

BAB II

LANDASAN TEORI

Guna memudahkan pemahaman dalam penyusunan skripsi, penulis perlu kiranya mengadakan studi keputusan mengenai arti dan istilah-istilah yang digunakan dalam penelitian skripsi sehingga dapat memecahkan suatu masalah.

2.1. Sistem Pendukung Keputusan

Menurut Raymond McLeod, sistem pendukung keputusan merupakan sebuah sistem yang menyediakan kemampuan untuk penyelesaian masalah dan komunikasi untuk permasalahan yang bersifat semi-terstruktur. Menurut Moore and Chang, SPK dapat digambarkan sebagai sistem yang berkemampuan mendukung analisis ad hoc data, dan pemodelan keputusan, berorientasi keputusan, orientasi perencanaan masa depan, dan digunakan pada saat-saat yang tidak biasa. Jadi sistem pendukung keputusan adalah sistem yang dipakai untuk mendukung pengambilan keputusan dalam menyelesaikan suatu masalah agar masalah yang ada dapat diselesaikan dengan baik. SPK adalah sistem interaktif berbasis computer yang membantu untuk mengambil keputusan memanfaatkan data dan model untuk memecahkan masalah yang tidak terstruktur (Ardiles Sinaga, 2012).

Sistem Pendukung Keputusan (SPK) dapat didefinisikan sebagai suatu program computer yang menyediakan informasi dalam domain aplikasi

yang diberikan oleh suatu model analisis keputusan dan akses ke database, dimana hal ini ditunjukkan untuk mendukung pembuatan keputusan (*decision maker*) dalam mengambil keputusan secara efektif baik dalam kondisi yang kompleks dan tidak teratur. (Kusrini, 2007)

Ada beberapa tahapan untuk mengambil keputusan menurut Herbert A. Simon, berikut tahapan yang harus dilalui dalam proses pengambilan keputusan :

1. Tahap Pemahaman (*Intelligence phase*)

Tahap ini merupakan proses penelusuran dan pendeteksian dari lingkungan problematika serta proses pengenalan masalah. Data masukan diperoleh, diproses dan diuji dalam rangka mengidentifikasi masalah.

2. Tahap Perancangan (*Design Phase*)

Tahapan ini merupakan proses pengembangan dan pencarian alternatif tindakan / solusi yang dapat diambil. Tersebut merupakan representasi kejadian nyata yang disederhanakan, sehingga diperlukan proses validasi dan verifikasi untuk mengetahui keakuratan model dalam meneliti masalah yang ada.

3. Tahap Pemilihan (*Choice Phase*)

Tahap ini dilakukan pemilihan terhadap berbagai alternative solusi yang muncul pada tahap perancangan agar ditentukan / dengan memperhatikan kriteria – kriteria berdasarkan tujuan yang akan dicapai.

4. Tahap Implementasi (*Implementation Phace*)

Tahap ini dilakukan penerapan terhadap rancangan sistem yang telah dibuat pada tahap perancangan serta pelaksanaan alternative tindakan yang telah dipilih pada tahap pemilihan.

2.2. Produksi

Menurut Rianto dan Euis(2010) kegiatan produksi merupakan salah satu aktivitas ekonomi yang sangat menunjang selain kegiatan konsumsi. Tanpa adanya kegiatan produksi, konsumen tidak dapat mengonsumsi barang dan jasa yang dibutuhkannya. Kegiatan produksi dan kegiatan konsumsi adalah satu mata rantai yang saling berkaitan dan tidak bisa saling dilepaskan.

Dalam kehidupan sehari-hari, apabila kita mendengar kata produksi, maka yang terbayang dipikiran kita adalah suatu kegiatan besar yang memerlukan peralatan yang canggih dan menggunakan ribuan tenaga kerja untuk mengerjakannya. Sebenarnya perkiraan tersebut tidak benar. Produksi adalah kegiatan menambah nilai guna suatu barang atau jasa untuk keperluan orang banyak, jadi tidak semua kegiatan yang menambah nilai guna suatu barang dapat dikatakan proses produksi. Contohnya : seorang ibu yang membuat kue untuk keluarganya di rumah, kegiatan tersebut tidak dapat dikatakan proses produksi karena tujuannya bukan untuk masyarakat banyak.

Pada saat kebutuhan manusia masih sedikit dan masih sederhana, kegiatan produksi dan konsumsi seringkali dilakukan sendiri, yaitu seseorang memproduksi untuk memenuhi kebutuhannya sendiri. Namun, seiring dengan semakin beragamnya kebutuhan dan keterbatasan sumber daya, sehingga seseorang tidak dapat lagi memproduksi sendiri barang dan jasa yang dibutuhkannya, sehingga ia membutuhkan pihak lain untuk memproduksi apa yang menjadi kebutuhannya tersebut. Secara teknis produksi dapat diartikan sebagai suatu proses mentransformasi input menjadi output, tetapi pengertian produksi dalam ilmu ekonomi mencakup tujuan kegiatan menghasilkan output serta karakter yang melekat padanya.

2.3. Prediksi

Prediksi adalah suatu proses memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi masa lalu dan sekarang yang dimiliki, agar kesalahannya (selisih antara sesuatu yang terjadi dengan hasil perkiraan) dapat diperkecil. Prediksi tidak harus memberikan jawaban secara pasti kejadian yang akan terjadi, melainkan berusaha untuk mencari jawaban sedekat mungkin yang akan terjadi. (Ismail, 2014)

2.4. Algoritma Fuzzy

Logika *fuzzy* yang pertama kali diperkenalkan oleh Lutfi A. Zadeh, memiliki derajat keanggotaan dalam rentang 0 (nol) hingga 1 (satu), berbeda dengan logika digital yang hanya memiliki dua nilai yaitu 1 (satu) atau 0 (nol). Logika *fuzzy* digunakan untuk menerjemahkan suatu besaran yang diekspresikan menggunakan bahasa (linguistic), misalkan besaran kecepatan laju kendaraan yang diekspresikan dengan pelan, agak cepat, cepat dan sangat cepat. Secara umum dalam sistem logika *fuzzy* terdapat empat buah elemen dasar, yaitu :

1. Basis kaidah (*rule base*), yang berisi aturan – aturan secara linguistik yaitu bersumber dari para pakar.
2. Suatu mekanisme pengambilan keputusan (*inference engine*), yang memperagakan bagaimana para pakar mengambil suatu keputusan dengan menerapkan pengetahuan (*knowledge*).
3. Proses fuzzifikasi (*fuzzification*), yang mengubah besaran tegas (*crisp*) ke besaran *fuzzy*.
4. Proses defuzzifikasi (*defuzzification*), yang mengubah besaran *fuzzy* hasil dari inference engine, menjadi besaran tegas (*crisp*).

Logika *fuzzy* adalah metodologi sistem kontrol pemecahan masalah, yang cocok untuk diimplementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, embedded system, jaringan PC, multichannel atau workstation berbasis akuisisi data, dan sistem kontrol. Metodologi ini dapat diterapkan pada perangkat keras, perangkat lunak, atau kombinasi

keduanya. Dalam logika klasik dinyatakan bahwa segala sesuatu bersifat biner, yang artinya adalah hanya mempunyai dua kemungkinan, “Ya atau Tidak”, “Benar atau Salah”, “Baik atau Buruk”, dan lain - lain. Oleh karena itu, semua ini dapat mempunyai nilai keanggotaan 0 atau 1. Akan tetapi, dalam logika *fuzzy* kemungkinan nilai keanggotaan berada diantara 0 dan 1. Artinya, bisa saja suatu keadaan mempunyai dua nilai “Ya dan Tidak”, “Benar dan Salah”, “Baik dan Buruk” secara bersamaan, namun besar nilainya tergantung pada bobot keanggotaan yang dimilikinya. (Sutoyo, dkk, 2011)

2.5. Himpunan *Fuzzy*

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu item x dalam suatu himpunan A , yang sering ditulis dengan $A[x]$, memiliki 2 kemungkinan (Kusumadewi S, Purnomo H, 2010) yaitu :

1. Satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan.
2. Nol (0), yang berarti suatu item tidak menjadi anggota dalam suatu himpunan.

Terkadang kemiripan antara keanggotaan *fuzzy* dengan probabilitas menimbulkan kerancuan. Kedua memiliki nilai pada interval $[0,1]$, namun interpretasi nilainya sangat berbeda antara kedua kasus tersebut. Keanggotaan *fuzzy* memberikan suatu ukuran terhadap pendapat atau keputusan, sedangkan probabilitas mengindikasikan proporsi terhadap keseragaman suatu hasil bernilai benar dalam jangka panjang. Misalnya, jika

nilai keanggotaan bernilai suatu himpunan *fuzzy* USIA adalah 0,9 maka tidak perlu dipermasalahkan berapa seringnya nilai itu diulang secara individual untuk mengharapkan suatu hasil yang hamper pasti muda. Di lain pihak, nilai propabilitas 0,9 usia berarti 10% dari himpunan tersebut diharapkan tidak mudah. Himpunan *fuzzy* memiliki 2 atribut, yaitu :

1. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti : MUDA, PAROBAYA, TUA
2. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti : 40, 25, 50, dsb.

Dalam buku Aplikasi Logika *Fuzzy* untuk pendukung keputusan (Kusumadewi S dan Purnomo H, 2010). Ada beberapa hal yang perlu diketahui dalam memahami sistem *fuzzy*, yaitu :

a. Variabel *Fuzzy*

Variabel *fuzzy* merupakan variabel yang hendak dibahas dalam suatu sistem *fuzzy*. Contoh : umur, temperature, permintaan, dsb.

b. Himpunan *Fuzzy*

Himpunan *fuzzy* merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*.

c. Semesta Pembicaraan atau Domain

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*. Semesta pembicaraan merupakan himpunan bilangan *real* yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan.

Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Ada kalanya nilai semesta pembicaraan ini tidak dibatasi batas atanya.

2.6. Fungsi Keanggotaan *Fuzzy*

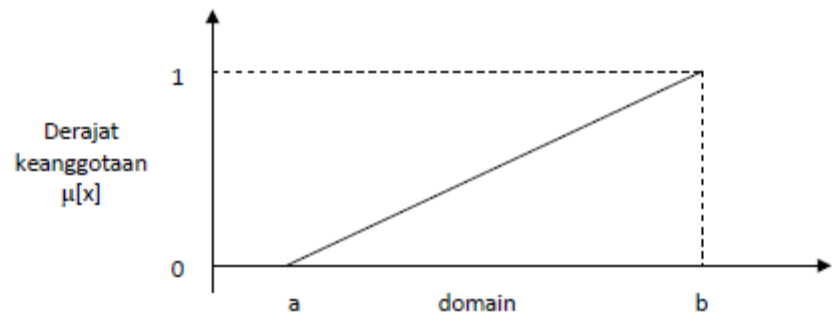
Fungsi derajat keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik – titik input data ke dalam derajat keanggotaan yang memiliki interval antara 0 sampai 1 (Kusumadewi dan Purnomo, 2009).

Menurut Kusumadewi dan Purnomo (2010) suatu fungsi derajat keanggotaan *fuzzy* memiliki beberapa fungsi yang dapat digunakan :

a. Representasi Linear

Pada representasi *linear*, pemetaan input ke derajat keanggotaan digambar sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas.

Ada 2 keadaan himpunan *fuzzy* yang *linear*. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol (0) bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi (Kusumadewi dan Purnomo, 2010).

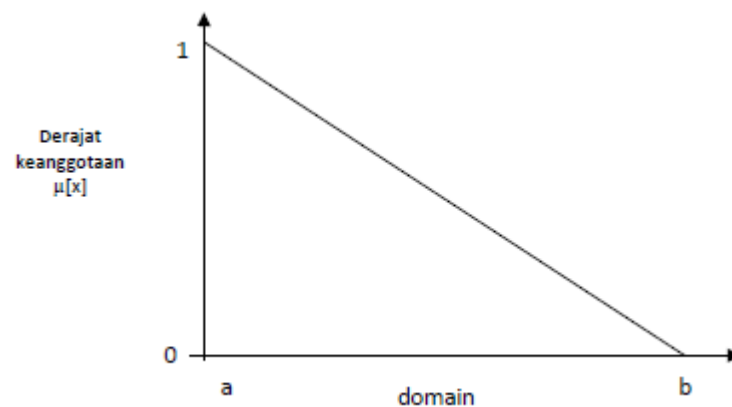


Gambar 2. 1 Representasi Linear Naik

Fungsi keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq a \\ \frac{x-a}{b-a}; & a < x < b \\ 1; & x \geq b \end{cases}$$

Kedua, merupakan kebalikan dari yang pertama. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah.



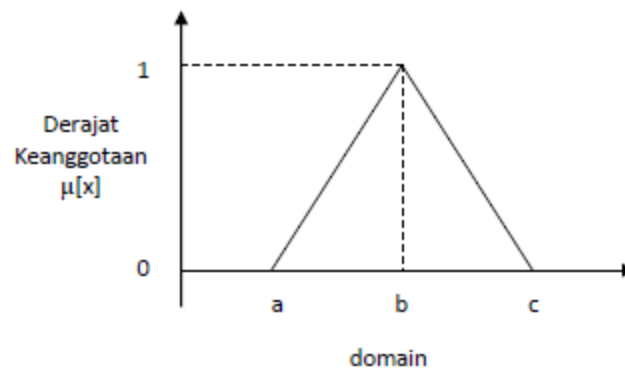
Gambar 2. 2 Representasi Linear Turun

Fungsi keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \geq b \\ \frac{b-x}{b-a}; & a < x < b \\ 1; & x \leq a \end{cases}$$

b. Representasi kurva segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (*linear*).



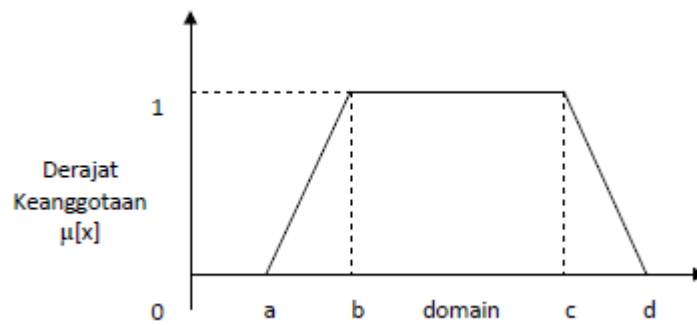
Gambar 2. 3 Representasi Kurva Segitiga

Fungsi keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \geq c \text{ atau } x \leq a \\ \frac{x-a}{b-a}; & a < x < b \\ \frac{c-x}{c-b}; & b < x < c \end{cases}$$

c. Representasi kurva trapesium

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada titik yang memiliki nilai keanggotaan 1.



Gambar 2. 4 Representasi Kurva Trapesium

Fungsi keanggotaan:

$$f(x) = \begin{cases} 0; & x \geq d \text{ atau } x \leq a \\ \frac{(x-a)}{b-a}; & a < x < b \\ \frac{d-x}{d-c}; & c < x < d \\ 1; & b \leq x \leq c \end{cases}$$

2.7. Operasi Himpunan *Fuzzy*

Operasi himpunan fuzzy diperlukan untuk proses inferensi atau penalaran. Dalam hal ini yang diperoleh adalah derajat keanggotaannya. Derajat keanggotaan sebagai hasil dari operasi dua himpunan *fuzzy* disebut dengan *fire strength* atau α -predikat. Operasi yang sering digunakan untuk mengkombinasi dan memodifikasi himpunan *fuzzy* sebagai berikut (Kusumadewi dan Purnomo, 2009):

1. Operasi Gabungan (*Union*)

Operasi gabungan sering disebut operator OR pada himpunan A dan B yang dinyatakan sebagai $A \cup B$. dalam logika *fuzzy*, operasi

gabungan disebut sebagai *Max*. Operasi *Max* ditulis dengan persamaan :

$$\mu_{A \cup B}(x) = \max.\{\mu_A(x), \mu_B(x)\}$$

Derajat keanggotaan setiap unsure himpunan *fuzzy* $A \cup B$ adalah derajat keanggotaannya pada himpunan *fuzzy* A atau B yang memiliki nilai terbesar.

2. Operasi Irisan (*Intersection*)

Operasi irisan sering disebut operator AND pada himpunan A dan B yang dinyatakan sebagai $A \cap B$. Dalam logika *fuzzy*, operasi irisan disebut dengan *Min*. Operasi *Min* ditulis dengan persamaan :

$$\mu_{A \cap B}(x) = \min.\{\mu_A(x), \mu_B(x)\}$$

Derajat keanggotaan setiap unsure himpunan *fuzzy* $A \cap B$ adalah derajat keanggotaannya pada himpunan *fuzzy* A atau B yang memiliki nilai terkecil.

3. Operator Komplemen (*Complement*)

Bila himpunan *fuzzy* A pada himpunan universal X mempunyai fungsi keanggotaan $\mu_A(x)$ maka komplemen dari himpunan *fuzzy* A (sering disebut NOT) adalah himpunan *fuzzy* A^C dengan fungsi keanggotaan untuk setiap x elemen X ,
 $\mu_{A^C}(x) = 1 - \mu_A(x)$.

2.8. Fungsi Implikasi

Dalam basis pengetahuan *fuzzy*, tiap tiap rule selalu berhubungan dengan relasi *fuzzy*. Dalam fungsi implikasi, digunakan bentuk

IF x is A THEN y is B

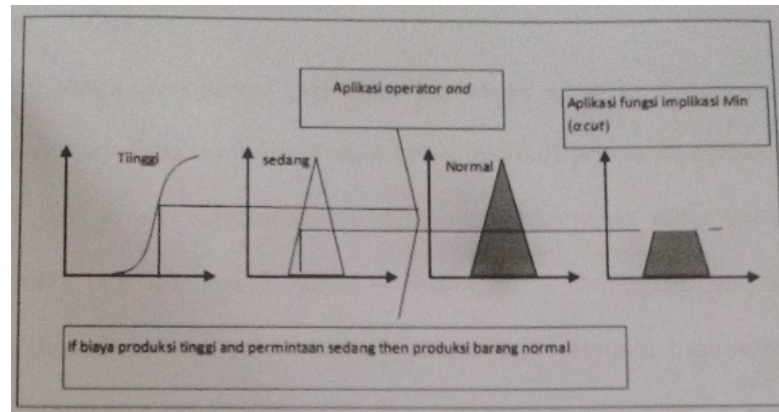
Dengan x dan y adalah skalar, A dan B adalah himpunan *fuzzy*. Aplikasi fungsi implikasi pada kontrol logika *fuzzy* merupakan bentuk relasi / implikasi “jika-maka” atau “*if-then*”. Proposisi yang mengikuti “*if*” disebut anteseden sedangkan proposisi yang mengikuti “*then*” disebut konsekuen. Proposisi ini dapat diperluas dengan penghubung *fuzzy*. Secara umum dapat dituliskan $IF (T_1 \text{ is } t_1) * (T_2 \text{ is } t_2) * \dots * (T_n \text{ is } t_n) * THEN (U_1 \text{ is } u_1) * (U_2 \text{ is } u_2) * \dots * (U_n \text{ is } u_n)$, dengan tanda (*) adalah suatu operator OR atau AND, seperti pada pernyataan berikut :

“JIKA” X = A DAN “JIKA” Y = B “MAKA” Z = C

Menurut Kusumadewi jika suatu proposisi menggunakan bentuk terkondisi maka ada dua fungsi implikasi secara umum yang dapat digunakan, yaitu :

a. Metode Minimum (α -cut)

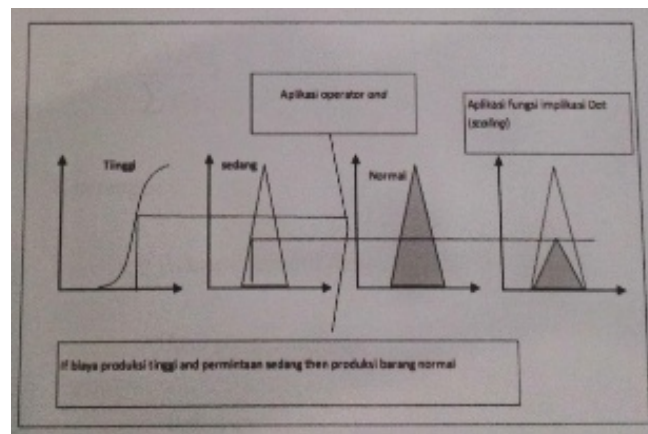
Metode ini akan memotong output himpunan *fuzzy*. Penggambaran metode minimum ditunjukkan pada gambar dibawah ini.



Gambar 2. 5 Metode Minimum (α -cut)

b. Metode Dot (*scaling*)

Metode ini akan menskala output himpunan *fuzzy*. Penggambaran metode dot ditunjukkan pada gambar dibawah ini.



Gambar 2. 6 Metode Dot (*scaling*)

2.9. Defuzzifikasi

Defuzzifikasi merupakan proses pemetaan himpunan *fuzzy* ke himpunan tegas (*crisp*). Proses ini merupakan kebalikan dari proses *fuzzyfikasi*. Input dari proses *defuzzifikasi* adalah suatu himpunan *fuzzy* yang

diperoleh dari komposisi aturan – aturan *fuzzy*, sedangkan output yang dihasilkan merupakan suatu bilangan pada domain himpunan *fuzzy* tersebut. Sehingga jika diberikan suatu himpunan *fuzzy* dalam range tertentu, maka harus dapat diambil suatu nilai *crisp* tertentu sebagai output.

Banyak cara untuk melakukan *defuzzifikasi* diantaranya metode berikut ini :

- a. Metode rata – rata (*Average*).

$$Z = \frac{\sum \alpha_i x_i}{\sum \alpha_i}$$

Keterangan :

Z = Keluaran kontrol *fuzzy*

x = Hasil penalaran *fuzzy*

α = Hasil α -pedikat

- b. Metode Centroid.

Pada metode ini penetapan nilai *crisp* dengan cara mengambil titik pusat daerah *fuzzy*.

$$Z^* = \frac{\int_z z \mu(z) dz}{\int_z \mu(z) dz}$$

- c. Metode Bisektor.

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai pada domain *fuzzy* yang memiliki nilai keanggotaan seperti dari jumlah total nilai keanggotaan pada daerah *fuzzy*.

- d. Metode Means of Maximum (MOM).

Pada metode ini, solusi crisp diperoleh dengan cara mengambil nilai rata – rata domain yang memiliki nilai keanggotaan maksimum.

2.10. *Fuzzy Tsukamoto*

Menurut Kusumadewi dan Purnomo (2004) *Fuzzy Tsukamoto* merupakan salah satu metode dari *Fuzzy Inference System*. Dimana pada metode *tsukamoto*, setiap konsekuen pada aturan berbentuk *if-then* harus direpresnetasikan dengan suatu himpunan *fuzzy* dengan fungsi keanggotaan yang monoton.

- a. *Fuzzifikasi* adalah proses yang dilakukan untuk mengubah variabel nyata menjadi variabel *fuzzy*, ini ditujukan agar masukan kontroler *fuzzy* bisa dipetakan menuju jenis yang sesuai dengan himpunan *fuzzy*. Pemetaan dilakukan dengan bantuan model dari fungsi keanggotaan agar dapat diketahui besar masukan tersebut (derajat keanggotaan). Terdapat beberapa jenis penggambaran fungsi keanggotaan, yaitu Gaussian, Segitiga, Trapesium dan lainnya.
- b. *Fuzzy Inferensi System* adalah proses penggabungan banyak aturan berdasarkan data yang tersedia. Komponen yang melakukan inferensi dalam sistem pakar disebut mesin inferensi. Menurut Srei Kusumadewi dan Sri Hartati (2006:34) sistem inferensi *fuzzy* merupakan suatu kerangka komputasi yang

berdasarkan pada teori himpunan *fuzzy*, aturan *fuzzy* yang berbentuk IF-THEN, dan penalaran *fuzzy*.

- c. *Defuzzifikasi* adalah proses yang digunakan untuk mengubah kembali variabel *fuzzy* menjadi variabel nyata, atau dengan kata lain aksi kontrol *fuzzy* yang masih berupa himpunan, dirubah menjadi nilai nyata yang berupa nilai tunggal. Banyak metode yang dapat digunakan untuk defuzzifikasi, salah satu contohnya adalah metode center of gravity.



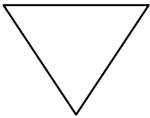

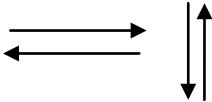
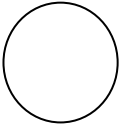
Ada beberapa alasan mengapa orang menggunakan logika *fuzzy* (Kusumadewi S, Purnomo H, 2010) antara lain:

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 nonlinear yang sangat kompleks.
5. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
6. Logika *fuzzy* dapat bekerjasama dengan teknik – teknik kendali konvensional.
7. Logika *fuzzy* didasarkan pada bahasa alami.

2.11. Bagan Alir Dokumen

Bagan alir (*flowchart*) adalah bagan yang menunjukkan alir di dalam program atau prosedur sistem logika. Bagan alir ini digunakan terutama untuk mendefinisikan hubungan antara bagian (pelaku proses), proses yang dari manusia maupun proses dari computer dan aliran data dalam bentuk masukan dan keluaran.

Tabel 2. 1 Bagan Alir Dokumen

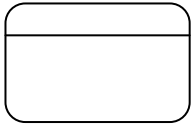
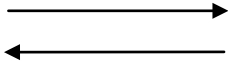
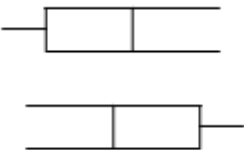
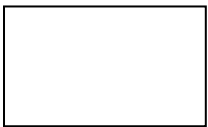
No	Simbol	Keterangan
1		Dokumen, Menunjukkan simbol dokumen input dan output baik untuk proses manual, mekanik, atau komputer.
2		Proses Manual, Menunjukkan kegiatan manual
3		Simpanan Off-Line, untuk menerangkan file arsip non komputer.
4		Proses, menunjukkan kegiatan proses dari operasi program komputer.
5		Garis Alir, digunakan untuk menunjukkan arus proses.
6		Penghubung (konektor), menunjukkan hubungan dengan bagian lain dalam satu halaman.

2.12. Diagram Alir Data

Diagram Alir Data adalah gambaran sistem secara logical dimana gambaran ini tidak tergantung pada perangkat keras, perangkat lunak, struktur data atau organisasi file yang digunakan untuk memudahkan pemakai yang kurang menguasai bidang komputer untuk mengerti sistem yang akan dikerjakan atau dikembangkan.(Ekawati, 2014)

Dalam menggambarkan diagram alir data menggunakan simbol – simbol seperti dibawah ini :

Tabel 2. 2 Diagram Alir Data

No	Simbol	Keterangan
1		<ul style="list-style-type: none"> • Simbol proses • Menunjukkan Proses komputerisasi
2		<ul style="list-style-type: none"> • Symbol Alir Data • Menunjukkan arah proses
3		<ul style="list-style-type: none"> • Symbol penyimpanan • Menunjukkan sebagai komponen untuk memudahkan kumpulan data atau informasi
4		<ul style="list-style-type: none"> • Simbol terminator • Menunjukkan organisasi (kelompok organisasi atau organisasi diluar sistem yang member atau menerima data

2.13. *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram (ERD) merupakan suatu bagan yang berisi komponen - komponen himpunan entitas dan himpunan relasi yang masing – masing dilengkapi dengan atribut – atribut yang menggambarkan seluruh fakta tentang suatu obyek yang kita tinjau.(Ekawati, 2014)

Entity Relationship Diagram (ERD) merupakan suatu kumpulan file – file yang saling berkaitan satu dengan yang lainnya pada model data relation hubungan antara file direlasikan dengan kunci relasi (*Relation Key*) yang merupakan kunci utama dimasing – masing file. Dengan adanya perancangan database membuat kinerja paket program seperti *MySQL* bekerja dengan optimal.

Entity Relationship Diagram menggunakan sejumlah notasi dan symbol untuk menggambarkan struktur dan hubungan antara data. Pada dasarnya ada 3 macam simbol yang digunakan, yaitu :

1. *Entity*

Suatu obyek yang sapat didefinisikan dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang dibuat. Sebagai contoh *entity* dapat berupa seseorang, sebuah tempat, sebuah objek dan sebuah kejadian atau konsep.


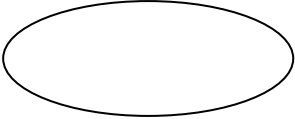
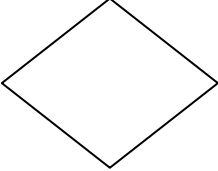
2. Atribut

Entity mempunyai elemen yang disebut atribut berfungsi mendeskripsikan suatu karakter.

3. Hubungan

Entity dapat berhubungan satu sama yang lain. Hubungan ini dinamakan relationship. Sebagai halnya *entity* maka dalam hubungannyapun harus dibedakan bentuk hubungan antar *entity* dengan isi dari hubungan tersebut. Beberapa relasi database yang mempermudah dalam penggunaan alur sistem dalam program.

Tabel 2. 3 Entity Relation Diagram

Simbol	Nama Simbol
	<i>Entity</i>
	<i>Atribut</i>
	Hubungan

2.14. *MySQL*

MySQL merupakan suatu database. *MySQL* dapat juga dikatakan sebagai database yang sangat cocok bila dipadukan dengan PHP. *MySQL* merupakan sebuah basis data yang mengandung satu atau sejumlah tabel. Tabel ini terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom. (Kustiyaningsih, 2010)

Database *MySQL* bersifat *open source* dan mampu menangani data yang sangat besar hingga ukuran Giga Byte, dengan kemampuan daya tampung data ini maka *MySQL* sangat cocok digunakan untuk mengcover data pada perusahaan baik yang kecil sampai perusahaan besar. (Anhar, 2010)

2.15. *Hypertext Preprocessor (PHP)*

PHP atau yang memiliki kepanjangan *Hypertext Preprocessor* merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu website dinamis. PHP menyatu dengan kode HTML, maksudnya adalah beda kondisi. HTML digunakan sebagai pembangun atau pondasi kerangka layout web, sedangkan PHP difungsikan sebagai prosesnya sehingga dengan adanya PHP tersebut, web akan sangat mudah *dimaintenance*.(Saputra, 2011)

2.16. Aplikasi

Aplikasi adalah perangkat lunak yang digunakan oleh *user* untuk melaksanakan pekerjaan atau aplikasi tertentu seperti mengetik, menggambar, menghitung, mendengarkan musik, dan lain-lain. Aplikasi yang dimaksud adalah semua perangkat lunak selain sistem operasi, diantaranya program aplikasi perkantoran, bahasa pemrograman, *virus*, *utility*, dan lain-lain. (Departemen Pendidikan Nasional, 2004).

Menurut (Maryono & Istiana, 2007) aplikasi merupakan program yang dikembangkan untuk memenuhi kebutuhan pengguna dalam menjalankan pekerjaan tertentu.