

BAB II

LANDASAN TEORI

2.1 Kajian Pustaka

Penelitian mengenai sistem pendukung keputusan dengan penerapan Metode *Fuzzy* bukanlah pertama kali dilakukan, sudah ada penelitian terdahulu yang menggunakan metode ini. Penelitian terdahulu yang relevan dengan penelitian ini adalah sebagai berikut.

Penelitian Syandi Chaesani Putra tahun 2017 dengan judul “Penentuan Jumlah Pembelian Produk Oval Ayam di PT. Seinergy Citra Karya Sembada Menggunakan *Fuzzy Tsukamoto*”. Sistem tersebut dibuat untuk menentukan berapa banyak pembelian yang harus dilakukan sesuai dengan permintaan konsumen[2].

Penelitian Rifki Setya Armanda dan Wayan Firdaus Mahmudy tahun 2016 dengan judul “Penerapan Algoritma Genetika Untuk Penentuan Batasan Fungsi Keanggotaan *Fuzzy Tsukamoto* Pada Kasus Peramalan Permintaan Barang”. Penelitian ini digunakan untuk meramalkan permintaan barang dengan melakukan penerapan algoritma genetika pada penentuan batasan fungsi keanggotaan *Fuzzy Tsukamoto*[3].

Penelitian Ginanjar Abdurrahman tahun 2011 dengan judul “Penerapan Metode *Tsukamoto* (Logika *Fuzzy*) Dalam Sistem Pendukung Keputusan Untuk Menentukan Jumlah Produksi Barang Berdasarkan Data Persediaan Dan Jumlah Permintaan”. Penelitian tersebut dibuat untuk menentukan berapa jumlah barang yang harus diproduksi berdasarkan dengan data persediaan dan jumlah

permintaan. meramalkan kunjungan wisatawan yang data ke Kota Batu untuk mempermudah pelaku pariwisata dalam memberikan pelayanan yang optimal [4].

Hasil dari penelitian-penelitian sebelumnya dapat menjadi informasi dan acuan bagi peneliti saat ini yang menerapkan metode yang sama, dimana penelitian ini menerapkan metode *Fuzzy Tsukamoto* untuk membuat Sistem Penunjang Keputusan Penentuan Produk Terlaris di Sogan Batik Rejodani dengan Menggunakan Metode *Fuzzy*

2.2 Konsep Sistem Pendukung Keputusan

2.2.1 Definisi Sistem

Murdick dan Ross (1993) mendefinisikan sistem sebagai seperangkat elemen yang digabungkan satu dengan lainnya untuk suatu tujuan bersama [5].

Mc.Leod (1995) mendefinisikan sistem sebagai sekelompok elemen-elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. Sumber daya mengalir dari elemen output dan untuk menjamin prosesnya berjalan dengan baik maka dihubungkan dengan mekanisme kontrol [5].

Scoot (1996), sistem terdiri dari unsur-unsur seperti masukan (*input*), pengolahan (*processing*), serta keluaran (*output*).

Dengan demikian, secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung satu sama lain [5].

2.2.2 Definisi Sistem Penunjang Keputusan (*Decision Support System*)

Sistem Penunjang Keputusan atau *Decision Support System (DSS)* merupakan sistem informasi interaksi yang menyediakan informasi, permodelan, dan pemanipulasian data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur, di mana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat (Alter, 2002) [6].

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

Aplikasi DSS menggunakan data, memberikan antarmuka pengguna yang mudah, dan dapat menggabungkan pemikiran pengambil keputusan. DSS lebih ditujukan untuk mendukung manajemen dalam melakukan pekerjaan yang bersifat analitis dalam situasi yang kurang terstruktur dan dengan kriteria yang kurang jelas

DSS tidak dimaksudkan untuk mengotomatisasikan pengambilan keputusan, tetapi memberikan perangkat interaktif yang memungkinkan pengambilan keputusan untuk melakukan berbagai analisis menggunakan model-model yang tersedia.

2.3 Logika *Fuzzy*

2.3.1 Pengertian Logika *Fuzzy*

Logika *Fuzzy* pertama kalinya diperkenalkan oleh Prof Lutfi A. Zadeh seorang peneliti di Universitas California Barkley pada tahun 1965. Menurut Prof Zadeh logika benar salah tidak dapat mewakili setiap pemikiran manusia, kemudian dikembangkanlah Logika *Fuzzy* yang dapat mempresentasikan setiap keadaan atau mewakili pemikiran manusia. Perbedaan antara logika tegas dan logika *fuzzy* terletak pada keanggotaan elemen dalam suatu himpunan. Jika dalam logika tegas suatu elemen mempunyai dua pilihan yaitu terdapat dalam himpunan atau bernilai 1 yang berarti benar dan tidak pada himpunan atau bernilai 0 yang berarti salah. Sedangkan dalam logika *fuzzy*, keanggotaan elemen berada di interval $[0,1]$.

Logika *fuzzy* memiliki beberapa kelebihan dalam pengambilan keputusan sebagai berikut [2]:

1. Logika *fuzzy* memiliki konsep yang sangat sederhana sehingga mudah untuk dimengerti.
2. Logika *fuzzy* sangat fleksibel, artinya mampu beradaptasi dengan perubahan-perubahan dan ketidakpastian.
3. Logika *fuzzy* memiliki toleransi terhadap data yang tidak tepat.
4. Logika *fuzzy* mampu mensistemkan fungsi-fungsi non-liner yang sangat kompleks.
5. Logika *fuzzy* dapat mengaplikasikan pengalaman atau pengetahuan dari para pakar.

6. Logika *fuzzy* dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
7. Logika *fuzzy* didasarkan pada bahasa sehari-hari sehingga mudah dimengerti.

2.3.2 Himpunan *Fuzzy*

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu item x dalam suatu himpunan A , yang sering ditulis dengan $\mu_A [x]$, memiliki 2 kemungkinan, yaitu

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

Pada penggunaan himpunan *crisp*, adanya perubahan kecil saja pada suatu nilai mengakibatkan perbedaan kategori yang cukup signifikan. Himpunan *fuzzy* digunakan untuk mengantisipasi hal tersebut. Seberapa besar ekstensinya dalam himpunan tersebut dapat dilihat pada nilai keanggotaannya.

Kalau pada himpunan *crisp*, nilai keanggotaan hanya ada 2 kemungkinan, yaitu 0 atau 1, pada himpunan *fuzzy* nilai keanggotaan terletak pada rentang 0 sampai 1. apabila nilai keanggotaan *fuzzy* $\mu_A [x] = 0$ berarti x tidak menjadi anggota himpunan A , demikian pula apabila x memiliki nilai keanggotaan $\mu_A [x] = 1$ berarti x menjadi anggota penuh himpunan A .

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 10, 25, 40, dsb.

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

1. Variabel *fuzzy*

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

2. Himpunan *fuzzy*

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

Contoh: Variabel umur, terbagi menjadi 3 himpunan *fuzzy*, yaitu: MUDA, PAROBAYA, TUA.

3. Semesta pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*.

Contoh: Semesta pembicaraan untuk variabel umur: $[0 +\infty]$

4. Domain

Domain himpunan *fuzzy* adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan *fuzzy*.

Contoh:

- MUDA = $[0 \ 45]$

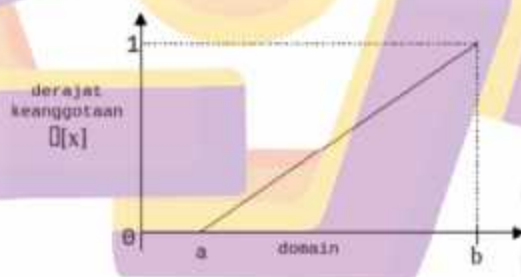
- TUA = $[45 \ +\infty)$

2.3.3 Fungsi Keanggotaan

Fungsi keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (derajat keanggotaan). Ada beberapa fungsi yang bisa digunakan, antara lain [2] :

1. Representasi Linear Naik

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. (Gambar 2.1)



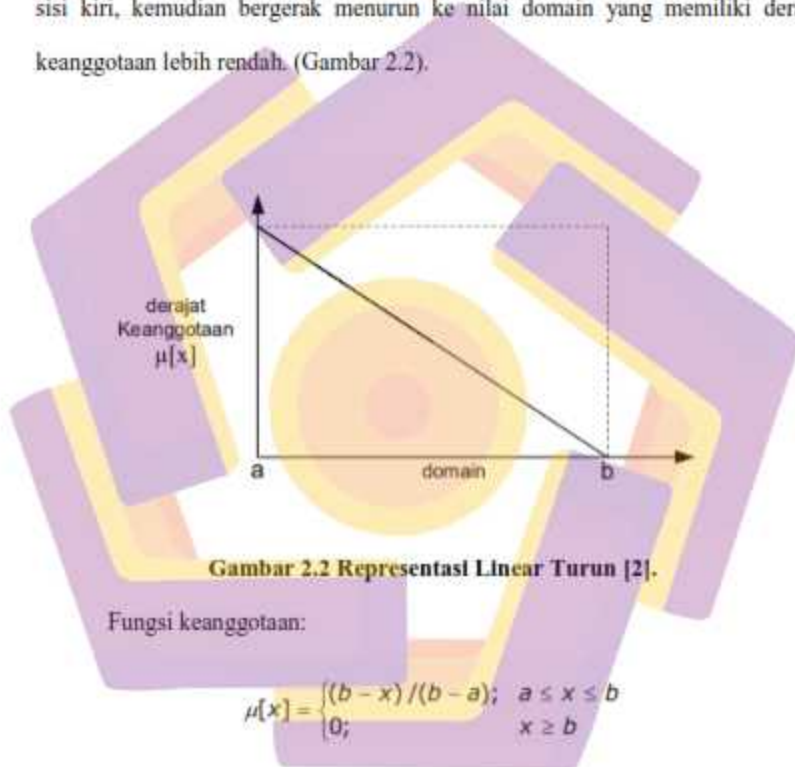
Gambar 2.1 Representasi Linear Naik [2].

Fungsi Keanggotaan:

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

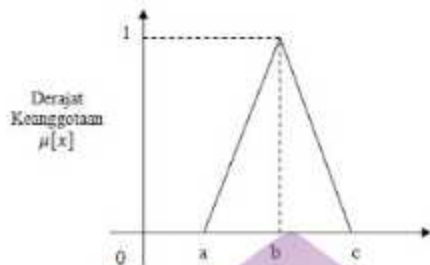
2. Representasi Linear Turun

Representasi Linear Turun merupakan kebalikan dari Representasi Linear Naik. 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).



3. Representasi Kurva Segitiga

Representasi kurva segitiga merupakan gabungan dari representasi linier (Klir, Clair & Yuan, 1997: 83-86). Representasi dapat dilihat seperti pada (Gambar 2.3)



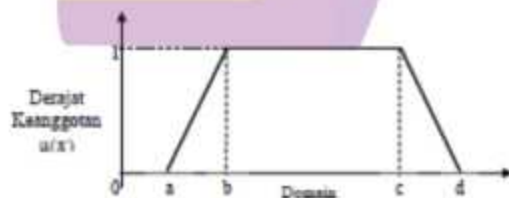
Gambar 2.3 Representasi Kurva Segitiga [2].

Fungsi keanggotaannya:

$$\mu(x) = \begin{cases} 0, & x \leq a \text{ atau } x \geq c \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \end{cases}$$

4. Representasi Kurva Trapesium

Representasi kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1. Representasi kurva trapesium dapat dilihat pada (Gambar 2.4).



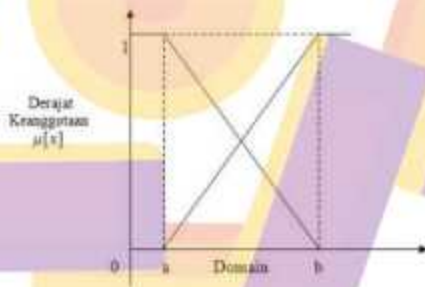
Gambar 2.4 Representasi Kurva Trapesium [2].

Fungsi Keanggotaannya:

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

5. Representasi Kurva Bentuk Bahu

Daerah yang terletak di tengah-tengah suatu variabel yang direpresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik turun. Tetapi terkadang salah satu sisi dari variabel tersebut tidak mengalami perubahan. Himponan *fuzzy* "bahu", bukan segitiga, digunakan untuk mengakhiri variabel suatu daerah *fuzzy*. Bahu kiri bergerak dari benar ke salah, demikian juga bahu kanan bergerak dari salah ke benar. Representasi fungsi keanggotaan untuk kurva bahu seperti pada (Gambar 2.5)



Gambar 2.5 Representasi Kurva Bentuk Bahu [2].

Fungsi keanggotaanya:

$$\mu[x, a, b] = \begin{cases} 0; & x \leq b \\ (b-x) / (b-a); & a \leq x \leq b \\ 1; & x \geq a \\ 0; & x \leq a \\ (x-a) / (b-a); & a \leq x \leq b \\ 1; & x \geq b \end{cases}$$

2.3.4 Operator Dasar Fuzzy

Ada beberapa operasi yang didefinisikan secara khusus untuk mengkombinasi dan memodifikasi himpunan fuzzy. Nilai keanggotaan sebagai hasil dari 2 operasi disebut fire strength atau α -predikat. Operator dasar fuzzy yaitu :

1. Operator AND

Operator ini berhubungan dengan operasi interseksi pada himpunan. α -predikat sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antara elemen pada himpunan-himpunan yang bersangkutan.

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

2. Operator OR

Operator ini berhubungan dengan operasi union pada himpunan. α -predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antara elemen pada himpunan-himpunan yang bersangkutan.

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

3. Operator NOT

Operator ini berhubungan dengan operasi komplemen himpunan. α -predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangi nilai keanggotaan elemen pada himpunan dengan 1.

$$\mu_{A'} = 1 - \mu_A[x]$$

4. Penalaran Monoton

Metode penalaran secara monoton digunakan sebagai dasar untuk teknik implikasi *fuzzy*. Jika 2 daerah *fuzzy* direlasikan dengan implikasi sederhana sebagai berikut:

IF x is A THEN y is B

Transfer fungsi :

$$y = f(x, A, B)$$

maka sistem *fuzzy* dapat berjalan tanpa harus melalui komposisi dan dekomposisi *fuzzy*. Nilai output dapat diestimasi secara langsung dari nilai keanggotaan yang berhubungan dengan antisendennya. Penalaran ini sudah jarang digunakan.

6. Fungsi Implikasi

Tiap-tiap aturan (proposisi) pada basis pengetahuan *fuzzy* akan berhubungan dengan suatu relasi *fuzzy*. Bentuk umum dari aturan yang digunakan dalam fungsi implikasi adalah :

IF x is A THEN y is B

Dengan x dan y adalah skalar, dan A dan B adalah himpunan *fuzzy*.
 Proposisi yang mengikuti IF disebut antisenden, sedangkan proposisi yang mengikuti THEN disebut konsekuen. Proposisi dapat diperluas dengan menggunakan operator *fuzzy*, seperti :

IF (x1 is A1)o(x2 is A2)o(x3 is A3) ... o(xn is An) THEN y is B dengan o adalah operator (misal : OR atau AND).

Secara umum, ada 2 fungsi implikasi yang dapat digunakan, yaitu :

a. Min (*minimum*)

Fungsi ini akan memotong output himpunan *fuzzy*.

b. Dot (*product*)

Fungsi ini akan menskala output himpunan *fuzzy*.

2.3.5 Fuzzy Inference System (FIS)

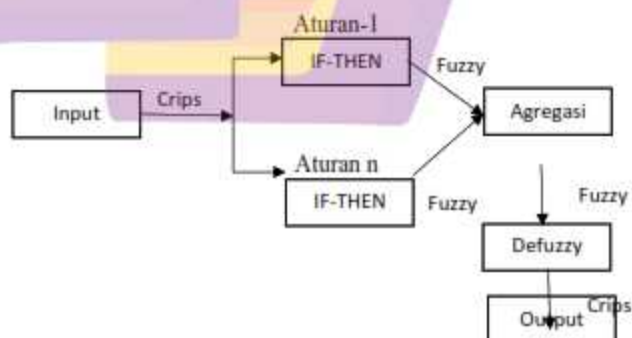
Inferensi adalah proses penggabungan banyak aturan berdasarkan data yang tersedia. Komponen yang melakukan inferensi dalam sistem pakar disebut mesin inferensi. Dua pendekatan untuk menarik kesimpulan pada *IF-THEN rule* (aturan jika-maka) adalah *forward chaining* dan *backward chaining* (Turban dkk, 2015) [7].

Sistem ini berfungsi untuk mengambil keputusan melalui proses tertentu dengan mempergunakan aturan inferensi berdasarkan logika *fuzzy*. Sistem inerensi *fuzzy* memiliki empat tahap, yaitu [7]:

1. Fuzzifikasi
2. Penalaran logika *fuzzy* (*fuzzy logic reasoning*)

3. Basis pengetahuan (*knowledge base*), yang terdiri dari dua bagian:
 - a. Basis data (*database*), yang memuat fungsi-fungsi keanggotaan dari himpunan-himpunan *fuzzy* yang terkait dengan nilai dari variabel-variabel *linguistic* yang dipakai.
 - b. Basis aturan (*rule base*), yang memuat aturan-aturan berupa implikasi *fuzzy*.
4. Defuzzifikasi

Pada sistem inferensi *fuzzy*, nilai-nilai masukan tegas dikonversikan oleh unit fuzzifikasi ke nilai *fuzzy* yang sesuai. Hasil pengukuran yang telah *difuzzykan* kemudian diproses oleh unit penalaran dengan menggunakan unit basis pengetahuan yang akan menghasilkan himpunan *fuzzy* sebagai keluarannya. Langkah terakhir dikerjakan oleh unit defuzzifikasi akan menerjemahkan himpunan keluaran ke dalam nilai yang tegas. Nilai tegas inilah yang kemudian direalisasikan dalam bentuk suatu tindakan yang dilaksanakan dalam proses tersebut. Langkah-langkah tersebut akan dijelaskan berikut ini:



Gambar 2.6 Struktur Dasar Inferensi Fuzzy [7].

S

istim inferensi *fuzzy* menerima *input crisp*. *Input* ini kemudian dikirim ke basis pengetahuan yang berisi n aturan *fuzzy* dalam bentuk *IF-THEN*. *Fire strength* (nilai keanggotaan anteseden akan dicari pada setiap aturan). Apabila aturan lebih dari satu, maka akan dilakukan agregasi semua aturan. Selanjutnya pada hasil agregasi akan dilakukan *defuzzy* untuk mendapatkan nilai *crisp* sebagai *output* sistem.

2.3.6 Fuzzy Tsukamoto

Menurut Setiadji (2009), pada Metode *Tsukamoto* implikasi setiap aturan berbentuk implikasi “sebab-akibat” atau implikasi “*input-output*” yang mana antara anteseden dan konsekuen harus ada hubungannya. Setiap aturan direpresentasikan menggunakan himpunan-himpunan *fuzzy*, dengan fungsi keanggotaan yang monoton. Kemudian untuk menentukan hasil tegas (*crisp solution*) digunakan rumus defuzzifikasi yang disebut metode rata-rata terpusat atau metode defuzzifikasi rata-rata terpusat (*center average defuzzifier*) [7].

Terdapat empat tahap dalam menganalisis produksi barang menggunakan Metode *Tsukamoto* (Agustin, 2015), yaitu [(Agustin, 2015), yaitu [7]:

1. Fuzzifikasi

Fuzzifikasi adalah proses mengubah nilai masukan tegas menjadi nilai masukan *fuzzy*. Nilai masukan tegas pada tahap ini dimasukkan ke dalam

fungsi pengaburan yang telah dibentuk sehingga menghasilkan nilai masukan *fuzzy*.

2. Pembentukan Aturan *Fuzzy*

Aturan *Fuzzy* dibentuk untuk memperoleh hasil keluaran tegas. Aturan *fuzzy* yang digunakan adalah aturan “jika-maka” dengan operator antar variabel masukan adalah operator “dan”. Pernyataan yang mengikuti “jika” disebut sebagai antiseden dan pernyataan yang mengikuti “maka” disebut sebagai konsekuen.

3. Analisis Logika *Fuzzy*

Setiap aturan yang dibentuk merupakan suatu pernyataan implikasi. Analisis logika *fuzzy* yang digunakan pada tahap ini adalah fungsi implikasi *min*, karena operator yang digunakan pada aturan “jika-maka” adalah operator “dan”. Fungsi implikasi *min* yaitu mengambil nilai keanggotaan terkecil antar elemen pada himpunan *fuzzy* yang bersangkutan. Hasil fungsi implikasi dari masing-masing aturan disebut α -predikat atau bisa ditulis α

4. Defuzzifikasi

Defuzzifikasi adalah proses mengubah nilai keluaran *fuzzy* menjadi nilai keluaran tegas. Rumus yang digunakan untuk tahap ini adalah rata-rata terbobot.

$$z = \frac{\sum x_i \alpha_i}{\sum \alpha_i}, i = 1, 2, 3, \dots$$

Dengan

z : nilai rata-rata terbobot

x_i : nilai konsekuen pada aturan ke- i

α_i : nilai α -predikat pada aturan ke- i


2.4 Flowchart System







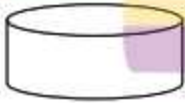
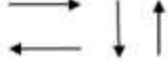
Menurut Yakub (2012) *flowchart* adalah bagian yang menggambarkan urutan instruksi proses dan hubungan satu proses dengan proses lainnya menggunakan simbol-simbol tertentu. *Flowchart* digunakan sebagai alat bantu komunikasi dan dokumentasi [6].

Menurut Supardi (2013) *flowchart* merupakan bagan yang menunjukkan alir di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi [6].

Flowchart disusun dengan simbol-simbol. Simbol ini dipakai sebagai alat bantu menggambarkan proses di dalam program. Ada beberapa simbol yang digunakan saat membuat *flowchart*, yaitu [6]:

Tabel 2.1 Simbol Flowchart.

No.	Simbol	Keterangan
1.		Simbol Start atau End yang mendefinisikan awal atau akhir dari sebuah flowchart

2.		Simbol pemerosesan yang terjadi pada sebuah alur kerja.
3.		Simbol yang menyatakan bagian dari program (sub program)
4.		Simbol Input/Output yang mendefinisikan masukan dan keluaran proses.
5.		Menyatakan penyambung ke simbol lain dalam satu halaman.
6.		Menyatakan penyambung ke halaman lainnya.
7.		Menyatakan pencetakan (dokumen) pada kertas.
8.		Menyatakan media penyimpanan drum magnetik.
9.		Menyatakan arah aliran pekerjaan (proses).


2.5 Entity Relationship Diagram (ERD)

Menurut Yakub (2012) menjelaskan bahwa Entity Relationship Diagram (ERD) merupakan suatu model jaringan yang menggunakan susunan data yang disimpan pada sistem secara abstrak. ERD juga menggambarkan hubungan antara satu entitas yang memiliki sejumlah atribut dengan entitas yang lain dalam suatu sistem yang terintegrasi. ERD digunakan oleh perancang sistem untuk memodelkan data yang nantinya akan dikembangkan menjadi basis data (database) [6].

Ada beberapa simbol yang digunakan saat membuat *Entity Relationship Diagram*, yaitu [6]:

Tabel 2.2 Simbol Entity Relationship Diagram.

No.	Simbol	Keterangan
1.		Entitas, yaitu kumpulan dari objek yang dapat diidentifikasi secara unik.
2.		Relasi, yaitu hubungan yang terjadi antara satu atau lebih entitas. Jenis hubungan antara lain; satu ke satu, satu ke banyak, dan banyak ke banyak.
3.		Atribut, yaitu karakteristik dari entity atau relasi yang merupakan penjelasan detail tentang entitas.


4.		Hubungan antara entity dengan atributnya dan himpunan entitas dengan himpunan relasinya.
----	---	--

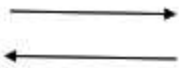


2.6 Data Flow Diagram (DFD)

Menurut Kristanto (2011), DFD merupakan 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 [6].

Ada beberapa simbol yang digunakan saat membuat *Data Flow Diagram*, yaitu [6]:

Tabel 2.3 Simbol Data Flow Diagram.

No.	Simbol	Keterangan
1.		Entity Luar, merupakan sumber atau tujuan dari aliran data dari atau ke sistem. Entity luar merupakan lingkungan luar sistem, jadi tidak tahu menahu mengenai apa yang terjadi di entity luar.

2.		Aliran data, menggambarkan aliran data dari satu proses ke proses lainnya.
3.		Proses, proses atau fungsi yang mentransformasikan data secara umum.
4.		Tempat penyimpanan, merupakan komponen yang berfungsi untuk menyimpan data atau file

