

## BAB II TINJAUAN PUSTAKA

### 2.1 Penelitian Terkait

Beberapa penelitian terkait dengan penelitian ini diantaranya sebagai berikut:

Faz Faidhani (2018) melakukan penelitian dengan judul Sistem Pendukung Keputusan Penentu Bidang Keahlian Mahasiswa Program Studi Informatika Universitas Tanjungpura dengan Metode ELECTRE menjelaskan tujuan dari penelitian tersebut untuk menghasilkan rekomendasi dalam menentukan bidang keahlian mahasiswa berdasarkan rata-rata nilai mata kuliah kelompok keahlian, kemampuan mahasiswa dan minat mahasiswa.

Yulmaini (2015) melakukan penelitian dengan judul Penggunaan Metode *Fuzzy Inference System* (FIS) Mamdani Dalam Pemilihan Peminatan Mahasiswa Untuk Tugas Akhir menjelaskan untuk menghasilkan sistem *fuzzy* dalam penentuan peminatan tugas akhir mahasiswa menggunakan nilai mata kuliah. Dalam penelitian ini menggunakan 12 variabel masukan yaitu nilai-nilai mata kuliah yang terkait, dengan himpunan *fuzzy* tinggi [50-100] dan rendah [0-80], kemudian 3 variabel keluaran yaitu kelompok peminatan dengan himpunan *fuzzy* tinggi [50-100] dan rendah [0-80].

Astrie Kusuma Dewi, Adhistya Erna Permanasari, Indriana Hidayah (2016) melakukan penelitian dengan judul Kesesuain Minat Mahasiswa Dengan Judul Tesis Mahasiswa Menggunakan Metode *Fuzzy Mamdani* menjelaskan tentang pemilihan minat tesis sesuai dengan minat mahasiswa dan motivasi dalam proses pengerjaan tesis. Data dalam penelitian ini menggunakan data kuisisioner minat dan kuisisioner motivasi yang diisi oleh mahasiswa baru semester 1 dan 2. Dari tahap pengumpulan data hasil kuisisioner dibuat parameter untuk pemilihan minat tesis, kemudian dilakukan *processing* data sebelum diproses menggunakan metode *fuzzy mamdani*. Tujuan dari penelitian ini untuk membantu mahasiswa masuk dalam grup riset sesuai dengan judul tesis atau proposal tesis.

**Tabel 2.1** Penelitian Terkait

No.	Penulis	Judul	Keterangan
1.	Faz Faidhani (2018)	Sistem Pendukung Keputusan Penentu Bidang Keahlian Mahasiswa Program Studi Informatika Universitas Tanjungpura Dengan Metode ELECTRE	Kriteria yang digunakan adalah rata-rata nilai mata kuliah kelompok keahlian, minat mahasiswa, dan kemampuan mahasiswa.
2.	Yulmaini (2015)	Penggunaan Metode <i>Fuzzy Inference System</i> (FIS) Mamdani Dalam Pemilihan Peminatan Mahasiswa Untuk Tugas Akhir	Variabel yang digunakan adalah nilai-nilai mata kuliah yang terkait dengan tugas akhir.
3.	Astrie Kusuma Dewi, Adhistya Erna Permasari, Indriana Hidayah (2016)	Kesesuaian Minat Mahasiswa dengan Judul Tesis Mahasiswa Menggunakan Metode <i>Fuzzy Mamdani</i>	Menggunakan kuisisioner minat terhadap grup riset, dan kuisisioner motivasi.

## 2.2 *Hypertext Preprocessor (PHP)*

PHP (*Hypertext preprocessor*) adalah bahasa standar yang digunakan dalam dunia *website* dan dibangun sekitar tahun 1995. Menurut Sidik (2012), PHP secara umum dikenal sebagai bahasa pemrograman *script* yang membuat dokumen HTML secara *on the fly* yang dieksekusi dalam server *web*, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemrograman *server-side*.

PHP adalah *script* bersifat *server-side* yang berarti pengerjaan *script* dilakukan di server, baru kemudian hasilnya dikirimkan ke *browser*. *Script* ini akan membuat suatu aplikasi dapat diintegrasikan kedalam HTML sehingga suatu halaman *website* tidak lagi bersifat statis namun menjadi dinamis. PHP bersifat *open source* sehingga dapat digunakan secara cuma-cuma dan mampu lintas *platform*, yaitu dapat berjalan pada sistem operasi *Windows* maupun *Linux*. PHP juga dibangun sebagai modul pada *web server apache*.

Dalam menulis *Script* PHP, selalu diawali dengan tanda **<?php** dan diakhiri dengan tag **?>**. Pada *block code* berikut, kita akan menggunakan kedua tanda tersebut agar sistem mendeteksi *script* kita sebagai PHP, bukan HTML

biasa. Ketika membuat program dengan PHP, operasi yang paling sering digunakan yaitu operasi CRUD atau biasa dikenal dengan *Create*, *Read*, *Update*, dan *Delete*.

*Create* adalah sebuah proses menciptakan atau membuat data dimana operasi ini seringkali digunakan ketika membuat suatu program. *Read* adalah suatu proses untuk membaca atau menampilkan data yang telah dibuat kedalam sebuah tabel. *Update* atau edit adalah suatu proses mengubah data dari data yang semula ada diubah sesuai dengan kebutuhan. Dalam sebuah program dibutuhkan sebuah *button* yang berfungsi untuk merubah data gunanya apabila terjadi suatu kesalahan tanpa menghapus data tersebut. *Delete* adalah suatu proses menghapus *record* yang telah ditentukan dari dalam tabel. *File delete* ini keberadaannya sangat fatal karena berfungsi untuk menghapus sebuah kesalahan masukan data.

### 2.3 Laravel

Laravel merupakan *framework* PHP yang menekankan pada kesederhanaan dan fleksibilitas pada desainnya. Laravel dirilis dibawah lisensi MIT dengan sumber kode yang disediakan di GitHub. Sama seperti *framework* lainnya PHP lainnya, laravel dibangun dengan *Model View Controller*, laravel dilengkapi *command line tool* yang bernama "artisan" yang bisa digunakan untuk *packaging bundle* dan *installasi bundle* (Rohman, 2014).

### 2.4 MySQL

MySQL merupakan sistem *database* yang banyak digunakan untuk pengembangan aplikasi *website*. Hampir semua aplikasi *website* yang dikembangkan saat ini membutuhkan teknologi *database* untuk menyimpan dan mengelola data-data yang digunakan didalamnya (Budi Raharjo, 2016).

Menurut Arief (2011), MySQL adalah salah satu jenis *database* server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *website* yang menggunakan *database* sebagai sumber dan pengolahan datanya. MySQL merupakan *database* yang pertama kali didukung oleh bahasa pemrograman *script* untuk internet (PHP dan Perl). MySQL dan PHP dianggap sebagai pasangan *software* pembangun aplikasi *website* yang ideal. MySQL lebih sering digunakan

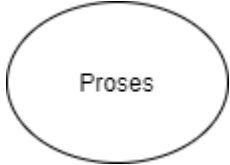


untuk membangun aplikasi berbasis *website*, umumnya pengembangan aplikasinya menggunakan bahasa pemrograman *script* PHP.


## 2.5 Data Flow Diagram (DFD)

Menurut Sukamto dan Shalahuddin (2013), *Data Flow Diagram* 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*).

DFD dapat digunakan untuk merepresentasikan sebuah sistem atau perangkat lunak pada beberapa level yang lebih detail untuk merepresentasikan aliran informasi atau fungsi yang lebih detail. DFD menyediakan mekanisme untuk pemodelan fungsional ataupun pemodelan aliran informasi. Berikut ini adalah beberapa simbol atau notasi dalam *Data Flow Diagram* menurut Sukamto dan Shalahuddin (2013), seperti pada Tabel 2.2.

**Tabel 2.2** Simbol - Simbol DFD

No.	Simbol	Keterangan
1.		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 didalam kode program. Catatan: nama yang diberikan pada sebuah proses biasanya berupa kata kerja.
2.		<i>File</i> atau basis data atau penyimpanan; 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 ( <i>Entity Relationship Diagram</i> (ERD), <i>Conceptual Data Model</i> (CDM), <i>Physical Data Model</i> (PDM)). Catatan: nama yang diberikan pada sebuah penyimpanan biasanya kata benda.
3.		Entitas luar ( <i>external entity</i> ) atau masukan ( <i>input</i> ) atau keluaran ( <i>output</i> ) atau orang yang memakai / berinteraksi dengan perangkat lunak yang

No.	Simbol	Keterangan
		dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan. Catatan: nama yang digunakan pada masukan ( <i>input</i> ) atau keluaran ( <i>output</i> ) berupa kata benda.
4.		Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan ( <i>input</i> ) atau keluaran ( <i>output</i> ). Catatan: nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa”.

Sukamto dan Shalahuddin (2013) menjelaskan tentang tahapan-tahapan perancangan dengan menggunakan *Data Flow Diagram* yaitu:

1. DFD Level 0 atau sering disebut juga *Context Diagram*

DFD Level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DFD Level 0 digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.

2. DFD Level 1

DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan. DFD Level 1 merupakan hasil *breakdown* DFD Level 0 yang sebelumnya sudah dibuat.

3. DFD Level 2

Modul-modul pada DFD Level 1 dapat di-*breakdown* menjadi DFD Level 2. Modul mana saja yang harus di-*breakdown* lebih detail tergantung pada kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu di-*breakdown* lagi. Untuk sebuah sistem, jumlah DFD Level 2 sama dengan jumlah modul pada DFD Level 1 yang di-*breakdown*.

4. DFD Level 3 dan seterusnya

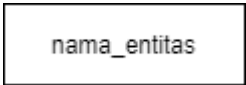

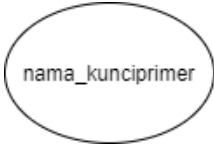
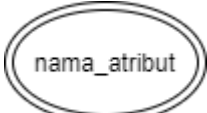
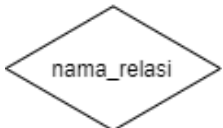
DFD Level 3,4,5, dan seterusnya merupakan *breakdown* dari modul pada DFD Level di-atasnya. *Breakdown* pada level 3,4,5 dan seterusnya aturannya sama persis dengan DFD Level 1 atau 2.

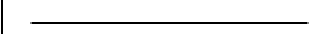
## 2.6 Entity Relationship Diagram (ERD)

Menurut Sukamto dan Shalahuddin (2013), *Entity Relationship Diagram* (ERD) merupakan bentuk paling awal dalam melakukan perancangan basis data relasional. ERD biasanya memiliki hubungan *binary* (satu relasi menghubungkan dua buah entitas). Beberapa metode perancangan ERD menoleransi hubungan relasi *ternary* (satu relasi menghubungkan tiga buah relasi) atau *N-ary* (satu relasi menghubungkan banyak entitas), tapi banyak metode perancangan ERD yang tidak mengizinkan ternary atau *N-ary*.

Pada umumnya banyak sekali model yang digunakan dalam ERD, berikut ini adalah simbol atau notasi yang dijelaskan Sukamto dan Shalahuddin (2013), yaitu:

**Tabel 2.3** Simbol - Simbol ERD

No.	Simbol	Keterangan
1.	<p>Entitas/entity</p> 	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 komputer; penamaan entitas biasanya lebih ke data benda dan belum merupakan nama tabel.
2.	<p>Atribut</p> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
3.	<p>Atribut Kunci Primer</p> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).
4.	<p>Atribut multi nilai / <i>multi value</i></p> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
5.	<p>Relasi</p> 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.
6.	Asosiasi / <i>association</i>	Penghubung antara relasi dan entitas dimana di

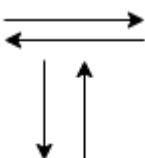



No.	Simbol	Keterangan
		kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas A dan entitas B.

## 2.7 Flowchart

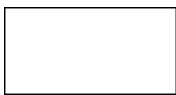
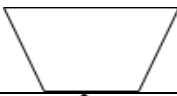
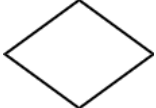



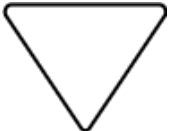

Menurut Indrajani (2011), *flowchart* merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program. Biasanya mempermudah penyelesaian masalah yang khususnya perlu dipelajari dan dievaluasi lebih lanjut.

Ladjamudin (2013), *flowchart* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. Berikut ini adalah beberapa simbol atau notasi yang digunakan dalam *flowchart* seperti pada tabel berikut:



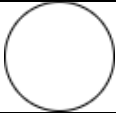



**Tabel 2.4** Simbol-Simbol *Flowchart*

No.	Simbol	Keterangan
1.		Untuk menyatakan jalannya arus suatu proses.
2.		Untuk menyatakan bahwa adanya transisi suatu data/informasi dari satu lokasi ke lokasi lainnya.
3.		Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang sama.
4.		Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang berbeda.

**Tabel 2.5** Simbol-Simbol Proses

No.	Simbol	Keterangan
1.		Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang berbeda.
2.		Untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer.
3.		Untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, ya/tidak.
4.		Untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
5.		Untuk menyatakan permulaan atau akhir suatu program.
6.		Untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai <i>keyboard</i> .
7.		Untuk menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu.
8.		Untuk memasukkan data secara manual dengan menggunakan <i>online keyboard</i> .

**Tabel 2.6** Simbol-simbol *Input-Output*

No.	Simbol	Keterangan
1.		Untuk menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.
2.		Untuk menyatakan <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu.
3.		Untuk menyatakan <i>input</i> berasal dari pita magnetis atau <i>output</i> disimpan ke pita magnetis.
4.		Untuk menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i> .
5.		Untuk menyatakan laporan ke <i>printer</i> .
6.		Untuk menyatakan peralatan <i>output</i> yang digunakan berupa layar (video, komputer).



## 2.8 Logika Fuzzy

### 2.8.1 Pengertian Logika Fuzzy

Logika *Fuzzy* (*fuzzy logic*) merupakan cabang dari sistem kecerdasan buatan (*artificial intelligence*) yang memungkinkan kemampuan manusia dalam berfikir dibuat dalam bentuk algoritma yang kemudian dijalankan oleh mesin (komputer) (Kusumadewi, 2003).

Logika *fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang *input* kedalam suatu ruang *output*. Ada beberapa alasan mengapa menggunakan Logika *fuzzy* (Kusumadewi dan Purnomo, 2004), antara lain:

- a. Konsep logika *fuzzy* mudah dimengerti. Konsep matematis yang mendasari penalaran *fuzzy* sangat sederhana dan mudah dimengerti.
- b. Logika *fuzzy* sangat fleksibel.
- c. Logika *fuzzy* memiliki toleransi terhadap data-data yang tidak tepat.
- d. Logika *fuzzy* mampu memodelkan fungsi-fungsi nonlinear yang sangat kompleks.
- e. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman - pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
- f. Logika *fuzzy* dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
- g. Logika *fuzzy* didasarkan pada bahasa alami.

Logika *fuzzy* dapat diimplementasikan dalam tiga tahap, yaitu:

1. Tahap *Fuzzyfikasi* (*Fuzzyfication*), yaitu pemetaan dari nilai masukan tegas kedalam himpunan *fuzzy*.
2. Tahap Inferensi, yaitu pembangkitan aturan *fuzzy*.
3. Tahap *Defuzzyfikasi* (*Defuzzyfication*), yaitu transformasi keluaran dari nilai *fuzzy* ke nilai tegas (*crisp*).

### 2.8.2 Variabel Fuzzy

Variabel *fuzzy* adalah variabel-variabel yang akan dibicarakan dalam suatu sistem *fuzzy*. Contoh variabel *fuzzy*, yaitu: umur, temperatur, permintaan, dan lainnya. Variabel *fuzzy* umumnya harus di definisikan terlebih dahulu ketika telah menemukan masalah yang akan di selesaikan.

### 2.8.3 Himpunan *Fuzzy*

Himpunan *fuzzy* adalah himpunan-himpunan yang akan dibicarakan atau mewakili suatu keadaan tertentu pada suatu variabel dalam sistem *fuzzy*. Namun, dalam himpunan tegas setiap elemen harus didefinisikan secara tegas apakah elemen tersebut termasuk dalam anggota himpunan atau tidak. Sebagai contoh, yaitu variabel umur dengan himpunan *fuzzy* muda, parobaya, tua. Himpunan *fuzzy* memiliki 2 atribut, sebagai berikut:

1. Linguistik, yaitu penamaan suatu kelompok yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti: LAMBAT, SEDANG, CEPAT.
2. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel, seperti: 50, 65, 70, dan sebagainya.

### 2.8.4 Semesta Pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel dari nilai terkecil hingga nilai terbesar. Semesta pembicaraan bersifat monoton naik., dan dapat berupa bilangan positif maupun negatif.

### 2.8.5 Domain Himpunan *Fuzzy*

Domain himpunan *fuzzy* adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan *fuzzy*. Domain merupakan himpunan bilangan riil yang naik (bertambah) secara monoton dari kiri ke kanan. Nilai domain dapat berupa bilangan positif maupun negatif.

### 2.8.6 Fungsi Keanggotaan

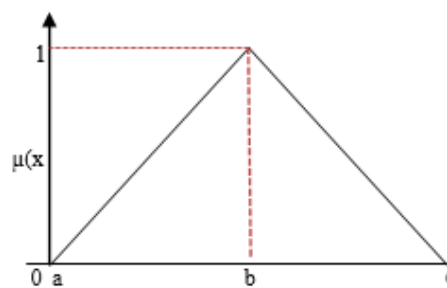
Fungsi keanggotaan (*membership functions*) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data kedalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval dari 0 sampai 1.

Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah melalui pendekatan fungsi. (Kusumadewi dan Purnomo 2004).

Dalam sistem *fuzzy* dikenal beberapa fungsi keanggotaan (*membership functions*), diantaranya yaitu:

a. Fungsi Segitiga

Fungsi ini merupakan gabungan antara 2 garis linear. Fungsi ini memiliki satu nilai  $x$  yang memiliki derajat keanggotaan sama dengan 1, yaitu ketika  $x=b$ . Tetapi nilai-nilai di sekitar  $b$  memiliki derajat keanggotaan yang turun cukup tajam (menjauh dari 1).



**Gambar 2.1** Fungsi Segitiga

Fungsi keanggotaan :

$$\mu(x) = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ \frac{x-a}{b-a}; & a \leq x \leq b \\ \frac{c-x}{c-b}; & b \leq x \leq c \end{cases} \quad (2.1)$$

Keterangan:

$a$  = nilai domain terkecil yang mempunyai derajat keanggotaan nol

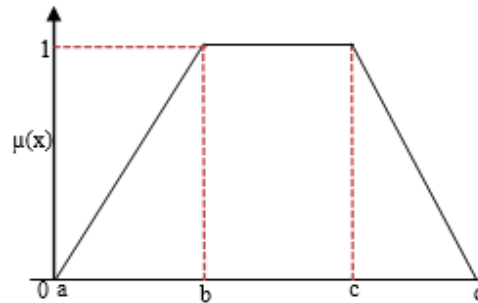
$b$  = nilai domain yang mempunyai derajat keanggotaan satu

$c$  = nilai domain terbesar yang mempunyai derajat keanggotaan nol

$x$  = nilai *input* yang akan diubah ke dalam bilangan *fuzzy*

### b. Fungsi Trapesium

Fungsi trapesium pada dasarnya seperti bentuk segitiga karena gabungan dari garis linear hanya saja terdapat beberapa nilai  $x$  yang memiliki derajat keanggotaan yang sama dengan 1.



**Gambar 2.2** Fungsi Trapesium

Fungsi Keanggotaan :

$$\mu(x) = \begin{cases} 0; & x \leq a \text{ atau } x \geq d \\ \frac{x-a}{b-a}; & a \leq x \leq b \\ 1; & b \leq x \leq c \\ \frac{d-x}{d-c}; & c < x \leq d \end{cases} \quad (2.2)$$

Keterangan :

$a$  = nilai domain terkecil yang mempunyai derajat keanggotaan nol

$b$  = nilai domain terkecil yang mempunyai derajat keanggotaan satu

$c$  = nilai domain terbesar yang mempunyai derajat keanggotaan satu

$d$  = nilai domain terbesar yang mempunyai derajat keanggotaan nol

$x$  = nilai *input* yang akan di ubah ke dalam bilangan *fuzzy*

## 2.9 Metode Mamdani

Metode ini diperkenalkan oleh Ebrahim H. Mamdani pada tahun 1975. Metode mamdani sering juga dikenal dengan nama metode MAX-MIN. Menggunakan MIN pada fungsi implikasi, dan MAX pada komposisi antar fungsi implikasi. Untuk mendapatkan *output*, diperlukan empat tahapan:

1. Pembentukan Himpunan *Fuzzy*, pada metode mamdani baik variabel *input* maupun variabel *output* dibagi menjadi satu atau lebih himpunan *fuzzy*.
2. Aplikasi Fungsi Implikasi, pada metode mamdani fungsi implikasi yang digunakan adalah min.

3. Komposisi Aturan, metode yang digunakan dalam melakukan inferensi sistem *fuzzy*, yaitu metode max (maximum).
4. Penegasan (*defuzzy*), pada komposisi aturan mamdani dengan menggunakan metode MOM. Dimana pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai rata-rata domain yang memiliki nilai keanggotaan maksimum.

### 2.10 *Fuzzy Inference System*

Motivasi utama teori logika *fuzzy* adalah memetakan sebuah ruang *input* kedalam ruang *output* menggunakan IF-THEN *rules*. Pemetaan dilakukan dalam suatu sistem inferensi *fuzzy* disebut juga *fuzzy inference engine* adalah sistem yang dapat mengevaluasi semua *rule* secara simultan untuk menghasilkan kesimpulan dan urutan bisa sembarang (Naba, 2009).

Oleh karena itu, semua aturan atau *rule* harus didefinisikan lebih dahulu sebelum membangun sebuah FIS yang akan digunakan untuk menginterpretasikan sebuah *rule* tersebut. Terdapat beberapa jenis FIS yang dikenal yaitu Mamdani, Sugeno, dan Tsukamoto (Naba, 2009).

### 2.11 *Pengujian Black Box*

Menurut Pressman (2010), *Black Box Testing* (Pengujian Kotak Hitam), juga disebut pengujian perilaku, berfokus pada persyaratan fungsional perangkat lunak. Artinya, teknik pengujian kotak hitam memungkinkan untuk membuat beberapa kumpulan kondisi masukan yang sepenuhnya akan melakukan semua kebutuhan fungsional untuk program. Pengujian kotak hitam bukan teknik alternative untuk kotak hitam. Pengujian *black box* berupaya untuk menemukan kesalahan dalam kategori:

1. Fungsi yang salah atau hilang
2. Kesalahan antarmuka
3. Kesalahan dalam struktur data akses basis data eksternal
4. Kesalahan perilaku atau kerja
5. Kesalahan inisialisasi atau penghentian.

## 2.12 Pengujian Korelasi

Analisis korelasi adalah metode statistika yang digunakan untuk menentukan suatu besaran yang menyatakan bagaimana kuat hubungan suatu variabel dengan variabel lain dengan tidak mempersoalkan apakah suatu variabel tertentu tergantung kepada variabel lain (Sekaran, 2010). Semakin nyata hubungan linier (garis lurus), maka semakin kuat atau tinggi derajat hubungan garis lurus antara kedua variabel atau lebih.

Koefisien korelasi sederhana pada umumnya disebut juga dengan Koefisien Korelasi *Pearson Product Moment* karena memiliki rumus perhitungan. Korelasi Pearson menghasilkan koefisien korelasi yang berfungsi untuk mengukur kekuatan hubungan linier antara dua variabel. Jika hubungan dua variabel tidak linier, maka koefisien korelasi Pearson tersebut tidak mencerminkan kekuatan hubungan dua variabel yang sedang diteliti, meski kedua variabel mempunyai hubungan kuat. Koefisien korelasi ini disebut koefisien korelasi Pearson karena diperkenalkan pertama kali oleh Karl Pearson tahun 1990 (Firdaus, 2009).

$$r = \frac{N \sum XY - (\sum X)(\sum Y)}{\sqrt{(N \sum X^2 - (\sum X)^2)(N \sum Y^2 - (\sum Y)^2)}} \quad (2.3)$$

**Gambar 2.3** Rumus Korelasi *Product Moment*

Keterangan:

r: koefisien korelasi

N: banyak pasangan nilai X dan Y

$\sum XY$ : jumlah dari hasil kali nilai X dan nilai Y

$\sum X$ : jumlah nilai X

$\sum Y$ : jumlah nilai Y

$\sum X^2$ : jumlah dari kuadrat nilai X

$\sum Y^2$ : jumlah dari kuadrat nilai Y

## 2.13 *Matrix Laboratory* (Matlab)

Matlab (*Matrix Laboratory*) adalah sebuah bahasa pemrograman tingkat tinggi berbasis matrix untuk komputasi teknik. Program ini merupakan produk komersial dari perusahaan Mathworks, Inc yang dikembangkan menggunakan

Bahasa C++ dan Assembler. Di dalam Matlab terdapat banyak *toolbox*, yaitu kumpulan fungsi komprehensif yang digunakan untuk menyelesaikan kelas masalah dan memungkinkan kita mempelajari dan mengaplikasikan teknologi tertentu. Berbagai *toolbox* dalam matlab antara lain *signal processing*, *control system*, *neural network*, *fuzzy logic*, *wavelets*, *simulation*, *statistics*, *optimization*, *bioinformatics*, *genetic algorithm*, dan lain-lain (Santosa, 2007).

Pada penelitian ini menggunakan bantuan matlab untuk simulasi grafik dan *rule* karena pada matlab menyediakan perkakas untuk membuat *fuzzy inference system* (FIS) yaitu *Fuzzy Logic Toolbox* yang di dalamnya terdapat *Graphical User Inferensi* (GUI) untuk merancang FIS (Naba, 2009), GUI tersebut adalah:

1. FIS Editor
2. *Membership Function Editor*
3. *Rule Editor*
4. *Rule Viewer*