



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1 Pengertian Komputer

Komputer adalah sekumpulan alat elektronik yang saling bekerja sama, dapat menerima data (input), mengolah data (proses), dan memberikan informasi (output) serta terkoordinasi dibawah kontrol program yang tersimpan di memori komputer. Jika pada zaman sebelumnya penggunaan komputer hanya terbatas pada beberapa aktivitas saja maka pada zaman sekarang komputer dapat digunakan dalam berbagai aspek kehidupan seperti perusahaan, sekolah, perkantoran, rumah dan tempat-tempat umum sebagai layanan informasi [3].

Dari definisi di atas penulis menyimpulkan bahwa Komputer adalah perangkat keras yang dapat digunakan untuk membantu dalam berbagai pekerjaan manusia seperti pembuatan dokumen.

2.1.2 Pengertian Perangkat Lunak (Software)

Software atau Perangkat Lunak merupakan suatu data yang diprogram sedemikian rupa dan disimpan dalam bentuk digital yang tidak terlihat secara fisik tetapi tersimpan dalam media penyimpanan komputer [4].

Dari definisi di atas penulis menyimpulkan bahwa Perangkat Lunak adalah kumpulan data yang tersimpan di komputer dan dapat dikendalikan oleh perangkat komputer.

2.1.3 Pengertian Basis Data (Database)

Basis data (Database) merupakan kumpulan dari data (arsip) yang berhubungan dan diorganisasikan sedemikian rupa supaya dapat digunakan dengan cepat dan mudah [5].



Basis data (Database) adalah kumpulan informasi yang disimpan dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi [6].

Dari beberapa pendapat mengenai definisi Basis Data yang telah dikemukakan oleh beberapa para ahli di atas, maka dapat disimpulkan bahwa Basis Data atau *Database* merupakan suatu kumpulan data yang dapat berupa berbagai macam *file* yang tersimpan di dalam storage khusus yang digunakan untuk kepentingan suatu organisasi dan dapat diakses ataupun digunakan oleh siapapun yang berhak dan juga membutuhkannya.

2.2. Teori Judul

2.2.1 Pengertian Penerapan

Penerapan adalah suatu perbuatan mempraktekkan suatu teori, metode, dan hal lain untuk mencapai tujuan tertentu dan untuk suatu kepentingan yang diinginkan oleh suatu kelompok atau golongan yang telah terencana dan tersusun sebelumnya [7].

Berdasarkan dari pengertian di atas, penulis menyimpulkan bahwa kata penerapan (implementasi) bermuara pada aktifitas, adanya aksi dan tindakan, serta mekanisme suatu system. Ungkapan mekanisme mengandung arti bahwa penerapan (implementasi) bukan sekedar aktifitas, tetapi suatu kegiatan yang terencana dan dilakukan secara sungguh-sungguh berdasarkan acuan norma tertentu untuk mencapai tujuan kegiatan.

2.2.2 Pengertian Pembuatan

Pembuatan adalah kegiatan menciptakan atau memproses sesuatu. Kegiatan ini bertujuan untuk menciptakan sesuatu dengan beberapa cara atau langkah yang sesuai dengan hal yang akan dibuat [8]. Pendapat lain menyatakan pembuatan adalah tindakan yang terencana guna mendapatkan bentuk dan model yang dapat digunakan [9].



Dari definisi di atas, penulis menyimpulkan bahwa Pembuatan adalah suatu proses untuk menghasilkan atau menciptakan sesuatu dengan menggunakan langkah-langkah yang sesuai dengan hal yang akan dibuat.

2.2.3 Pengertian Aplikasi

Aplikasi merupakan penerapan, menyimpan sesuatu hal, data, permasalahan, dan pekerjaan ke dalam suatu sarana atau media yang dapat digunakan untuk diterapkan menjadi sebuah bentuk yang baru [10].

Aplikasi berasal dari kata *Aplication*, *Aplication* adalah masalah yang memakai teknik pemrosesan data aplikasi biasanya mengacu pada komputasi yang diinginkan, atau pemrosesan data [11].

Jadi aplikasi merupakan sebuah transformasi dari sebuah permasalahan atau pekerjaan berupa hal yang sulit difahami menjadi lebih sederhana, mudah dan dapat dimengerti oleh pengguna. Sehingga dengan adanya aplikasi, sebuah permasalahan akan terbantu lebih cepat dan tepat.

2.2.4 Pengertian Penerima

Penerima adalah orang yang menerima, atau pengertian lainnya dari penerima adalah alat yang berfungsi untuk menerima [12].

2.2.5 Pengertian Beasiswa

Beasiswa adalah faktor pendukung yang diberikan kepada pelajar atau mahasiswa sebagai bantuan biaya belajar. Beasiswa dapat juga menjadi faktor yang mendukung motivasi belajar para pelajar atau mahasiswa [13].

Beasiswa adalah pemberian berupa bantuan yang diberikan kepada perorangan yang bertujuan untuk digunakan demi keberlangsungan pendidikan yang ditempuh [14].

Dari definisi di atas, penulis menyimpulkan bahwa Beasiswa adalah sebuah bantuan di bidang pendidikan yang diberikan kepada pelajar



atau mahasiswa untuk menjadi faktor motivasi belajar dan demi keberlangsungan pendidikan yang ditempuh.

2.2.6 Pengertian Website

Website merupakan media informasi yang dapat di akses oleh siapa pun dalam suatu jaringan baik yang terhubung ke internet maupun tidak. Pada dasarnya *Website* merupakan suatu kumpulan *hyperlink* yang menuju dari alamat satu ke alamat lainnya dengan bahasa *HTML* (Hyper Text Markup Language) dan merupakan layanan yang banyak dimanfaatkan di internet [15].

Website merupakan kumpulan komponen terdiri dari teks, gambar, suara animasi sehingga merupakan media informasi yang menarik dan sangat dimininati untuk dipergunakan sebagai media berbagi informasi. Teknologi *Website* mengolah data menjadi sebuah informasi dengan cara mengidentifikasi, mengumpulkan, mengelola dan menyediakan untuk dapat diakses secara bersama-sama [16].

Dari definisi di atas penulis menyimpulkan bahwa *Website* adalah merupakan sekumpulan komponen yang terdiri dari teks, gambar, dan suara animasi untuk dipergunakan sebagai media informasi yang dapat diakses oleh semua orang dalam suatu jaringan baik yang terhubung ke internet.

2.2.7 Pengertian Metode Logika Fuzzy

Logika *Fuzzy* pertama kali diperkenalkan oleh Prof. Lotfi A. Zadeh, ilmuwan sistem teori dari Universitas Berkeley pada tahun 1965. Logika *Fuzzy* adalah logika yang dapat digunakan untuk menganalisis masalah yang mengandung ketidakpastian, salah satu contohnya proses prediksi [17].



2.2.8 Pengertian Penerapan Metode Logika *Fuzzy* untuk Pembuatan Aplikasi Penerima Beasiswa di Politeknik Negeri Sriwijaya Berbasis *Website*

Penerapan Metode Logika *Fuzzy* untuk Pembuatan Aplikasi Penerima Beasiswa di Politeknik Negeri Sriwijaya Berbasis *Website* adalah program komputer berbentuk *website* yang dibangun dengan manfaat untuk membantu bidang Akademik dan Kemahasiswaan dalam melakukan penyimpanan dan pengolahan data mahasiswa yang mengajukan beasiswa yang nantinya akan menghasilkan sebuah *ouput* data-data mahasiswa yang menerima beasiswa di Politeknik Negeri Sriwijaya dengan menggunakan Metode Logika *Fuzzy* ketika melakukan analisis masalah.

2.3 Teori Khusus

2.3.1 Pengertian Diagram Konteks

Diagram Konteks adalah diagram yang berada level yang paling tinggi yaitu level nol yang menggambarkan ruang lingkup sistem yang global [18]. Dengan adanya Data Flow Diagram (DFD), maka pemakai sistem yang kurang memahami di bidang computer dapat mengerti sistem yang sedang berjalan. Didalam DFD terdapat 3 (tiga) level, yaitu [19]:

a. 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 penyimpangan data dan tampak seederhana untuk diciptakan.

b. 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. Di dalam diagram ini memuat penyimpanan data.

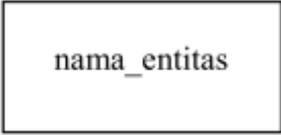
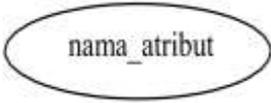
c. Diagram Rinci

Merupakan diagram yang menguraikan proses apa yang ada dalam diagram nol.

2.3.2 Pengertian *Entity Relationship Diagram* (ERD)

Pemodelan basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram* (ERD) [20]. ERD digunakan untuk pemodelan basis data relasional dan berikut ini adalah simbol-simbol yang digunakan pada *Entity Relationship Diagram* [20] :

Tabel 2.1 Simbol-simbol pada ERD

No.	Simbol	Deskripsi
1.	Entitas / <i>Entity</i> 	Entitas merupakan data inti yang akan disimpan, bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi <i>computer</i> ; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel
2.	Atribut / <i>Attribute</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas
3.	Atribut Kunci Primer	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i>



Lanjutan Tabel 2.1 Simbol-simbol pada ERD

No.	Simbol	Deskripsi
		yang diinginkan, biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda dengan tanpa ada yang sama)
4.	Atribut Multi Nilai 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu
5.	Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja
6.	Asosiasi 	Penghubung antara relasi dan entitas dimana diantara kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian

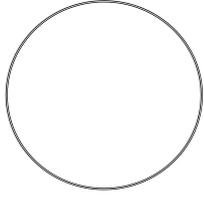
Sumber : Sukanto dan Shalahuddin (2018:50-51)

2.3.3 Pengertian Data Flow Diagram (DFD)

Data Flow Diagram (DFD) atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (input) dan keluaran (output) [18]. Berikut ini adalah simbol-simbol yang digunakan pada *Data Flow Diagram* [20] :



Tabel 2.2 Simbol-Simbol dalam *Data Flow Diagram* (DFD)

No	Notasi	Keterangan
1.		<p>Proses atau fungsi atau prosedur : pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program. Catatan nama yang diberikan pada sebuah proses biasanya berupa kata kerja</p>
2.		<p>File atau basis data atau penyimpanan (storage); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada basis data (ERD). Catatan: nama yang diberikan pada sebuah penyimpanan biasanya kata benda.</p>
3.		<p>Entitas luar (external entity) atau masukan (input) atau keluaran (output) atau orang yang akan chart [sic! Memakain/] berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan.</p>



Lanjutan Tabel 2.2 Simbol-Simbol dalam *Data Flow Diagram* (DFD)

No	Notasi	Keterangan
		Catatan: nama yang digunakan pada (input) atau keluaran (output) biasanya berupa kata benda.
4.		Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses atau dari proses ke (input) atau keluaran (output) Catatan: nama yang digunakan biasanya berupa kata benda, dapat diawali dengan kata data atau tanpa kata data

Sumber: Sukanto dan Shalahuddin (2018:71-72)

2.3.4 Pengertian Pemodelan Data

Pemodelan Data adalah kumpulan daftar elemen data yang mengalir pada *system* perangkat lunak sehingga masukan (input) dan keluaran (output) dapat dipahami secara umum (memiliki standar cara penulisan) [20]. Berikut ini simbol-simbol yang di gunakan dalam pemodelan data, yaitu :

Tabel 2.3 Simbol-simbol pada Pemodelan Data

No	Simbol	Keterangan
1	=	Disusun atau terdiri dari
2	+	Dan
3	()	Data opsional
4	[]	Baik ... atau ...
5	*...*	Batas komentar
6	{ }"	N kali diulang / bernilai banyak

Sumber: Sukanto dan Shalahuddin (2018:74)



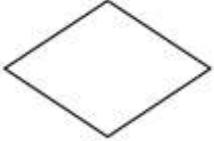
2.3.5 Pengertian *Flowchart*

Bagan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur *system* secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. Pada waktu akan menggambar suatu bagan alir, analisis *system* atau pemrograman dapat mengikuti pedoman-pedoman sebagai berikut ini [21]:

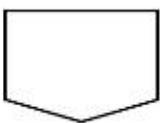
1. Bagan alir sebaiknya digambarkan dari atas kebawah dan mulai dari suatu halaman.
2. Kegiatan dalam bagan alir harus ditunjukkan dengan jelas.
3. Harus ditunjukkan dari mana kegiatan akan dimulai dan dimana berakhirnya.
4. Masing-masing kegiatan di dalam bagan alir sebaiknya digunakan suatu yang mewakili pekerjaan.
5. Masing-masing kegiatan dalam bagan alir harus di dalam urutan semestinya.
6. Kegiatan yang terpotong akan disambung di tempat lain harus ditunjukkan dengan jelas menggunakan symbol penghubung.
7. Gunakanlah simbol-simbol bagan alir yang standar.

Berikut ini adalah simbol-simbol yang biasa digunakan pada *Flowchart* [22]:

Tabel 2.4 Simbol-simbol pada *Flowchart*

No.	Simbol	Fungsi
1.		Permulaan sub program
2.		Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya

Lanjutan Tabel 2.4 Simbol-simbol pada *Flowchart*

No.	Simbol	Fungsi
3.		Penghubung bagian-bagian flowchart yang berada pada satu halaman
4.		Penghubung bagian-bagian flowchart yang berada pada halaman berbeda
5.		Permulaan/akhir program
6.		Arah aliran program
7.		Proses inialisasi/pemberian harga awal
8.		Proses penghitung/proses pengolahan data
9.		Proses input/output data

Sumber: Santoso dan Nurmalina (2018:86)



2.4 Teori Program

2.4.1 Pengertian *HTML*

HTML singkatan dari *Hypertext Markup Language* yaitu *script* berupa penandaan awal dan akhir untuk membuat dan mengatur struktur *website* [23].

Hypertext Markup Language (*HTML*) adalah sebuah text biasa dapat berfungsi lain, kita dapat membuatnya menjadi link yang dapat berpindah dari satu halaman ke halaman lainnya hanya dengan meng-klik text tersebut [24].

2.4.1.1. *Script HTML*

```
<html>
  <head>
    <title>Ini judul dokumen HTML </title>
  </head>
  <body>
    Teks ini adalah teks yang muncul di Body dari dokumen
  </body>
</html>
```

Gambar 2.1 *Script HTML*
Sumber : *Rerung (2018:18)*

2.4.2 Sekilas tentang *PHP*



Gambar 2.2 Logo *PHP*
Sumber : *(ITX Design)*



2.4.2.1. Pengertian PHP

PHP adalah bahasa pemrograman *script server side* yang dirancang untuk membuat dan mengembangkan sebuah *website*. Bahasa pemrograman ini dirancang untuk para pengembang *web* agar dapat menciptakan suatu halaman *web* yang bersifat dinamis [24].

2.4.2.2. Script PHP

```
<html>
<head>
<title>Contoh</title>
</head>
<?php
Echo "Hai, saya dari script PHP!";
?>
</body>
</html>
```

Gambar 2.3 Script PHP
Sumber : Rerung (2018:19)

2.4.3 Pengertian CSS

Cascading Style Sheets (CSS) adalah biasanya selalu dikaitkan dengan *HTML*, karena keduanya memang saling melengkapi dimana *HTML* ditujukan untuk membuat struktur, atau konten dari halaman *web* [24].



2.4.4 Sekilas tentang XAMPP



Gambar 2.4 Logo XAMPP
(Sumber : ITX Design)

2.4.4.1 Pengertian XAMPP

XAMPP adalah perangkat lunak bebas (free software) yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsi xampp sendiri sebagai server yang berdiri sendiri (localhost), yang terdiri dari beberapa program, antara lain: *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl* [25].

XAMPP adalah salah satu paket instalasi *apache*, *PHP*, dan *MySQL* secara instant yang dapat digunakan untuk membantu proses instalasi ketiga produk tersebut [25].

Berdasarkan pengertian diatas dapat disimpulkan bahwa XAMPP adalah sebuah aplikasi perangkat lunak yang di dalamnya sudah tersedia *database server MySQL* dan *support PHP programming*.

2.4.5 Sekilas tentang MySQL



Gambar 2.5 Logo MySQL
(Sumber : ITX Design)



2.4.5.1 Pengertian MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data *SQL* (Database Management System) atau *DBMS* yang *multithread*, *multi-user*, dengan sekitar 6 (enam) juta instalasi di seluruh dunia [26].

MySQL adalah sistem manajemen *database SQL* yang sifatnya *open source* (terbuka) dan paling banyak digunakan saat ini. Sistem *database MySQL* mampu mendukung beberapa fitur seperti *multithreaded*, *multi-user*, dan *SQL database management system (DBMS)* [27].

Dari uraian diatas, dapat disimpulkan bahwa *MySQL* adalah jenis basis data sistem yang berguna untuk mengolah *database* serta membangun aplikasi *web* dengan basis data sebagai sumber pengelolaan datanya.

2.4.6 Sekilas tentang Sublime Text



Gambar 2.6 Logo Sublime Text
(Sumber: www.MacUpdate.com)

2.4.6.1 Pengertian Sublime Text

Sublime text merupakan perangkat lunak *text editor* yang digunakan untuk membuat atau mengedit suatu aplikasi [28].

Sublime Text merupakan salah satu *text editor* yang sangat *powerfull* yang dapat meningkatkan produktivitas dan mengembangkan kualitas kode yang tinggi [29].

Jadi dapat disimpulkan bahwa *Sublime Text* adalah perangkat lunak berupa teks editor yang digunakan untuk mengembangkan kualitas kode *programmer*.



2.4.7 Sekilas tentang PHP MyAdmin



Gambar 2.7 tabel
(Sumber : ITX Design)

2.4.7.1 Pengertian PHP MyAdmin

PHP MyAdmin merupakan aplikasi berbasis *web* yang digunakan untuk membuat *database MySQL*, sebagai tempat untuk menyimpan data-data *website*, keduanya biasanya sudah disediakan dalam satu paket aplikasi seperti *Appserv* atau *Xampp* [30].

2.5 Metode Analisa Penelitian

2.5.1 Metode Logika Fuzzy

Logika *fuzzy* adalah perpanjangan dari logika Boolean oleh Lotfi Zadeh pada tahun 1965 berdasarkan pada teori matematika dari himpunan *fuzzy*, yang merupakan generalisasi teori himpunan klasik. Dengan memperkenalkan pengertian derajat dalam verifikasi suatu kondisi, sehingga memungkinkan suatu kondisi menjadi dalam keadaan selain benar atau salah, logika *fuzzy* memberikan nilai yang sangat berharga fleksibilitas untuk penalaran, yang memungkinkan untuk memperhitungkan ketidakakuratan dan ketidakpastian. Logika *fuzzy* telah digunakan dalam bidang manajemen transportasi, analisis sentimen, pencarian informasi, dan ekstraksi informasi. Logika *fuzzy* adalah modifikasi yang tepat dari konseptualisasi bersama dalam kategori tertentu, yang merupakan format yang dimengerti manusia dan dapat dibaca oleh mesin. Untuk spesifikasi lebih lanjut dari proses analisis sentimen, logika *fuzzy* dapat diperkenalkan. Oleh karena itu, analisis



sentimen dengan bantuan logika *fuzzy* yang berurusan dengan penalaran dan memberikan pandangan lebih dekat ke nilai sentimen yang tepat akan membantu pengklasifikasian sejarah ujaran seseorang [31].

Logika *fuzzy* membantu penelitian ini untuk memberi nilai kecenderungan sentimen seseorang berdasarkan sejarah ujarannya. Algoritma *fuzzy* mampu membuat kriteria berdasarkan himpunan data yang batasnya tidak tegas. Contohnya adalah ketika mayoritas ujaran seseorang nilai kecenderungan positif dan negatifnya tidak jelas yang disebabkan oleh banyaknya ujaran yang tidak bisa dinilai oleh teknik analisis sentimen atau bersifat netral. Sementara jumlah ujaran orang tersebut terus bertambah, sehingga batasan yang dibuat apabila kita menggunakan teknik perhitungan matematika sederhana hasilnya kurang maksimal. Alasan digunakannya *Fuzzy* antara lain [31]:

1. Konsep logika Fuzzy mudah dimengerti. Konsep matematis yang mendasari penalaran Fuzzy sangat sederhana dan mudah dimengerti.
2. Logika Fuzzy sangat fleksibel.
3. Logika Fuzzy memiliki toleransi terhadap data-data yang tidak tepat.
4. Logika Fuzzy mampu memodelkan fungsi-fungsi non-linear yang sangat kompleks.
5. Logika Fuzzy dapat membangun dan mengaplikasikan beberapa pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
6. Logika Fuzzy dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
7. Logika Fuzzy didasarkan pada bahasa alami.

2.5.2 Himpunan *Fuzzy*

Pada himpunan tegas di setiap elemen dalam semestanya akan selalu ditentukan secara tegas apakah elemen itu betul merupakan anggota himpunan tersebut atau bukan. Akan tetapi dalam kenyataannya hampir tidak semua himpunan akan terdefinisi secara tegas. Misalnya saja



himpunan mahasiswa cerdas, dalam contoh ini tidak bisa dinyatakan dengan tegas karena kita lihat tidak ada yang dijadikan ukuran pasti untuk tingkat kecerdasan dari seseorang. Oleh karena itu perlu didefinisikan suatu himpunan *fuzzy* yang bisa menyatakan kejadian tersebut [32]. Himpunan *fuzzy* memiliki 2 (dua) atribut, yaitu [32]:

1. Linguistik, yaitu penamaan suatu kelompok yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti: JAUH, SEDANG, DEKAT.
2. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel, seperti: 10, 40, 80, dan sebagainya.

Beberapa hal yang perlu diketahui dalam memahami sistem *fuzzy*, yaitu:

a. Variabel *Fuzzy*

Variabel *fuzzy* merupakan variabel yang akan dibahas dalam suatu sistem *fuzzy*, seperti: tingkat kecenderungan, potensi ironi, dan sebagainya.

b. Himpunan *Fuzzy*

Himpunan *fuzzy* merupakan suatu kelompok yang mewakili suatu keadaan tertentu dalam suatu variabel *fuzzy*.

c. Semesta pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif

d. Domain

Domain himpunan *fuzzy* adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu himpunan *fuzzy*. Nilai domain dapat berupa bilangan positif maupun negatif.

2.5.3 Langkah-Langkah Penyelesaian *Fuzzy*

Dalam melakukan penilaian menggunakan *Fuzzy* terdapat beberapa langkah atau tahapan, langkah-langkahnya adalah sebagai berikut [33]:



a. Fuzzifikasi Input

Fuzzifikasi input adalah proses pertama dalam sistem *Fuzzy* adalah mentransformasi nilai *crisp* (mentah) menjadi nilai keanggotaan melalui fungsi keanggotaan. Hal ini berarti bahwa fungsi keanggotaan untuk setiap himpunan *Fuzzy* harus ditentukan terlebih dahulu. Hasil dari tahap pertama ini adalah mentransformasi semua *antecedent* atau nilai *crisp* (mentah) ke dalam derajat keanggotaan yang memiliki nilai antara 0 (nol) dan 1 (satu) yang didefinisikan secara matematis oleh persamaan :

$$\mu_A(x): X \rightarrow [0,1] \quad \dots\dots\dots(P1)$$

Setiap elemen x dipetakan pada sebuah nilai keanggotaan oleh MF. Nilai dibawah ini merupakan derajat keanggotaan dari x pada himpunan *Fuzzy* A.

$$\mu_A(x) = Degree(x \in A) \quad \dots\dots\dots(P2)$$

Dimana nilai keanggotaan dari x dibatasi oleh:

$$0 \leq \mu_A(x) \leq 1 \quad \dots\dots\dots(P3)$$

$$\mu_A(x) = \begin{cases} 1, & \\ \frac{x-a}{b-a}, \frac{c-x}{c-b} & \\ 0 & \end{cases} \quad \dots\dots\dots(P4)$$

Keterangan :

$\mu[x]$ = Nilai keanggotaan pada himpunan *fuzzy* (cara mencari nilai keanggotaan ini juga bisa dengan dua cara yaitu atau , tergantung kondisi inputan *crisp* (mentah). Kedua cara ini sama-sama digunakan dalam menghasilkan nilai keanggotaan, sehingga satu nilai inputan *crisp* (mentah) bisa memiliki satu atau dua nilai keanggotaan, tergantung kondisi inputan *crisp* (mentah) tersebut.

x = Nilai inputan *crisp*

a = Nilai batas tengah / bawah *linguistic term* yang mendekati inputan *crisp*

b = Nilai batas atas *linguistic term* yang mendekati inputan *crisp*



c = Nilai batas atas *linguistic term* yang mendekati inputan *crisp*

b = Nilai batas tengah / bawah *linguistic term* yang mendekati inputan *crisp* [33].

b. Operator Fuzzy

Jika terdapat dua atau lebih premis pada setiap *rule* yang terlibat maka harus digunakan operasi *Fuzzy* untuk setiap premise pada *rule* tersebut. Operasi *Fuzzy* ini pada dasarnya mirip dengan operasi pada himpunan klasik. Operasi tersebut antara lain seperti operasi AND yang diganti dengan *min*, operasi OR yang diganti dengan *max*, dan operasi NOT yang diganti dengan *komplemen* suatu bilangan [33].

1. Operator AND

Penulisan operator AND biasanya diganti dengan menggunakan *min*. Operator AND antara dua buah himpunan *Fuzzy* A dan B akan menghasilkan interseksi antara A dan B pada X yang didefinisikan sebagai berikut [33]:

$$\begin{aligned}\mu_{A \cap B}(X) &= \min(\mu_A(x), \mu_B(x)) \text{ untuk semua } x \in X \\ &= \min(\mu_A(x), \mu_B(x)) \\ &= \mu_A(x) \cap \mu_B(x). \dots\dots\dots(P6)\end{aligned}$$

2. Operator NOT

Penulisan operator NOT biasanya diganti dengan menggunakan *komplemen* suatu bilangan. Operator NOT pada himpunan *Fuzzy* A akan memberikan hasil *komplemen* dari A, yaitu [33]:

$$\mu_{\sim A}(x) = 1 - \mu_A(x) \dots\dots\dots(P7)$$

c. Inferensi Fuzzy (Implikasi)

Proses selanjutnya adalah proses inferensi (implikasi) jika diketahui fakta (premise) untuk menghasilkan konklusi. Tahap ini menghasilkan suatu keputusan yang didapat dari *rule Fuzzy*. Proses implikasi ini juga diterapkan



untuk menghasilkan nilai keluaran (inferensi). Bagian *consequent* dari *IfThen rule* ini memetakan semua himpunan *Fuzzy* pada keluaran [33].

d. Agregasi semua keluaran

Agregasi adalah tahap memasukkan nilai crisp ke beberapa rule *Fuzzy*, sehingga nantinya didapatkan suatu luasan yang menunjukkan hasil akhir agregasi. Agregasi ini dapat dilakukan dengan menggabungkan beberapa fungsi keanggotaan pada keluaran sistem *Fuzzy* yang memiliki nilai keanggotaan maksimum (disebut metode agregasi Max). Atau alternatif kedua dengan cara menjumlahkan semua fungsi keanggotaan pada keluaran sistem *Fuzzy* (disebut metode agregasi Sum) [33].

e. Defuzzifikasi

Proses terakhir yang harus dilakukan dari sebuah sistem *Fuzzy* adalah proses defuzzifikasi, yaitu proses untuk mentransformasikan kembali dari himpunan *Fuzzy* pada bagian konklusi menjadi sebuah bilangan keluaran (crisp output). Proses defuzzifikasi dilakukan dengan berbagai macam metode. Misalnya, *maximum defuzzification*, *centroid* (center of gravity) *defuzzification*, *weighted average defuzzification*, dan lain-lain. Berfokus pada *centroid* (center of gravity) *defuzzification*, yaitu metode defuzzifikasi yang paling umum digunakan dan memberikan hasil yang sangat akurat. Metode ini dirumuskan sebagai berikut [33]:

$$x^* = \frac{\sum_{i=1}^N x_i \mu_A(x_i)}{\sum_{i=1}^N \mu_A(x_i)} \dots\dots\dots (P8)$$

