dalam sistem pendukung keputusan promosi kenaikan jabatan. Salah satu proses yang sangat penting dalam Departemen Sumber Daya Manusia (SDM) suatu perusahaan atau badan yang promosi promosi. Secara umum, promosi itu diberikan pada bos rekomendasi atau unit kerja masing-masing berdasarkan pekerjaan lama, penilaian kinerja dan penilaian perilaku karyawan dalam melaksanakan tugasnya. Untuk itu maka diperlukan penilaian karyawan. Saat ini pengolahan data penilaian karyawan perusahaan masih dilakukan dengan komputersasi excel, sehingga semakin besar risiko kesalahan memasukkan mengingat jumlah karyawan sangat banyak dan dan dibutuhkan waktu yang relatif lama. Hal ini juga masih sering membingungkan informasi mengenai pergerakan pembentukan karyawan. Kriteria yang digunakan adalah masa kerja, penilaian kinerja dan perilaku. Metode yang digunakan dalam menentukan promosi Promosi ini Simple Additive Weight (SAW).

Penelitian selanjutnya dilakukan oleh (Murtopo, dkk. 2016) tentang perancangan sistem pendukung keputusan seleksi penerimaan pegawai menggunakan metode SAW pada PDAM Tirta Dharma Tegal. Salah satu contoh,

Dalam praktek seleksi penerimaan pegawai pada PDAM Tegal dimana profesional dalam penerimaan pegawai baru sangat di perlukan, banyak hal bisa terjadi dalam proses penerimaan pegawai baru jika hanya mengandalkan diterima karena adanya hubungan pertemanan, keluarga ataupun penyusunan. Selain itu kecepatan dan keakuratan data sangat diperlukan dalam pengambilan keputusan dalam proses seleksi pegawai. Melihat hal itu selain dukungan teknologi informasi dibutuhkan juga satu sistem untuk mendukung pengambilan keputusan sebagai alat bantu menentukan pegawai baru. Untuk mendukung keakuratan data yang diolah oleh sebuah sistem maka diperlukan sebuah metode, dalam penelitian ini metode yang digunakan adalah SAW. Kriteria yang digunakan untuk proses seleksi dalam penelitian ini antara lain tes tertulis, psikotes, pendidikan, IPK dan wawancara. Hasil dari penelitian sebuah perancangan sistem pendukung keputusan sebagai alat bantu untuk pengambilan keputusan manajemen, nilai kriteria dan bobot yang sudah ditentukan sehingga akan mendapatkan hasil yang lebih akurat terhadap siapa yang akan diterima menjadi pegawai baru di suatu perusahaan.

Berdasarkan penelitian yang dilakukan (Solikhah, dkk. 2016) mengenai perancangan sistem pendukung keputusan pemilihan pelanggan terbaik menggunakan metode *Simple Additive Weighting* (SAW) pada Bravo Supermarket Jombang. Untuk menghadapi persaingan bisnis tersebut banyak cara yang dapat dilakukan oleh Bravo untuk tetap mempertahankan pelanggannya, diantaranya dengan melakukan analisis terhadap pelanggan-pelanggan tersebut dan memilih pelanggan yang terbaik. Pelanggan- pelanggan yang terbaik akan menjadi

prioritas utama bagi pihak Bravo, seperti pemberian reward (penghargaan) dan menjalin kerjasama dengan pelanggan. Sistem yang saat ini berjalan di Bravo adalah dengan melakukan undian secara acak kepada para pelanggannya, yang mana sistem tersebut belum akurat dan belum tepat sasaran. Oleh karena itu perlunya sebuah Perancangan Sistem Pendukung Keputusan (SPK) untuk menganalisis dan memilih pelanggan terbaik berdasarkan kriteria total belanja, keaktifan belanja, penghasilan pelanggan dan alamat pelanggan sehingga keputusan yang diberikan bisa tepat sasaran. Metode yang digunakan dalam perancangan sistem pendukung keputusan ini menggunakan metode SAW (Simple Additive Weighting) dalam proses pemilihan pelanggan terbaik.

Didasarkan kepada hasil ketiga penelitian terdahulu dapat dikemukakan bahwa metode *Simple Additive Weighting* (SAW) dapat menghasilkan sistem pendukung keputusan yang akurat. Dapat dilihat perbedaan dari ketiga penelitian tersebut terdapat dari segi kasusnya. Hasil penelitian diatas dapat dijadikan rujukan atau referensi dalam penelitian tentang sistem pendukung keputusan pemilihan lokasi usaha menggunakan metode *Simple Additive Weighting* (SAW) dengan kriteria yang sesuai.

Selanjutnya penelitian terdahulu yang pernah dilakukan (Pembari, 2017) mengenai sistem pendukung keputusan untuk menentukan lokasi pendirian pabrik kelapa sawit. Ada beberapa metode yang dapat digunakan dalam membangun SPK diantaranya *Simple Additive Weighting*. Konsep dasar dari metode simple additive weighting (SAW) adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif dari semua atribut. Kriteria yang

digunakan yakni kepadatan tanah lokasi pabrik, ketersediaan air, jarak dengan sungai, jarak dengan areal potensi tandan buah segar (TBS), jarak dengan tempat pengiriman *Crude Palm Oil* (CPO), akses jalan jarak ke pemukiman, jarak ke pasar dan luas tanah. Berdasarkan hasil penelitian yang telah dilakukan terdapat perbedaan studi kasus yaitu penelitian yang akan dilakukan dalam pemilihan lokasi usaha yaitu di Kota Pekanbaru. Pada penelitian yang terdahulu memiliki kriteria yang berbeda juga dengan kriteria yang akan digunakan pada penelitian yang akan dilakukan menggunakan kriteria harga sewa, kelengkapan ruko, jumlah penduduk, lahan parkir, jumlah usaha disekitar lokasi, angkutan umum, akses lokasi, target pasar, keamanan, tingkat keramaian, dan jarak lokasi dengan pusat kota. Pada penelitian yang akan dilakukan nilai bobot akan di inputkan oleh calon penyewa itu sendiri sesuai dengan tingkat kebutuhannya.

## **2.2 Dasar Teori**

### **2.2.1 Lokasi Usaha**

Lokasi Usaha adalah tempat perusahaan beroperasi atau tempat perusahaan melakukan kegiatan untuk menghasilkan barang dan jasa yang mementingkan segi ekonominya (Tjiptono, 2002). Menentukan lokasi usaha yang strategis memiliki peran andil yang besar dalam menentukan keberhasilan suatu perusahaan. Pemilihan lokasi usaha ini tentunya akan mempertimbangkan antara strategis pemasaran jasa dan preferensi pemilik. Memilih lokasi yang tepat berarti menghindari sebanyak mungkin efek-efek negatif yang mungkin timbul dari penentuan lokasi yang kurang strategis, tanpa perencanaan lokasi yang tepat, perusahaan dapat membuat kesalahan-kesalahan dalam pemilihan lokasi.

Pemilihan lokasi suatu usaha akan mempengaruhi risiko dan keuntungan perusahaan tersebut secara keseluruhan, mengingat lokasi sangat mempengaruhi biaya tetap maupun biaya variabel, baik dalam jangka menengah maupun jangka panjang. Sebagai contoh, biaya transportasi saja bisa mencapai 25% harga jual produk (tergantung kepada produk dan tipe produksi atau jasa yang diberikan). Hal ini berarti bahwa seperempat total pendapatan perusahaan mungkin dibutuhkan hanya untuk menutup biaya pengangkutan bahan mentah yang masuk dan produk jasa yang keluar dari perusahaan. (Heizer & Render, 2006).

Mengacu pada pendapat di atas dapat dikemukakan bahwa menentukan lokasi usaha dapat mempertimbangkan informasi-informasi yang sudah didapatkan seperti harga sewa, kelengkapan ruko, jumlah penduduk, lahan parkir, jumlah usaha disekitar lokasi, angkutan umum, akses lokasi, target pasar, keamanan, tingkat keramaian, jarak lokasi dengan pusat kota. Jadi informasi yang diterima calon penyewa akan mempengaruhi keputusan dalam menentukan lokasi pendirian usaha. Oleh karena itu kecepatan informasi yang diterima calon penyewa menjadi hal yang menentukan dalam mempengaruhi pengambilan keputusan untuk menentukan lokasi pendirian usaha.

### 2.2.2 Sistem Pendukung Keputusan

Menurut Alter (Kusrini, 2007), pendukung keputusan menggunakan pendekatan sistematis dengan cara melakukan pengumpulan fakta-fakta yang ada kemudian menentukan alternatif yang matang dan melakukan perhitungan untuk tindakan yang paling tepat. Sering kali pembuatan mengalami kerumitan dalam pengambilan keputusan dikarenakan banyak data yang ada. Sistem pendukung keputusan membantu pengambilan keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur di mana tidak ada yang tahu pasti bagaimana keputusan yang harus diambil.

Sistem pendukung keputusan merupakan panggabungan dari sumber-sumber kecerdasan individu yang memiliki kemampuan untuk dapat memperbaiki kualitas dari suatu yang memiliki kemampuan untuk dapat memperbaiki kualitas dari suatu keputusan. Dari pengertian tersebut dapat disimpulkan sistem pendukung keputusan bukan merupakan alat pengambilan keputusan melainkan sistem yang membantu dalam mengambil keputusan atau menunjang keputusan yang didasari oleh informasi dari data yang diolah dengan relevan yang diperlukan untuk membuat tentang suatu masalah dengan cepat dan akurat, sehingga sistem tidak dimaksud untuk menggantikan pengambilan keputusan dalam proses pembuatan keputusan.

Menurut (Hermawan, 2005), proses pengambilan keputusan melibatkan 4 tahapan, yaitu:

### 1. Tahap *Intelligence*

Dalam tahap ini pengambil keputusan mempelajari kenyataan yang terjadi sehingga kita bisa mengidentifikasi dan mendefinisikan masalah yang sedang terjadi, biasanya dilakukan analisis berurutan dari sistem ke subsistem pembentuknya. Dari tahap ini didapatkan keluaran berupa dokumen pernyataan masalah.

### 2. Tahap *Design*

Dalam tahap ini pengambil keputusan menemukan, mengembangkan, dan menganalisis semua pemecahan yang mungkin, yaitu melalui pembuatan model yang bisa mewakili kondisi nyata masalah. Dari tahap ini didapatkan keluaran berupa dokumen alternatif solusi.

### 3. Tahap *Choice*

Dalam tahap ini pengambil keputusan memilih salah satu alternatif pemecahan yang dibuat pada tahap design yang dipandang sebagai aksi yang paling tepat untuk mengatasi masalah yang sedang dihadapi. Dari tahap ini didapatkan keluaran berupa dokumen solusi dan rencana implementasinya.

### 4. Tahap *Implementation*

Dalam tahap ini pengambil keputusan menjalankan rangkaian aksi pemecahan yang dipilih di tahap choice. Implementasi yang sukses ditandai dengan terjawabnya masalah yang dihadapi, sementara kegagalan ditandai dengan tetap adanya masalah yang sedang dicoba untuk diatasi. Dari tahap ini didapatkan keluaran berupa laporan pelaksanaan solusi dan hasilnya.

Menurut (Hermawan, 2005), Sistem Pendukung Keputusan terdiri atas tiga komponen penting, yaitu:

a. Manajemen Data

Data Management melakukan pengambilan data yang diperlukan baik dari *database* yang berisi data internal maupun *database* yang berisi data eksternal. Jadi, fungsi komponen data ini sebagai pengatur data-data yang diperlukan oleh sistem pendukung keputusan.

b. Manajemen Model

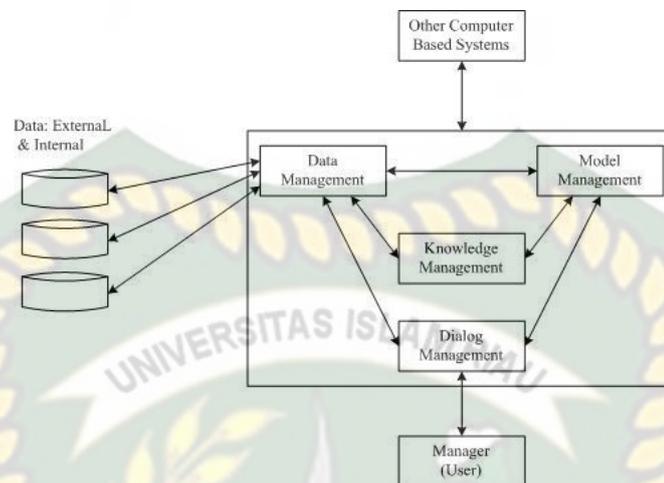
Model Management melalui Model Base Management melakukan interaksi baik dengan User Interface untuk mendapatkan perintah maupun Data Management untuk mendapatkan data yang akan diolah. Jadi, tujuan dari model management adalah untuk mengubah data yang ada pada *database* menjadi informasi yang berguna dalam pengambilan keputusan.

c. Antarmuka Pengguna

*User Interface* digunakan untuk berinteraksi antara user dengan DSS, baik untuk memasukkan informasi ke sistem maupun menampilkan informasi ke user. Karena begitu pentingnya komponen user interface bagi suatu sistem DSS, maka harus bisa merancang suatu user interface yang bisa mudah dipelajari dan digunakan user dan laporan yang bisa digunakan user serta pelaporan yang bisa secara mudah dimengerti oleh pengguna.

Komponen-komponen tersebut membentuk sistem aplikasi sistem pendukung keputusan yang bisa dikoneksikan ke intranet perusahaan, ekstranet atau internet.

Arsitektur dari sistem pendukung keputusan ditunjukkan pada gambar 2.1



**Gambar 2.1** Arsitektur Sistem Pendukung Keputusan

Dalam konteks penelitian ini, sistem pemilihan lokasi pendirian usaha merupakan sistem antar komponen berupa produk dan jasa. Komponen produk dan jasa saling berhubungan sebagai sebuah sistem yang dapat menghasilkan saran untuk pemilihan lokasi pendirian usaha. Komponen tersebut dapat membantu calon penyewa dalam mengetahui detail produk.

### 2.2.3 Simple Additive Weighting (SAW)

Metode *Simple Additive Weighting* (SAW) sering juga dikenal istilah metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut (Fishburn, 1967) (MacCrimmon, 1968). Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada.

Langkah penyelesaian metode SAW adalah sebagai berikut:

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.

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\text{Max}X_{ij}} & ; \text{ Jika } j \text{ adalah atribut keuntungan} \\ \frac{\text{Min}X_{ij}}{X_{ij}} & ; \text{ Jika } j \text{ adalah atribut biaya (cost)} \end{cases} \quad (2.1)$$

Dimana  $r_{ij}$  adalah rating kinerja ternormalisasi dari alternatif  $A_i$  pada atribut  $C_j = 1, 2, \dots, m$  dan  $j = 1, 2, \dots, n$ .

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 terbaik ( $A_i$ ) sebagai solusi.

Nilai preferensi untuk setiap alternatif ( $V_i$ ) diberikan sebagai berikut:

$$V_i = \sum_{j=1}^n w_j r_{ij} \quad r_{ij} \quad (2.2)$$

Keterangan:

$V_i$  = ranking untuk setiap alternatif

$w_j$  = nilai bobot dari setiap kriteria

$r_{ij}$  = nilai rating kinerja ternormalisasi

Nilai  $V_i$  yang lebih besar mengindikasikan bahwa alternatif lebih terpilih.

Contoh Perhitungan manual dapat dilihat sebagai berikut :

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan.

**Tabel 2.1 Kriteria**

<b>Kriteria</b>	<b>Keterangan</b>	<b>Jenis Atribut</b>
C1	Harga Sewa	<i>Cost</i>
C2	Kelengkapan ruko	<i>Benefit</i>
C3	Jumlah Penduduk	<i>Benefit</i>
C4	Lahan Parkir	<i>Benefit</i>
C5	Jumlah Usaha Di Sekitar Lokasi	<i>Cost</i>
C6	Angkutan Umum	<i>Benefit</i>
C7	Akses Lokasi	<i>Benefit</i>
C8	Target Pasar	<i>Benefit</i>
C9	Keamanan	<i>Benefit</i>
C10	Tingkat Keramaian	<i>Benefit</i>
C11	Jarak Lokasi Dengan Pusat Kota	<i>Benefit</i>

Berikut merupakan tabel- tabel kriteria dengan bobot nilai nya sebagai berikut:

1. Harga Sewa (Pertahun)
2. Kelengkapan Ruko

Dikatakan sangat lengkap apabila isi ruko mencakup tersedia semuanya (air,listriik,perabot). lengkap apabila salah satu nya terdapat dalam ruko (air /

listrik/ perabot). tidak lengkap apabila isi ruko kosong (tidak tersedia air, listrik ataupun perabot).

Bobot Kriteria Kelengkapan Ruko ditampilkan pada Tabel 2.2.

**Tabel 2.2 Kriteria Kelengkapan Ruko**

<b>Kelengkapan ruko</b>	<b>nilai</b>
Tidak lengkap	1
Lengkap	2
Sangat lengkap	3

### 3. Jumlah Penduduk Sekitar

Dikatakan banyak apabila jumlah penduduk  $> 50$  orang. Sedang apabila jumlah penduduk 20-50 orang. Sedikit apabila jumlah penduduk  $< 20$  orang.

Bobot Kriteria Jumlah Penduduk ditampilkan pada Tabel 2.3.

**Tabel 2.3 Kriteria Jumlah Penduduk**

<b>Jumlah penduduk</b>	<b>nilai</b>
sedikit	1
sedang	2
banyak	3

### 4. Lahan Parkir

Dikatakan sangat luas apabila tempat parkir dapat memuat  $> 5$  kendaraan (mobil dan motor). Luas apabila tempat parkir memuat 1-5 kendaraan. Tidak ada apabila tidak ada tempat parkir

Bobot Kriteria Lahan Parkir ditampilkan pada Tabel 2.4.

**Tabel 2.4 Kriteria Lahan Parkir**

<b>Lahan parkir</b>	<b>nilai</b>
Tidak ada	1
luas	2
Sangat luas	3

#### 5. Jumlah Usaha Di Sekitar Lokasi

Dikatakan banyak apabila terdapat usaha disekitar lokasi > 20 usaha. Sedang apabila terdapat usaha disekitar lokasi 10- 20 usaha. Sedikit apabila terdapat usaha disekitar lokasi <10usaha.

Bobot Kriteria Lahan Parkir ditampilkan pada Tabel 2.5.

**Tabel 2.5 Kriteria Jumlah Usaha Di Sekitar Lokasi**

<b>Jumlah usaha disekitar lokasi</b>	<b>nilai</b>
Sedikit	1
sedang	2
banyak	3

#### 6. Angkutan Umum

Dikatakan banyak apabila lokasi dilewati angkutan umum > 4 jenis (transmetro, angkot, taksi, bajai dll). Sedang lokasi dilewati angkutan umum 1-4 jenis. Tidak ada apabila tidak ada angkutan umum yang lewat disekitar lokasi.

Bobot Kriteria Angkutan Umum ditampilkan pada Tabel 2.6.

**Tabel 2.6 Kriteria Angkutan Umum**

<b>Angkutan umum</b>	<b>nilai</b>
Tidak ada	1
sedang	2
banyak	3

#### 7. Akses Lokasi

Dikatakan mudah apabila jalan menuju lokasi dapat dilewati > 3 jenis kendaraan (mobil, motor, truk dll). Sedang apabila jalan menuju lokasi dapat dilewati hanya 2 jenis kendaraan (mobil dan motor). Sulit apabila jalan menuju lokasi dapat dilewati hanya 1 jenis kendaraan (motor)

Bobot Kriteria Akses Lokasi ditampilkan pada Tabel 2.7

**Tabel 2.7 Kriteria Akses Lokasi**

<b>Akses lokasi</b>	<b>nilai</b>
sulit	1
sedang	2
mudah	3

#### 8. Target Pasar

Dikatakan jauh apabila target pasar (sekolah, kampus, kantor, puskesmas) >10 km.

Sedang apabila target pasar 5 - 10 km. Dekat apabila target pasar < 5 km.

Bobot Kriteria Target Pasar ditampilkan pada Tabel 2.8.

**Tabel 2.8 Kriteria Target Pasar**

<b>Target pasar</b>	<b>nilai</b>
jauh	1
sedang	2
dekat	3

#### 9. Keamanan

Dikatakan aman apabila tidak pernah terjadi kejahatan di sekitar lokasi usaha dalam 1 bulan. Dikatakan sedang apabila terjadi kejahatan 1- 2 kali dalam sebulan sekitar di lokasi usaha. Dan dikatakan tidak aman apabila terjadi kejahatan lebih dari 2 kali dalam sebulan di sekitar lokasi usaha tersebut.

Bobot Kriteria Keamanan ditampilkan pada Tabel 2.9.

**Tabel 2.9 Kriteria Keamanan**

<b>keamanan</b>	<b>nilai</b>
Tidak aman	1
sedang	2
aman	3

#### 10. Tingkat Keramaian

Dikatakan tidak ramai apabila kendaraan yang melintas < 20 unit/30 menit.

Dikatakan sedang apabila kendaraan yang melintas antara 20-50 unit/30 menit. Dan dikatakan sangat ramai apabila kendaraan yang melintas >50 unit/30 menit.

Bobot Kriteria Tingkat Keramaian ditampilkan pada Tabel 2.10.

**Tabel 2.10 Kriteria Tingkat Keramaian**

Tingkat keramaian	nilai
Tidak ramai	1
sedang	2
Sangat ramai	3

## 11. Jarak Lokasi Dengan Pusat Kota

Dikatakan jauh apabila pusat kota  $> 15$  km. Sedang apabila pusat kota 10- 15 km. Dekat apabila pusat kota  $< 10$  km.

Bobot Kriteria Jarak Lokasi Dengan Pusat Kota ditampilkan pada Tabel 2.11.

**Tabel 2.11 Kriteria Jarak Lokasi Dengan Pusat Kota**

Jarak lokasi dengan pusat kota	nilai
jauh	1
sedang	2
dekat	3

2. Menentukan rating kecocokan setiap alternatif. Setelah kriteria dan bobot nilai dari setiap kriteria telah ditentukan maka berikut langkah-langkah penyelesaiannya.

**Tabel 2.12 Rating Kecocokan Pada Setiap Kriteria**

No	Alternatif	Kriteria										
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
1	A1	20	1	2	3	2	2	2	2	1	2	1
2	A2	15	2	3	3	2	1	3	2	3	3	3
3	A3	30	2	3	2	1	3	2	3	2	3	2
4	A4	15	3	1	2	3	2	1	1	3	1	2
5	A5	35	1	3	1	1	2	2	2	3	3	2

3. Membuat matriks keputusan berdasarkan kriteria (Ci). Kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan n jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi R. Membuat matriks keputusan X, dibuat dari tabel kecocokan sebagai berikut:

$$X = \begin{bmatrix} 20 & 1 & 2 & 3 & 2 & 2 & 2 & 2 & 1 & 2 & 1 \\ 15 & 2 & 3 & 3 & 2 & 1 & 3 & 2 & 3 & 3 & 3 \\ 30 & 2 & 3 & 2 & 1 & 3 & 2 & 3 & 2 & 3 & 2 \\ 15 & 3 & 1 & 2 & 3 & 2 & 1 & 1 & 3 & 1 & 2 \\ 35 & 1 & 3 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 2 \end{bmatrix}$$

Pertama, dilakukan normalisasi matriks R untuk menghitung nilai masing-masing kriteria berdasarkan kriteria diasumsikan, sebagai kriteria keuntungan atau biaya dengan menggunakan rumus 2.1 sebagai berikut:

$$R = \begin{bmatrix} 0.75 & 0.33 & 0.66 & 1 & 0.5 & 0.66 & 0.66 & 0.66 & 0.33 & 0.66 & 0.33 \\ 1 & 0.66 & 1 & 1 & 0.5 & 0.33 & 1 & 0.66 & 1 & 1 & 1 \\ 0.5 & 0.66 & 1 & 0.66 & 1 & 1 & 0.66 & 1 & 0.66 & 1 & 0.66 \\ 1 & 1 & 0.33 & 0.66 & 0.33 & 0.66 & 0.33 & 0.33 & 1 & 0.33 & 0.66 \\ 0.42 & 0.33 & 1 & 0.33 & 1 & 0.66 & 0.66 & 0.66 & 1 & 1 & 0.66 \end{bmatrix}$$

4. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vektor bobot yang telah diberikan oleh pengambil keputusan berikut: **Vektor Bobot : W = [0.2, 0.05, 0.1, 0.05, 0.1 0.05, 0.1, 0.05, 0.1, 0.1, 0.1]** sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (sebagai solusi dengan menggunakan rumus 2.2. Hasil yang diperoleh adalah sebagai berikut:

$$V1=(0.2)(0.75)+(0.05)(0.33)+(0.1)(0.66)+(0.05)(1)+(0.1)(0.5)+(0.05)(0.66)+(0.1)(0.66) + (0.05)(0.66) + (0.1)(0.33) + (0.1)(0.66) + (0.1)(0.33) = 0.5965$$

$$V2= (0.2)(1)+(0.05)(0.66)+(0.1)(1)+(0.05)(1)+(0.1)(0.5)+(0.05)(0.33)+(0.1)(1)+(0.05)(0.66)+(0.1)(1)+(0.1)(1)+(0.1)(1) = 0.8825$$

$$V3=(0.2)(0.5)+(0.05)(0.66)+(0.1)(1)+(0.05)(0.66)+(0.1)(1)+(0.05)(1)+(0.1)(0.66)(0.05)(1)+(0.1)(0.66)+(0.1)(1)+(0.1)(0.66) = 0.7640$$

**V4 =**

$$(0.2)(1)+(0.05)(1)+(0.1)(0.33)+(0.05)(0.66)+(0.1)(0.33)+(0.05)(0.66)+(0.1)(0.33)+(0.05)(0.33)+(0.1)(1)+(0.1)(0.33)+(0.1)(0.66) = 0.6305$$

**V5 =**

$$(0.2)(0.42)+(0.05)(0.33)+(0.1)(1)+(0.05)(0.33)+(0.1)(1)+(0.05)(0.66)+(0.1)(0.66)+(0.05)(0.66)+(0.1)(1)+(0.1)(1)+(0.1)(0.66) = 0.7150$$

Didapatkan hasil akhir perankingan lokasi usaha yaitu:

**Tabel 2.13 Perankingan Lokasi Usaha**

No	Alternatif	Akhir
1	V2	0.8825
2	V3	0.7640
3	V5	0.7150
4	V4	0.6305
5	V1	0.5965

#### 2.2.4 Bahasa Pemograman

Bahasa pemrograman yang digunakan dalam pembuatan sistem pendukung keputusan ini antara lain HTML dan PHP. Berikut adalah penjelasan dari masing-masing bahasa pemrograman:

#### 2.2.4.1 HTML

HTML adalah singkatan dari *HyperText Markup Language*. HTML merupakan *file* teks yang ditulis menggunakan aturan-aturan kode tertentu untuk kemudian disajikan ke *user* melalui suatu aplikasi *web browser* (Budi Raharjo, 2012). Setiap informasi yang tampil di *web* selalu dibuat menggunakan kode HTML. Oleh karena itu, dokumen HTML sering disebut juga sebagai *web page* (halaman *web*). Untuk membuat dokumen HTML, kita tidak tergantung pada aplikasi tertentu, karena dokumen HTML dapat dibuat menggunakan aplikasi *Text Editor* apa pun, bisa *Notepad* (untuk lingkungan *MS Windows*), *Emacs* atau *Vi Editor* (untuk lingkungan *Linux*), dan sebagainya. Agar lebih mempermudah pembuatan dokumen HTML, sekarang telah banyak tersedia aplikasi HTML Editor yang dikhususkan untuk pembuatan kode-kode HTML. Berikut ini kerangka dokumen HTML:

```
<html><!--Tag untuk memulai dokumen HTML-->
<!--Bagian Header-->
<head>
<title>Judul Halaman</title>
</head>
<!--Bagian Body-->
<body>
```

Informasi yang akan ditampilkan pada *webbrowser* harus ditulis dalam bagian ini.

```
</body>
```

```
</html><!--Tag untuk mengakhiri dokumen HTML-->
```

---

#### 2.2.4.2 PHP

PHP (*Personal Home Page*) adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi *web* (Budi Raharjo, 2012). Ketika dipanggil dari *webbrowser*, program yang ditulis dengan PHP dan diterjemahkan kedalam dokumen HTML, yang selanjutnya akan ditampilkan kembali ke *web browser*. Karena pemrosesan program PHP dilakukan di lingkungan *webserver*, PHP dikatakan sebagai bahasa sisi *server* (*server-side*). Oleh sebab itu, seperti yang telah dikemukakan sebelumnya, kode PHP tidak akan terlihat pada saat user memilih perintah “View Source” pada *webbrowser* yang mereka gunakan. Selain menggunakan PHP, aplikasi *web* juga dapat dibangun dengan Java (JSP – *JavaServer Pages* dan *Servlet*), Perl, maupun ASP (*Active Server Pages*).

Berikut ini contoh kode PHP yang sangat sederhana.

```
<?php
    echo “Hello World! ”;
?>
```

Perintah `echo` di dalam PHP berguna untuk mencetak nilai, baik teks maupun numerik, ke layar *web browser*. Selain `echo`, kita juga dapat menggunakan perintah `print`, sehingga kode di atas dapat juga ditulis sebagai berikut.

```
<?php
    print “Hello World! ”;
```

?>

---

Cara kerja aplikasi *web* yang ditulis PHP:

1. User menulis *www.abcd.com/catalog.php* ke dalam *address* bar dari *web browser* (IE, Mozilla Firefox, Opera, dll)
2. *Web browser* mengirimkan pesan di atas ke komputer *server* (*www.abcd.com*) melalui internet, meminta halaman *catalog.php*
3. *Web server* (midalnya *Apache*), program yang berjalan di komputer *server*, akan menangkap pesa tersebut, lalu meminta interpreter PHP (program lain yang juga berjalan di komputer *server*) untuk mencari *filecatalog.php* dalam *disk drive*.
4. Interpreter PHP membaca *filecatalog.php* dari *disk drive*.
5. Interpreter PHP akan menjalankan perintah-perintah atau kode PHP yang ada dalam *filecatalog.php*. Jika kode dalam *catalog.php* melibatkan akses terhadap *database* (misalnya MySQL) maka interpreter PHP juga akan berhubungan dengan MySQL untuk melaksanakan perintah-perintah yang berkaitan dengan *database*.
6. Interpreter PHP mengirimkan halaman dalam bentuk HTML ke *Apache*.
7. Melalui internet, *Apache* mengirimkan halaman yang diperoleh dari interpreter PHP ke komputer *user* sebagai respon atas permintaan yang diberikan.
8. *Web browser* dalam komputer *user* akan menampilkan halaman yang dikirim oleh *Apache*.

### 2.2.5 Database

Data merupakan fakta mengenai suatu objek seperti manusia, benda, peristiwa, konsep, keadaan dan sebagainya yang dapat dicatat dan mempunyai arti secara implisit. Data dapat dinyatakan dalam bentuk angka, karakter atau simbol, sehingga bila data dikumpulkan dan saling berhubungan maka dikenal dengan istilah basis data (Elmasri Ramez, 2000). Sedangkan menurut George Tsuder Chou basis data merupakan kumpulan informasi bermanfaat yang diorganisasikan ke dalam aturan yang khusus. Informasi ini adalah data yang telah diorganisasikan ke dalam bentuk yang sesuai dengan kebutuhan seseorang (Abdul Kadir, 1999). Menurut *Encyclopedia of Computer Science and Engineer*, para ilmuwan di bidang informasi menerima definisi standar informasi yaitu data yang digunakan dalam pengambilan keputusan. Definisi lain dari basis data menurut Fabbri dan Schwab adalah sistem berkas terpadu yang dirancang terutama untuk meminimalkan duplikasi data.

Menurut Ramez Elmasri mendefinisikan basis data lebih dibatasi pada arti implisit yang khusus, yaitu:

1. Basis data merupakan penyajian suatu aspek dari dunia nyata (*real world*).
2. Basis data merupakan kumpulan data dari berbagai sumber yang secara logika mempunyai arti implisit. Sehingga data yang terkumpul secara acak dan tanpa mempunyai arti, tidak dapat disebut basis data.
3. Basis data perlu dirancang, dibangun dan data dikumpulkan untuk suatu tujuan. Basis data dapat digunakan oleh beberapa *user* dan beberapa aplikasi yang sesuai dengan kepentingan *user*.

Dari beberapa definisi-definisi tersebut, dapat dikatakan bahwa basis data mempunyai berbagai sumber data dalam pengumpulan data, bervariasi derajat interaksi kejadian dari dunia nyata, dirancang dan dibangun agar dapat digunakan oleh beberapa *user* untuk berbagai kepentingan (Waliyanto, 2000).

Pada hirarki data, data diorganisasikan kedalam bentuk elemen data (*field*), rekaman (*record*), dan berkas (*file*). Definisi dari ketiganya adalah sebagai berikut:

1. Elemen data adalah satuan data terkecil yang tidak dapat dipecah lagi menjadi unit lain yang bermakna. Misalnya data siswa terdiri dari NIS, Nama, Alamat, Telepon atau Jenis Kelamin.
2. Rekaman merupakan gabungan sejumlah elemen data yang saling terkait. Istilah lain dari rekaman adalah baris atau tupel.
3. Berkas adalah himpunan seluruh rekaman yang bertipe sama.

Model data dapat dikelompokkan berdasarkan konsep pembuatan deskripsi struktur basis data, yaitu:

1. Model data konseptual (*high level*) menyajikan konsep tentang bagaimana user memandang atau memperlakukan data. Dalam model ini dikenalkan tiga konsep penyajian data yaitu:
  - a. *Entity* (entitas) merupakan penyajian obyek, kejadian atau konsep dunia nyata yang keberadaannya secara eksplisit didefinisikan dan disimpan dalam basis data, contohnya Mahasiswa, Matakuliah, Dosen, Nilai dan lain sebagainya.

- b. *Attribute* (atribut) adalah keterangan-keterangan yang menjelaskan karakteristik dari suatu entitas seperti NIM, Nama, Fakultas, Jurusan untuk entitas Mahasiswa.
  - c. *Relationship* (hubungan) merupakan hubungan atau interaksi antara satu entitas dengan yang lainnya, misalnya entitas pelanggan berhubungan dengan entitas barang yang dibelinya.
2. Model data fiskal (*low level*) merupakan konsep bagaimana deskripsi detail data disimpan ke dalam komputer dengan menyajikan informasi tentang format rekaman, urutan rekaman, dan jalur pengaksesan data yang dapat membuat pencarian rekaman data lebih efisien.
  3. Model data implementasi (*representational*) merupakan konsep deskripsi data disimpan dalam komputer dengan menyembunyikan sebagian detail deskripsi data sehingga para user mendapat gambaran global bagaimana data disimpan dalam komputer. Model ini merupakan konsep model data yang digunakan oleh model hirarki, jaringan dan relasional.

Komponen-komponen DBMS terdiri dari:

1. *Interface*, yang didalamnya terdapat bahasa manipulasi data (*data manipulation language*).
2. Bahasa definisi data (*data definition language*) untuk skema eksternal, skema konseptual dan skema internal.
3. Sistem kontrol basis data (*Database Control System*) yang mengakses basis data karena adanya perintah dari bahasa manipulasi data.

Contoh bahasa menggunakan komponen-komponen tersebut adalah SQL (*Structured QueryLanguage*). Pada penelitian ini akan menggunakan SQL karena merupakan bahasa standar yang digunakan oleh kebanyakan aplikasi-aplikasi DBMS.

#### 2.2.5.1 Pengertian Dasar *MySQL*

*MySQL* merupakan sistem *database* yang banyak digunakan untuk pengembangan aplikasi web. Alasannya mungkin karena gratis, pengelolaan datanya sederhana, memiliki tingkat keamanan yang bagus, mudah diperoleh, dan lain-lain (Budi Raharjo, 2012).

Untuk memanipulasi data pada tabel-tabel yang terdapat didalam suatu *database*, berikut perintah-perintah yang perlu diketahui (Budi Raharjo, 2012):

- a. *SELECT*: digunakan untuk mengambil data dari *database*.
- b. *DELETE*: digunakan untuk menghapus data dari *database*.
- c. *INSERT*: digunakan untuk memasukkan data baru ke dalam *database*.
- d. *REPLACE*: digunakan untuk menggantikan data di dalam *database*. Jika terdapat *record* yang sama dalam suatu tabel, perintah ini akan menimpa *record* tersebut dengan yang data yang baru.
- e. *UPDATE*: digunakan untuk mengubah data di dalam suatu tabel.

Perintah-perintah di atas hanya digunakan untuk memanipulasi data. Untuk memanipulasi struktur objek *database*, gunakan perintah-perintah berikut:

- a. *CREATE*: digunakan untuk membuat *database*, tabel, atau indeks.
- b. *ALTER*: digunakan untuk memodifikasi struktur dari suatu tabel.
- c. *DROP*: digunakan untuk menghapus *database*, tabel, atau indeks.

## 2.2.6 Perancangan Sistem

### 2.2.6.1 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut (Andi Kristanto, 2003).

DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem. DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

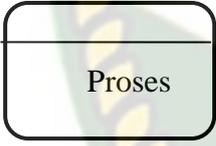
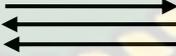
Fungsi dari Data Flow Diagram adalah:

1. Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi.
2. DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata

lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem.

- DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

**Tabel 2.14 Simbol Data Flow Diagram (DFD)**

<i>Gane/Sarson</i>	<i>Yourdon/De Marco</i>	Keterangan
		Entitas eksternal dapat berupa orang/unit terkait yang berinteraksi dengan sistem tetapi diluar sistem.
		Orang/unit yang mempergunakan atau melakukan transformasi data. Komponen fisik tidak diidentifikasi.
Aliran Data 	Aliran Data 	Aliran data dengan arah khusus dari sumber ke tujuan.
		Penyimpanan data atau tempat data yang dilihat oleh proses.

Didalam DFD terdapat 3 level, yaitu:

1. Diagram Konteks: menggambarkan satu lingkaran besar yang dapat mewakili seluruh proses yang terdapat di dalam suatu sistem. Merupakan tingkatan tertinggi dalam DFD dan biasanya diberi nomor 0 (nol). Semua entitas eksternal yang ditunjukkan pada diagram konteks berikut aliran-aliran data utama menuju dan dari sistem. Diagram ini sama sekali tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan.
2. Diagram Nol (diagram level-1): merupakan satu lingkaran besar yang mewakili lingkaran-lingkaran kecil yang ada di dalamnya. Merupakan pemecahan dari diagram Konteks ke diagram nol dan diagram ini memuat penyimpanan data.
3. Diagram Rinci: merupakan diagram yang menguraikan proses apa yang ada dalam diagram nol.

#### **2.2.6.2 Entity Relationship Diagram (ERD)**

Dalam rekayasa perangkat lunak, sebuah *Entity-Relationship Model* (ERM) merupakan abstrak dan konseptual representasi data. *Entity-Relationship* adalah salah satu metode pemodelan basis data yang digunakan untuk menghasilkan skema konseptual untuk jenis/model data semantik sistem. Dimana sistem seringkali memiliki basis data relasional, dan ketentuannya bersifat *top-down*. Diagram untuk menggambarkan model *Entity-Relationship* ini disebut *Entity-Relationship diagram*, *ER diagram*, atau ERD.

Pada rancangan konseptual diperlukan suatu pendekatan yang digunakan untuk menggambarkan hubungan antar data. Hubungan tersebut dapat dinyatakan

dalam bentuk model E-R. Mengingat model E-R adalah dasar penting dalam perancangan *dataset* (Abdul Kadir, 2002).

Jika diterapkan dengan benar atar tepat maka penggunaan ERD dalam pemodelan data memberikan keuntungan bagi perancang maupun pengguna, berikut kelebihan dan kelemahan ERD (Edhy Sutanta, 2011):

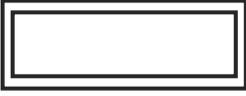
1. Memudahkan perancangan dalam hal menganalisis sistem yang akan dikembangkan.
2. Memudahkan perancangan saat merancang basis data.
3. Rancangan basis data yang dikembangkan berdasarkan ERD umumnya telah berada dalam bentuk optimal.
4. Dengan menggunakan ERD, pengguna umumnya mudah memahami sistem dan basis data yang dirancang oleh perancang.

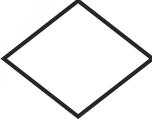
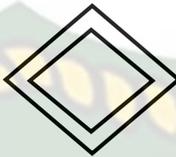
Kelemahan ERD di antaranya adalah (Edhy Sutanta, 2011):

1. Kebutuhan media yang sangat luas.
2. Sering kali ERD tampil sangat ruwet.

Notasi-notasi simbolik di dalam diagram ERD yang digunakan adalah sebagai berikut:

**Tabel 2.15 Simbol *Entity Relationship Diagram***

Simbol	Arti
	<i>Entity</i>
	<i>Weak Entity</i>

	<i>Relationship</i>
	<i>Identifying Relationship</i>
	<i>Atribut</i>
	<i>Atribut Primery Key</i>
	<i>Atribut Multivelue</i>
	<i>Atribut Compisite</i>

a. Entitas (*Entity*) dan Himpunan Entitas (*EntitySets*)

Entitas merupakan individu yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Sekelompok entitas yang sejenis dan berbeda dalam lingkup yang sama membentuk sebuah himpunan entitas (*Entity sets*).

b. Atribut (*attributes / properties*)

Setiap entitas pasti memiliki atribut yang mendeskripsikan karakteristik (*properties*) dan entitas tersebut. Penentuan/pemilihan atribut-atribut yang relevan

bagi sebuah entitas merupakan hal penting lainnya dalam pembentukan model data. Penempatan atribut sebuah entitas umumnya di dasarkan pada fakta yang ada, tetapi tidak selalu demikian.

c. Relasi (*Relationship*) dan Himpunan Relasi (*Relationship Sets*)

Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berasal dari himpunan entitas yang berada. Kumpulan semua relasi diantara entitas himpunan entitas tersebut membentuk himpunan relasi (*Relationship Sets*).

d. Kardinalitas/Derajat Relasi

Kardinalitas menunjukkan jumlah maksimum entitas yang dapat beralasi dengan entitas pada himpunan entitas yang lain. Dari sejumlah kemungkinan banyaknya hubungan tersebut, kardinalitas relasi merujuk kepada hubungan maksimum yang terjadi dari himpunan entitas yang satu ke himpunan entitas yang lain dan begitu juga sebaliknya.

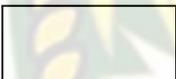
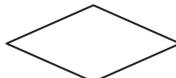
ERD dirancang untuk menggambarkan persepsi dari pemakai dan berisi Objek-objek dasar yang disebut entitas dan hubungan antar entitas tersebut yang disebut dengan *relationship*. Pada model ERD ditransformasikan dengan memanfaatkan perangkat konseptual menjadi sebuah diagram, yaitu ER (*Entity Relationship*). Diagram *Entity-Relationship* melengkapi penggambaran grafik dari struktur logika. Diagram E-R menggambarkan arti dari aspek seperti entitas-entitas, atribut-atribut, *relationship-relationship* disajikan (Deni Darmawan, 2013).

### 2.2.6.3 Pogram *Flowchart*

*Flowchart* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian

dari suatu algoritma(Ladjamudin Al Bahra, 2005). Adapun simbol-simbol yang digunakan dalam *flowchart* dapat dilihat pada tabel berikut ini.

**Tabel 2.16 Aliran Sistem (*Flowchart*)**

Simbol	Nama	Fungsi
	<b>TERMINATOR</b>	Permulaan/akhir program
	<b>GARIS ALIR (FLOW LINE)</b>	Arah aliran program
	<b>PREPARATION</b>	Proses inisialisasi/pemberian harga awal
	<b>PROSES</b>	Proses perhitungan/proses pengolahan data
	<b>INPUT/OUTPUT DATA</b>	Proses input/output data, parameter, informasi
	<b>PREDEFINED PROCESS (SUB PROGRAM)</b>	Permulaan sub program/proses menjalankan sub program
	<b>DECISION</b>	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	<b>ON PAGE CONNECTOR</b>	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	<b>OFF PAGE CONNECTOR</b>	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda

Tujuan membuat *flowchart*:

- a. Menggambarkan suatu tahapan penyelesaian masalah
- b. Secara sederhana, teratur, rapi dan jelas
- c. Menggunakan simbol-simbol standar

Dalam penulisan *flowchart* dikenal dua model, yaitu sistem *flowchart* dan program *flowchart*:

- 1) Sistem *Flowchart*, bagan yang memperlihatkan urutan prosedur dan proses dari beberapa *file* di dalam media tertentu. Melalui *flowchart* ini terlihat jenis media penyimpanan yang dipakai dalam pengolahan data. Selain itu juga menggambarkan *file* yang dipakai sebagai *input* dan *output*. Tidak digunakan untuk menggambarkan urutan langkah untuk memecahkan masalah. Hanya untuk menggambarkan prosedur dalam sistem yang dibentuk.
- 2) Program *Flowchart*, bagan yang memperlihatkan urutan dan hubungan proses dalam suatu program. Dua jenis metode penggambaran program *flowchart*:
  1. *Conceptual Flowchart*, menggambarkan alur pemecahan masalah secara global
  2. *Detail Flowchart*, menggambarkan alur pemecahan masalah secara rinci.

Simbol-simbol yang di pakai dalam *flowchart* dibagi menjadi 3 kelompok:

1. *Flow direction symbols*
  - a. Digunakan untuk menghubungkan simbol satu dengan yang lain.
  - b. Disebut juga connecting lin.
2. Processing symbols

Menunjukkan jenis operasi pengolahan dalam suatu proses/prosedu.

3. *Input / Output symbols*

Menunjukkan jenis peralatan yang digunakan sebagai media *input* atau *output*.



Dokumen ini adalah Arsip Miik :

Perpustakaan Universitas Islam Riau