



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Sistem

Jogianto (34:2008), sistem dapat didefinisikan sebagai kumpulan dari prosedur-prosedur yang mempunyai tujuan tertentu.

Kusrini (11:2007), sistem merupakan kumpulan elemen yang saling berkaitan yang bertanggung jawab memproses masukan (*input*) sehingga menghasilkan keluaran (*output*).

Sutabri (2012:3), sistem adalah suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu.

2.1.2. Pengertian Internet

Kamus Besar Bahasa Indonesia Pusat Bahasa (2013:543), internet adalah jaringan komunikasi elektronik yang menghubungkan jaringan komputer dan fasilitas komputer yang terorganisasi di seluruh dunia melalui telepon atau satelit.

2.1.3. Pengertian Web

Kamus Besar Bahasa Indonesia Pusat Bahasa (2013:1560), *web* adalah sistem untuk mengakses, memanipulasi, dan mengunduh dokumen yang terdapat dalam komputer yang dihubungkan melalui internet.

2.2. Teori Judul

2.2.1. Pengertian Sistem Pendukung Keputusan

Kusrini (15:2007), sistem pendukung keputusan merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data yang digunakan untuk membantu pengambilan keputusan dalam situasi yang



semiterstruktur dan situasi yang tidak terstruktur, dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat.

2.2.2. Pengertian Beasiswa

Menurut Undang-undang Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional, Bab V pasal 12 (1.c), menyebutkan bahwa setiap peserta didik pada setiap satuan pendidikan berhak mendapatkan beasiswa bagi yang berprestasi yang orangtuanya tidak mampu membiayai pendidikannya. Pasal 12 (1.d), menyebutkan bahwa setiap peserta didik pada setiap satuan pendidikan berhak mendapatkan biaya pendidikan bagi mereka yang orangtuanya tidak mampu membiayai pendidikannya.

2.2.3. Pengertian Fuzzy MADM

Kusumadewi.et.al (69:2006), *multi attribute decision making* (MADM) digunakan untuk menyelesaikan masalah-masalah dalam ruang diskret yang meliputi untuk melakukan penilaian atau seleksi terhadap beberapa alternatif dalam jumlah yang terbatas.

Secara garis besar untuk penentuan permasalahan pemilihan bobot kriteria yang ditentukan dapat digambarkan sebagai berikut:

1. Representasi permasalahan

Pada langkah representasi permasalahan ada 2 aktivitas yang harus dilakukan, yaitu:

a. Identifikasi tujuan dan alternatif keputusan

Tujuan keputusan dari permasalahan ini adalah dipilihnya calon penerima beasiswa yang mempunyai rangking tertinggi. Jika ada alternatif keputusan, maka alternatif tersebut dapat ditulis sebagai $A = \{A_i \mid i = 1, 2, 3, \dots, n\}$

b. Identifikasi kumpulan kriteria

Jika ada kriteria untuk menentukan pilihan dari beberapa alternatif keputusan maka dapat dituliskan $C = \{C_t \mid t = 1, 2, 3, \dots, k\}$. membangun struktur hirarki dari masalah tersebut berdasarkan pertimbangan-pertimbangan tertentu.



2.2.4. Pengertian Simple Additive Weighting

Kusumadewi.et.al (74:2006), *simple additive weighting* (SAW) digunakan untuk mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut.

Adapun langkah-langkah dari penyelesaian yang terdapat di metode *simple additive weighting* adalah sebagai berikut:

1. Menentukan kriteria-kriteri yang akan dijadikan acuan dalam pengambilan keputusan.
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan kriteria(Ci) kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan js atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi R
4. Hasil akhir diperoleh dari proses perangkingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vektor bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (Ai) sebagai solusi.

Formula untuk melakukan normalisasi matriks adalah sebagai berikut:

$$R_{ij} = (X_{ij} / \max\{X_{ij}\})$$

Dari rumus tersebut, setiap angka yang ada di satu kolom kriteria akan dibagi dengan angka yang terbesar yang ada dikolom tersebut.

Formula untuk menghitung bobt untuk setiap alternatif (Vi) yang ada adalah sebagai berikut:

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

Keterangan:

Vi = Nilai akhir dari alternatif

Wj = Bobot yang telah ditentukan

Rij = Normalisasi matriks



2.2.5. Pengertian Sistem Pendukung Keputusan Beasiswa pada Politeknik Negeri Sriwijaya dengan menggunakan Metode *Fuzzy MADM* dan *Simple Additive Weighting*

Merupakan suatu perangkat lunak yang dikembangkan untuk difungsikan secara khusus dalam mengelolah data beasiswa yang meliputi beasiswa PPA dan BPPA, Beasiswa PGN, Beasiswa Supersemar, beasiswa PT Pama Persada, Beasiswa IKA Polsri, dan Beasiswa Stuktural yang mana system tersebut mampu membuat laporan jumlah pendaftar dan penerima yang memenuhi kriteria dari penyeleksian beasiswa tersebut.

2.3. Teori Khusus

2.3.1. *Data Flow Diagram (DFD)*

Ladjamudin (2013:64), *data flow diagram (DFD)* atau diagram alir data merupakan model dari sistem untuk menggambarkan pembagian sistem ke modul yang lebih kecil. Diagram ini memudahkan pemakai atau *user* yang kurang menguasai bidang komputer untuk mengerti sistem yang akan dikerjakan. Ada beberapa tahapan perancangan dengan menggunakan *DFD*, yaitu:

1. Diagram Konteks

Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari *DFD* yang menggambarkan seluruh *input* ke sistem atau *output* dari sistem.

2. Diagram Nol / Zero (*Overview Diagram*)

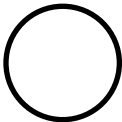
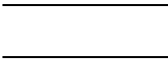
Diagram nol adalah diagram yang menggambarkan proses dari *data flow diagram*. Diagram nol memberikan pandangan secara menyeluruh mengenai sistem yang ditangani, menunjukkan tentang fungsi-fungsi utama atau proses yang ada, aliran data, dan eksternal entity.

3. Diagram Rinci (*Level Diagram*)

Diagram rinci adalah diagram yang menguraikan proses apa yang ada dalam diagram *zero* atau diagram level di atasnya.

Adapun simbol-simbol *Data Flow Diagram* adalah sebagai berikut:

**Tabel 2.1.** Simbol-Simbol *Data Flow Diagram (DFD)*

No.	Nama	Simbol	Keterangan
1.	Entitas Luar (<i>External Entity</i>)		Sesuatu yang berada di luar sistem, tetapi memberikan data ke dalam sistem atau memberikan data dari system
2.	Arus Data (<i>Data Flow</i>)		Merupakan tempat mengalirnya informasi dan menghubungkan komponen dari sistem, mengalir di antara proses, data store, dan menunjukkan arus data dari data yang berupa masukan untuk sistem atau hasil proses system
3.	Proses (<i>Process</i>)		Proses merupakan apa yang dikerjakan oleh sistem. Proses dapat mengolah data atau aliran data masuk menjadi aliran data ke luar. Proses berfungsi mentransformasikan satu atau beberapa data masukan menjadi satu atau beberapa data keluaran sesuai dengan spesifikasi yang diinginkan
4.	Simpanan Data (<i>Data Store</i>)		Simpanan data merupakan tempat penyimpanan data pengikat data yang ada dalam sistem

(Sumber: Ladjamudin, 2013:67-70)

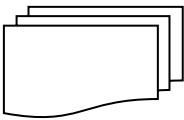
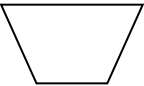
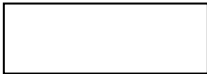
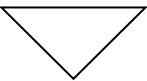
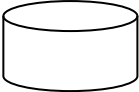


2.3.2. Block Chart

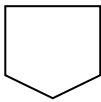
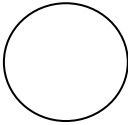
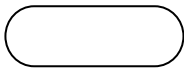
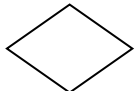
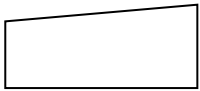
Kristanto (2008:75), *block chart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *block chart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.

Adapun simbol-simbol *Block Chart* adalah sebagai berikut:

Tabel 2.2. Simbol-Simbol *Block Chart*

No.	Simbol	Keterangan
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan
2.		Multi dokumen
3.		Proses manual
4.		Proses yang dilakukan oleh komputer
5.		Menandakan dokumen yang diarsipkan (arsip manual)
6.		Data penyimpanan (<i>data storage</i>)
7.		Proses apa saja yang tidak terdefinisi termasuk aktifitas fisik



8.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain
9.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama
10.		Terminasi yang menandakan awal dan akhir dari suatu aliran
11.		Pengambilan keputusan (<i>decision</i>)
12.		Layar peraga (<i>monitor</i>)
13.		Pemasukan data secara manual

(Sumber: Kristanto, 2008:64)

2.3.3. Flowchart

Flowchart adalah simbol-simbol pekerjaan yang menunjukkan bagan aliran proses yang saling terhubung. Jadi, setiap simbol *flowchart* melambangkan pekerjaan dan instruksinya.

Adapun simbol-simbol *Flowchart* adalah sebagai berikut:

Tabel 2.3. Simbol-Simbol *Flowchart*

No.	Simbol	Arti
1.		Simbol Start atau End yang mendefinisikan awal atau akhir dari sebuah <i>flowchart</i>



2.		Simbol pemrosesan yang terjadi pada sebuah alur kerja
3.		Simbol Input/Output yang mendefinisikan masukan dan keluaran proses
4.		Simbol untuk memutuskan proses lanjutan dari kondisi tertentu
5.		Simbol konektor untuk menyambung proses pada lembar kerja yang sama
6.		Simbol konektor untuk menyambung proses pada lembar kerja yang berbeda
7.		Simbol untuk menghubungkan antar proses atau antar simbol
8.		Simbol yang menyatakan piranti keluaran, seperti layar monitor, <i>printer</i> , dll
9.		Simbol yang mendefinisikan proses yang dilakukan secara manual
10.		Simbol masukan atau keluaran dari atau ke sebuah dokumen
11.		Simbol yang menyatakan bagian dari program (subprogram)
12.		Simbol masukan atau keluaran dari atau ke sebuah pita magnetic
13.		Simbol database atau basis data

(Sumber: Ewolf Community, 2012:17)

2.3.4. Entity Relationship Diagram (ERD)

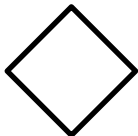
Yakub (2012:60), *entity Relationship Diagram (ERD)* merupakan suatu model jaringan yang menggunakan susunan data yang disimpan pada sistem



secara abstrak. *ERD* menggambarkan hubungan antara satu entitas yang memiliki sejumlah atribut dengan entitas yang lain dalam suatu sistem yang terintegrasi.

Adapun simbol-simbol *Entity Relationship Diagram* adalah sebagai berikut:

Tabel 2.4. Simbol-Simbol *Entity Relationship Diagram* (*ERD*)

No.	Nama	Simbol	Keterangan
1.	Entitas (<i>Entity</i>)		Kumpulan dari objek yang dapat diidentifikasi secara unik dan menunjukkan objek-objek dasar yang terkait di dalam system
2.	Atribut (<i>Attribute</i>)		Karakteristik dari entitas atau relasi yang merupakan penjelasan detail tentang entitas yang perlu disimpan sebagai basis data
3.	Relasi (<i>Relation</i>)		Hubungan yang terjadi antara satu atau lebih entitas. Jenis hubungan antara lain: satu ke satu, satu ke banyak, dan banyak ke banyak
4.	<i>Link</i>		Hubungan antara entitas dengan atributnya dan himpunan entitas dengan himpunan relasinya

(Sumber: Yakub, 2012:60)

2.3.5. Kamus Data (*Data Dictionary*)

Sukamto dan Shalahuddin (2013:73), kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar penulisan). Kamus data biasanya berisi:

- nama – nama dari data
- digunakan pada – merupakan proses-proses yang terkait data
- deskripsi – merupakan deskripsi data



- informasi tambahan – seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data

Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan sebagai berikut:

Tabel 2.5. Simbol-Simbol Kamus Data

No.	Simbol	Keterangan
1.	=	disusun atau terdiri dari
2.	+	Dan
3.	[]	baik ...atau...
4.	{ } ⁿ	n kali diulang / bernilai banyak
5.	()	data opsional
6.	*...*	batas komentar

(Sumber: Sukanto dan Shalahuddin, 2013:74)

2.3.6. Daftar Kejadian (*Event List*)

Kristanto (2008:70), menjelaskan tentang daftar kejadian (*event list*) sebagai berikut:

Daftar kejadian digambarkan dalam bentuk kalimat sederhana dan berfungsi untuk memodelkan kejadian yang terjadi dalam lingkungan sehari-hari dan membutuhkan tanggapan atau respon dari sistem. Suatu kejadian mewakili satu aliran data atau proses dalam diagram konteks serta deskripsi penyimpanan yang digunakan untuk memodelkan data harus diperhatikan dalam kaitannya dengan daftar kejadian.

Adapun cara-cara mendeskripsikan daftar kejadian adalah sebagai berikut:

1. Pelaku adalah entiti luar, jadi bukan sistem.
2. Menguji setiap entiti luar dan mencoba mengevaluasi setiap entiti luar yang terjadi pada sistem.
3. Hati-hati dengan kejadian yang spesifik, yang tak sengaja menyatu dalam paket yang sama.



4. Harus diingat bahwa kejadian yang dimodelkan bukan hanya interaksi normal antara sistem dengan entiti luar, karena itu harus dievaluasi kebutuhan sistem untuk menanggapi kejadian yang gagal.
5. Setiap aliran keluaran sebaiknya merupakan respondari kejadian.
6. Setiap kejadian yang tidak berorientasi pada waktu dalam daftar kejadian sebaiknya mempunyai masukan sehingga sistem dapat mendeteksi kejadian yang berlangsung.
7. Setiap kejadian sebaiknya menghasilkan keluaran langsung sebagai respon atau disimpan dalam berkas untuk bahan masukan.

2.4. Teori Program

2.4.1. PHP (PHP Hypertext Preprocessor)

2.4.1.1. Pengertian PHP

Madcoms (2010:341), *PHP* adalah bahasa pemrograman yang bekerja dalam sebuah webserver, dimana script *PHP* dibuat harus tersimpan dalam sebuah server dan dieksekusi atau diproses dalam server tersebut.

Nugroho (2013:153), *PHP* itu bahasa pemrograman berbasis *web*. Jadi, *PHP* itu adalah bahasa program yang digunakan untuk membuat aplikasi berbasis *web* (*website*, *blog*, atau aplikasi *web*).

2.4.1.2. Script Dasar PHP

Madcoms (2010:350), script *PHP* harus selalu diawali dengan `<?atau<?php` dan diakhiri dengan `?>`. Sedangkan untuk menampilkan ke dalam browser, digunakan fungsi `echo“data”;`. Semua teks yang diketik setelah tanda buka script (`<?`) dan tanda tutup script (`?>`) akan dieksekusi sebagai suatu script *PHP*. Contoh penulisan script *PHP* adalah sebagai berikut:

Tabel 2.6. Script Dasar *PHP*

<pre> <? ?> </pre>	Tempat penulisan script php
--------------------------------	-----------------------------



Atau	
<pre><?php</pre> <pre>?></pre>	Tempat penulisan script php
dan untuk menampilkan ke browser	
<pre><?php echo "Welcome to PHP"; ?></pre>	

(Sumber : Madcoms, 2010:350)

Di dalam script *PHP* juga dapat membuat komentar dan komentar tersebut tidak akan dieksekusi sebagai sebuah script. Beberapa cara untuk memberi keterangan di dalam script *PHP* adalah:

- Gunakan tag `/*` dan diakhiri tag `*/` apabila jumlah keterangan lebih dari 1 baris.
- Gunakan `//`, tag ini digunakan untuk keterangan yang hanya terdiri dari satu baris saja.
- Gunakan tag `#`, tag ini juga digunakan untuk 1 baris komentar saja.

Dalam penulisannya, baris perintah script *PHP* selalu diakhiri dengan menuliskan tanda titik koma (`;`). Satu baris script *PHP* tidak harus berada dalam satu baris, melainkan bisa untuk lebih dari satu baris.

2.4.2. Basis Data (*Database*)

Madcoms (2010:367), basis data (*database*) berfungsi sebagai penampung data yang diinputkan melalui *form website*.

Yakub (2012:51), basis data (*database*) merupakan kumpulan data yang saling berhubungan (punya relasi).



2.4.3. MySQL

2.4.3.1. Pengertian MySQL

Badiyanto (2013:57), *MySQL* merupakan sebuah *database server* *SQL multiuser* dan *multi threaded*.

Kadir (2008:2), *MySQL* (baca:mai-se-kyu-el) merupakan *software* yang tergolong sebagai *DBMS (Database Management System)* yang bersifat *Open Source*.

2.4.3.2. Keunggulan MySQL

Sebagai *software DBMS*, *MySQL* memiliki sejumlah fitur seperti yang dijelaskan di bawah ini:

a. Multiplatform

MySQL tersedia pada beberapa *platform (Windows, Linux, Unix, dan lain-lain)*.

b. Andal, cepat, dan mudah digunakan

MySQL tergolong sebagai *database server (server yang melayani permintaan terhadap database)* yang andal, dapat menangani *database* yang besar dengan kecepatan tinggi, mendukung banyak sekali fungsi untuk mengakses *database*, dan sekaligus mudah untuk digunakan.

c. Jaminan keamanan akses

MySQL mendukung pengamanan *database* dengan berbagai kriteria penaksesan. Sebagai gambaran, dimungkinkan untuk mengatur *user* tertentu agar bisa mengakses data yang bersifat rahasia (misalnya gaji pegawai), sedangkan *user* lain tidak boleh.

d. Dukungan SQL

Seperti tersirat dalam namanya, *MySQL* mendukung perintah *SQL (Structured Query Language)*. Sebagaimana diketahui, *SQL* merupakan standar pengaksesan *database* relasional. Pengetahuan akan *SQL* akan memudahkan siapa pun untuk menggunakan *MySQL*.