# BAB II

# TINJAUAN PUSTAKA

## 2.1 Teori Dasar

Teori dasar adalah gambaran terhadap seperangkat kumpulan konsep atau kontruk, definisi, dan proposisi yang terkait secara sistematis untuk menjelaskan tentang suatu fenomena atau gejala. Logika *fuzzy*, salah satu cara pengambilan keputusan berbasis aturan yang bertujuan untuk memecahkan masalah, dimana sistem tersebut sulit untuk dimodelkan atau terdapat ambigu dan ketidakjelasan.

### 2.1.1 Kecerdasan Buatan (*Artificial Intelligence*)

*Artificial Intelligence* dapat dikatakan kecerdasan yang dimasukkan ke dalam suatu mesin atau komputer agar dapat melakukan pekerjaan seperti manusia. Salah satu bidang yang menggunakan *Artificial intelligence* adalah logika *fuzzy*. Menurut (Budiharto & Suhartono, 2014: 2) kecerdasan buatan atau *Artificial Intelligence* (AI) merupakan bidang ilmu komputer yang mempunyai peran penting di era kini dan masa yang akan datang. Sedangkan menurut (Suyanto, 2011: 2) Sebagian kalangan menerjemahkan *Artificial Intelligence* sebagai kecerdasan buatan, kecerdasan *artificial*, *inteligensia artificial*, atau *intelijensia* buatan. Kecerdasan buatan juga berdasarkan pemikiran bahwa komputer bisa melakukan penalaran secara logis dan juga bisa melakukan aksi secara rasional berdasarkan hasil penalaran tersebut.

### 2.1.2 Sistem Pakar

Menurut (Suputra, 2012) sistem pakar terdiri dari dua bagian utama yaitu basis pengetahuan (*knowledge* b*ase*) dan penalaran (*reasoning*). *Knowledge base* dalam sistem pakar mengandung pengetahuan yang bersumber dari pengetahun seorang (atau sekelompok) pakar, buku, jurnal, dan sebagainya, yang berhubungan dengan bidang tertentu yang direpresentasikan dengan format tertentu yang dapat diterima oleh komputer. Pekerjaan pengumpulan dan representasi pengetahun ini dilakukan oleh *knowledge engineer*. Sementara proses penalaran diimplementasikan berdasarkan konsep dan metode yang disesuaikan dengan kebutuhan atas pemecahan masalah yang dihadapi.

Pengetahuan dan informasi (fakta) tidak selalu seratus persen tepat. Bahkan sangat mungkin pada suatu kasus, pengetahuan yang diperoleh tidak lengkap. Untuk menghadapi ketidakyakinan atas suatu pengetahuan maupun fakta, suatu aturan (representasi atas pengetahuan) perlu diasosiasikan pada suatu *confident factor* atau suatu bobot tertentu. Metode paling naif dalam penalaran atas ketidakpsatian adalah metode *Certainty Factor* (*CF*).

Sistem pakar berbasis aturan dapat dibangun dengan dua cara, yaitu dengan membangun sistem keseluruhan dari rancangan dasar menggunakan bahasa pemrograman atau membangunnya menggunakan perangkat lunak khusus yang disebut dengan ”*tool*” atau ”*shell*”. *Shell* sistem pakar berbasis aturan merupakan suatu sistem yang dibangun berdasarkan sekumpulan metode-metode penalaran tanpa tergantung pada domain tertentu. Membangun sistem pakar berbasis aturan menggunakan *shell* ini menawarkan keuntungan yang signifikan.

Sistem dapat dibangun untuk melakukan tugas tertentu (pada *domain* tertentu) dengan memasukkan pengetahuan yang dibutuhkan ke dalam *shell*. Mesin inferensi yang terinstalasi dalam *shell* akan mengolah pengetahuan tersebut untuk melakukan tugasnya. Dengan adanya *shell* ini, pembangunan sebuah sistem pakar dapat difokuskan pada pengumpulan dan representasi pengetahuan saja. Selain itu, *shell* sistem pakar dapat digunakan dalam pembelajaran bagaimana proses pemecahan masalah dilakukan melalui serangkaian inferensi simbolik.

Sistem pakar terdiri dari beberapa komponen sebagai berikut:

1. *User interface*

*User-interface* merupakan sebuah mekanisme bagaimana para pengguna berinteraksi dengan sistem pakar.

2. *Explanation facility*

Fasilitas ini memberikan informasi kepada pengguna mengenai jalannya penalaran sehingga dihasilkan suatu keputusan.

3. *Working memory*

*Working-memory* menyimpan fakta-fakta yang digunakan pada aturan selama proses penalaran berlangsung.

4. *Inference engine*

*Inference-engine* melakukan proses inferensi dengan cara memutuskan aturan mana yang memenuhi kondisi fakta atau objek, mempertimbangkan prioritas atas aturan-aturan yang memenuhi tadi, lalu mengeksekusi aturan yang memiliki prioritas tertinggi dari sederetan aturan tadi.

5. *Agenda*

Agenda berisi aturan-aturan yang memiliki prioritas yang dibuat oleh *inference engine*, di mana aturan-aturan tersebut dipilih (dibangkitkan) berdasarkan fakta-fakta yang diketahui yang memenuhi kondisi aturan tersebut.

6. *Knowledge acquisition facility*

Fasilitas ini mengakomodasi kebutuhan akan akuisisi dan representasi pengetahuan dengan cara yang lebih mudah dibandingkan dengan mengolah pengetahuan secara eksplisit dengan cara mengkode basis pengetahuan.

### 2.1.3 Jaringan Saraf Tiruan

Menurut (Indah & Setiawan, 2011: 23) Jaringan saraf tiruan (JST) (*artificial neural network* (ANN)/ *simulated neural network* (SNN)/ *neural network* (NN)) adalah jaringan yang terdiri atas sekelompok unit pemroses kecil yang dimodelkan berdasarkan jaringan saraf manusia.

Adapun alasan-alasan yang mendasari banyak peneliti mempelajari jaringan saraf tiruan adalah sebagai berikut:

1. Kemampuan untuk menyelesaikan data yang kompleks atau yang tidak tepat.
2. Kemampuan untuk menemukan pola (*pattern*) dan *trend* yang terlalu kompleks untuk dikenali oleh manusia atau teknik komputasi lainnya.

Terdapat beberapa kelebihan dari penerapan jaringan saraf tiruan, antara lain:

1. Dapat menyelesaikan tugas yang tidak dapat dikerjakan oleh program linear.

2. Saat sebuah elemen dari JST gagal, jaringan dapat terus berjalan tanpa masalah.
3. Sebuah JST dapat belajar (dilatih) dan karenanya tidak perlu di-program ulang.
4. Dapat diterapkan pada sembarang aplikasi.
5. Dapat diterapkan tanpa banyak kendala.

Namun, selain beberapa kelebihan yang telah disebutkan sebelumnya, penerapan jaringan saraf tiruan juga memiliki beberapa kekurangan, seperti berikut:

1. JST memerlukan pelatihan (*training*) agar dapat beroperasi.
2. Memerlukan waktu proses yang lama untuk jaringan yang besar.

Secara umum, arsitektur jaringan saraf tiruan dibedakan menjadi dua, yakni antara lain jaringan saraf *singlelayer* dan jaringan saraf *multi-layer*.

1. Jaringan saraf *single-layer*
    a. *Neuron-neuron* dikelompokkan menjadi dua bagian, yakni unit-unit *input* dan unit-unit *output*.
    b. Unit-unit *input* menerima masukan dari luar, unit-unit *output* mengeluarkan respon dari jaringan sesuai dengan masukannya.
2. Jaringan saraf *multi-layer*
    a. *Neuron-neuron* dikelompokkan menjadi tiga bagian, yakni unit-unit *input*, unit-unit *hidden*, dan unit-unit *output*.

b. Jumlah unit-unit *hidden* tergantung pada kebutuhan. Semakin kompleks jaringan, unit-unit *hidden* yang dibutuhkan semakin banyak, demikian pula jumlah *layer*-nya.

### 2.1.4 Teori Khusus

#### 2.1.4.1 Data Runtun Waktu

Pengelompokan data terdiri dari beberapa bentuk, pengelompokan data dapat dilihat dari beberapa sudut pandang. Salah satu sudut pandang tersebut yaitu ekonometrika, dilihat dari sudut ekonometrika bahwa data dapat dikelompokkan kedalam tiga kelompok yaitu (Widarjono, 2013) :

1. Data silang (*cross section*) terdiri dari beberapa objek data pada suatu waktu, misalnya data pada suatu restoran akan terdiri dari data penjualan, data pembelian bahan baku, data jumlah karyawan dan data-data lainnya.
2. Data panel (*pooled data*) adalah data yang menggabungkan antara data runtun waktu (*time series*) dan data silang (*cross section*).
3. Data runtun waktu (*time series*) merupakan data yang terdiri atas objek tetapi meliputi beberapa periode waktu misalnya harian, mingguan, tahunan, dan lain-lain.

Data runtun waktu juga sangat berguna bagi pengambil keputusan untuk memperkirakan kejadian dimasa yang akan mendatang, karena diyakini pola perubahan data runtun waktu beberapa periode masa lampau akan kembali

terulang dimasa kini. Jika kita merujuk sesuatu hal yang terjadi didunia ini suatu pola misalnya model bisa saja berulang kembali dimasa yang akan datang. Pola perubahan inilah yang akan dianalisa sehingga dengan mempelajari pola tersebut kita mendapat sesuatu makna untuk digunakan sebagai pendukung pengambilan keputusan.

### 2.1.4.2 Logika Fuzzy

Menurut (Sutojo, T ; Mulyanto, Edy; Suhartono, 2011: 211) konsep tentang logika *fuzzy* diperkenalkan oleh Prof. Lotfi Astor Zadeh pada 1962, Logika *fuzzy* adalah metodologi sistem control pemecahan masalah, yang cocok untuk diimplementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, *embedded system*, jaringan *PC*, *multi-channel* atau *workstation* berbasis akuisisi *data*, dan sistem *control*. 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, sistem ini dapat mempunyai nilai keanggotaan 0 atau 1. Akan tetapi, dalam logika *fuzzy* memungkinkan nilai keanggotaan berada di antara 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.

*Fuzzy logic* dianggap mampu untuk memetakan suatu *input* kedalam suatu *output* tanpa mengabaikan faktor–faktor yang ada. *Fuzzy logic* diyakini dapat sangat fleksibel dan memiliki toleransi terhadap *data-data* yang ada. *Fuzzy logic*,

yang dalam bahasa Indonesia dapat diartikan sebagai Logika Kabur atau Logika Samar, dapat dikatakan sebagai “logika baru yang sudah lama”. Hal ini karena ilmu tentang *Fuzzy logic* secara modern dan metodis ditemukan pada tahun 1965, namun konsep *Fuzzy logic* sudah melekat pada diri manusia, sejak manusia ada. Konsep *Fuzzy logic* dapat dengan mudah kita temukan pada perilaku manusia dalam kesehariannya.

#### 2.1.4.3 Himpunan *Fuzzy*

Pada dasarnya, teori himpunan *fuzzy* merupakan perluasan dari teori himpunan kelasik. Menurut Tettamanzi dalam teori himpunan *fuzzy* merupakan kerangka matematis yang digunakan untuk mempresentasikan ketidakpastian, ketidakjelasan, ketidaktepatan, kekurangan informasi, dan kebenaran informasi. Sedangkan menurut Ross dalam pada teori himpunan *fuzzy*, komponen utama yang sangat berpengaruh adalah fungsi keanggotaan yaitu fungsi keanggotaan yang merepresentasikan derajat kedekatan suatu objek tertentu.

Himpunan *fuzzy* memiliki dua atribut yaitu:

1. *Linguistik,* yaitu nama suatu kelompok yang mewakili suatu keadaan tertentu dengan menggunakan bahasa alami, misalnya DINGIN, SEJUK, PANAS mewakili variabel temperatur.
2. *Numeris*, yaitu suatu nilai yang menunjukkan ukuran dari suatu variabel, misalnya 10, 35, 40, dan sebagainya.

### 2.1.4.4 Alasan Penggunaan Metode *Fuzzy Logic*

Dalam penelitian ini, penulis menggunakan metode *fuzzy logic*. Ada beberapa alasan penulis memilih menggunakan metode ini, antara lain sebagai berikut (Sri, kusumadewi, Sri hartati, Agus Harjoko, 2006: 2):

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

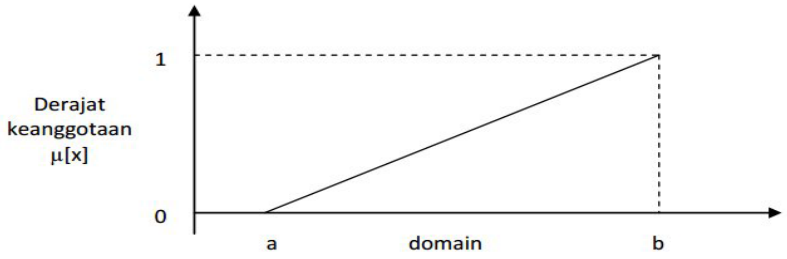
### 2.1.4.5. Fungsi Keanggotaan

Fungsi keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data kedalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki *interval* antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Apabila U menyatakan himpunan *universal* dan A adalah

himpunan fungsi *fuzzy* dalam U, maka A dapat dinyatakan sebagai pasangan terurut. Ada beberapa fungsi yang biasa digunakan (Sri, kusumadewi, Sri hartati, Agus Harjoko, 2006: 9-20).

1. Representasi *Linear*

Pada representasi *linear*, pemetaan *input* ke derajat keanggotaannya digambarkan 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. Seperti terlihat pada gambar 2.1.




**Gambar 2.1** Kurva Representasi *Linear* Naik

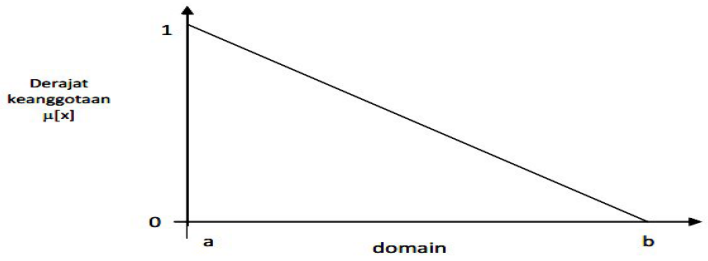
Fungsi keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \le a \\ (x-a)/(b-a); & a \le x \le b \\ l; & x = b \end{cases}$$ …**Rumus 2. 1** Fungsi Keanggotaan

Representasi linear (1)

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. Seperti terlihat pada gambar 2.2.




**Gambar 2.2** Representasi *Linear* Turun

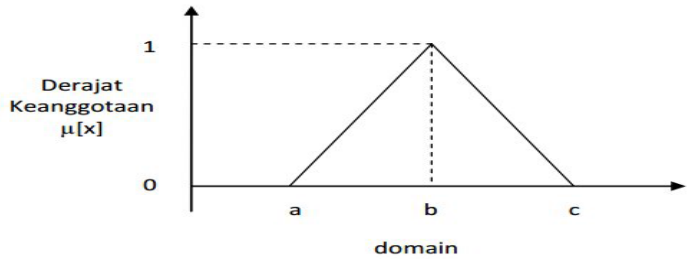
Fungsi keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \leq a \\ (b-x)/(b-a); & z \leq x \leq b \\ l; & x = b \end{cases} \text{...Rumus 2.2 Fungsi Keanggotaan}$$

Representasi linear (2)

2. Representasi kurva segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (*linear*). Seperti terlihat pada gambar 2.3.




**Gambar 2. 3** Kurva Segitiga

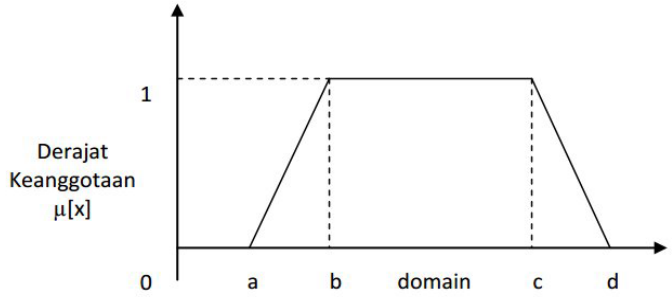
Fungsi Keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \leq a \; atau \; x \geq c \\ (x-a)/(b-a); & a \leq x \leq b \\ (b-x)/(c-b); & b \leq x \leq c \end{cases} \quad \text{... Rumus 2.3 Fungsi Keanggotaan}$$

Representasi Segitiga

3. Representase kurva trapesium

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada titik yang memiliki nilai keanggotaan 1. Seperti terlihat pada gambar 2.4.




**Gambar 2. 4** Representasi Kurva Trapesium

Fungsi keanggotaan:

$$\boldsymbol{\mu[x]} = \begin{cases} \mathbf{0}; & \boldsymbol{x \le a \; atau \; x \ge d} \\ \boldsymbol{(x-a)/(b-a)}; & \boldsymbol{a \le x \le b} \\ \mathbf{1}; & \boldsymbol{b \le x \le c} \\ \boldsymbol{(d-x)/(d-c)}; & \boldsymbol{c \le x \le d} \end{cases}$$

**Rumus 2.4** Fungsi Keanggotaan

Representasi Kurva Trapesium

### 2.1.4.6 Algoritma *Automatic Clustering*

Algoritma ini digunakan untuk menentukan interval, menurut *algoritma automatic clustering* disajikan sebagai berikut:

1. Mengurutkan data *numeric* dalam urutan menaik memilik n data numeric yang berbeda. Diasumsikan bahwa data *ascending* urutan tanpa data ganda akan ditampilkan sebagai berikut

   $d_1, d_2, d_3, \ldots, d_n$.

   Berdasarkan barisan di atas, dihitung nilai dari "*average_dif*" sebagai berikut:

$$\boldsymbol{average_dif} = \frac{\sum_{i=1}^{n-1} d_{i+1} - d_i}{n-1}$$ …… **Rumus 2.5** Rata-rata data

2. Mengambil data angka pertama (data terkecil dalam barisan data terurut naik) ke dalam pengelompokan sekarang. Berdasarkan nilai dari "*average_dif*", ditentukan apakah data angka mengikuti data pada pengelompokan sekarang pada barisan data terurut naik dapat diletakkan pada pengelompokan sekarang atau diletakkan pada pengelompokan baru berdasarkan prinsip berikut:

   a. Diasumsikan bahwa saat ini *cluster* adalah *cluster* pertama dan hanya ada satu data $d_1$ di dalamnya dan menganggap bahwa $d_2$ adalah data yang berdekatan dengan $d_1$, ditampilkan sebagai berikut:

      $\{d_1\}, d_2, d_3, \ldots, d_n$.

      Jika $d_2$-$d_1 \leq$ *average_dif*, maka $d_2$ diletakkan ke dalam pengelompokan sekarang yang mana $d_1$ termasuk. Sebaliknya dibentuk kelompok baru untuk $d_2$ dan biarkan *cluster* baru yang baru dibangun dimana $d_2$ termasuk ke dalam *cluster* sekarang.

   b. Diasumsikan bahwa *cluster* yang sekarang bukan yang pertama *cluster* dan hanya ada satu data $d_j$ di *cluster* saat ini. Diasumsikan bahwa $d_k$ adalah data yang berdekatan di sebelah $d_j$ dan mengganggap bahwa $d_i$ adalah data terbesar di *cluster* yang merupakan *anteseden cluster cluster* saat ini, akan ditampilkan sebagai berikut:

      $\{d_1, \ldots\}, \ldots, \ldots, \ldots, \{\ldots, d_1\}, \{dj\}, d_k, \ldots d_n$.

      Jika $d_k$ - $d_j \leq$ *average_dif* dan $d_k$–$dj < dj - d_i$, maka taruh $d_k$ ke

*cluster* yang saat ini milik $d_j$. Jika tidak, hasilkan suatu *cluster* baru untuk $d_k$ dan biarkan *cluster* yang baru dihasilkan dengan $d_k$ termasuk menjadi *cluster* saat ini.

c. Diasumsikan bahwa cluster yang sekarang bukan *cluster* yang pertama dan ada lebih dari satu data dicluster saat ini. Diasumsikan bahwa $d_i$ adalah data terbesar di *cluster* saat ini dan diasumsikan bahwa $d_j$ adalah data yang berdekatan di sebelah $d_i$, yang ditampilkan sebagai berikut:

$\{d_1, \ldots\}, \ldots, \{\ldots\}, \{\ldots., d_i\}, d_j, \ldots, d_n.$

Jika $d_j - d_i \leq$ *average_dif* dan $d_j - d_i <$ *cluster_dif*, maka $d_j$ diletakkan dalam *cluster* yang saat ini terdapat $d_i$. Jika tidak, hasilkan *cluster* baru untuk $d_j$ dan biarkan *cluster* baru yang dihasilkan sehingga $d_j$ termasuk dalam cluster saat ini, dimana "*cluster_dif*" menunjukan perbedaan rata-rata jarak antara setiap pasangan data yang berdekatan dalam *cluster* dan nilai dari *cluster_dif* dihitung sebagai berikut:

$$cluster_{dif} = \frac{\sum_{i=1}^{n-1} c_{i+1} - c_i}{(n-1)}$$

Dengan $c_1,0$, $c_2$, $0$, ... dan $c_{n,0}$ menggambarkan data dalam *cluster* saat ini.

3. Berdasarkan hasil pengelompokan yang diperoleh pada point 2, sesuaikan isi dari kelompok ini menurut prinsip berikut:
   a. Jika sebuah kelompok memiliki lebih dari dua data, maka kita menjaga data terkecil, menjaga data terbesar dan menghapus yang lain.
   b. Jika sebuah *cluster* memiliki tepat dua data, maka kita tinggalkan

(tidak berubah).

c. Jika sebuah *cluster* hanya memiliki satu data dq, maka kita meletakkan nilai-nilai dari "$d_q$ – *averange_dif*" dan "$d_q$ + *average_dif*" ke dalam *cluster* dan menghapus $d_q$ dari *cluster* ini. Terlebih lagi jika situasi berikut terjadi, *cluster* perlu disesuaikan lagi:

1) Jika situasi terjadi di *cluster* pertama, maka kita menghapus nilai dari "$d_q$ – *average_dif*" sebagai ganti dari $d_q$ dari cluster ini.

2) Jika situasi terjadi di *cluster* terakhir, maka kita menghapus nilai dari "$d_q$ + *average_dif*" sebagai ganti dari $d_q$ dari *cluster* ini.

3) Jika nilai dari "$d_q$ – *average_dif*" lebih kecil dari pada nilai terkecil dalam *cluster* yang terdahulu, maka semua tindakan dari Prinsip 3 dibatalkan.

4. Asumsikan bahwa hasil *cluster* yang diperoleh pada Langkah 3 adalah ditampilkan sebagai berikut:

$\{d_1, d_2\}, \{d_3, d_4\}, \{d_5, d_6\}, \ldots \{d_{n-1}, d_n\}$

Mengubah kelompok ini ke dalam interval yang bersebelahan dengan sub-langkah berikut:

Selanjutnya, mentransformasikan kluster-kluster tersebut kedalam interval yang bersebelahan dengan sub langkah berikut:

a. Mentransformasikan kluster pertama $\{d_1, d_2\}$ dalam interval $[ d_1, d_2]$

b. Jika interval yang ada $[ d_1, d_2]$ dan kluster yang ada adalah $\{d_3, d_4\}$, maka:

1) Jika $d_2 \geq d_3$, maka {$d_3$, $d_4$} dalam kluster yang ada ditransformasi ke dalam interval [ $d_3$, $d_4$] dan kluster selanjutnya {$d_5$, $dn$} menjadi kluster yang ada.
2) Jika $d_2 < d_3$, maka transformasikan *cluster* {$d_3$, $d_4$} dalam interval [$d3$, $d_4$]. Selanjutnya, [$d_3$, $d_4$] menjadi interval saat ini dan kluster berikutnya adalah {$d_5$, $dn$} menjadi klusternya. Jika nilai dalam interval adalah [ $d_3$, $d_4$] dan nilai klusternya adalah {$dn$}, maka transformasikan interval [ $d_3$, $d_4$] ke dalam [$d_4$, $dn$]. Kemudian [$d_4$, $dn$] menjadi interval dan kluster berikutnya menjadi kluster yang ada.

c. Melakukan pemeriksaan berulang terhadap interval yang ada hingga semua kluster ditransformasikan menjadi interval

5. Untuk setiap interval yang diperoleh dari langkah 4, kemudian bagi setiap interval tersebut dalam *p* sub interval dimana $p \geq 1$

#### 2.1.4.7 *Automatic Clustering Fuzzy Logic Relationship* (ACFLR)

Dalam bagian ini, disajikan metode untuk peramalan didasarkan pada metode *automatic clustering and fuzzy logic relationship*. Setelah mendapat interval dengan menggunakan automatic clustering technique, selanjutnya nilai peramalan dapat dihitung menggunakan langkah-langkah sebagai berikut:

1. Menerapkan metode *automatic clustering* untuk cluster data historis ke interval dan untuk menghitung titik tengah tiap interval.

2. Mengasumsikan bahwa terdapat *n* interval $u_1$, $u_2$, *...*, *dan* $u_n$, kemudian mendefinisikan setiap *fuzzy* set $A_i$, dimana $1 \leq i \leq n$, sebagai berikut:

$$A_1 = \frac{1}{u_1} + \frac{0.5}{u_2} + \frac{0}{u_3} + \cdots \frac{0}{u_{n-1}} + \frac{0}{u_n}$$

$$A_2 = \frac{0.5}{u_1} + \frac{1}{u_2} + \frac{0}{u_3} + \cdots \frac{0}{u_{n-1}} + \frac{0}{u_n}$$

$$\vdots$$

$$\boldsymbol{A_n = \frac{0}{u_1} + \frac{0}{u_2} + \frac{0}{u_3} + \cdots \frac{0}{u_{n-1}} + \frac{1}{u_n}}$$ **... Rumus 2.6** Rumus Pembentukan *fuzzy* set

3. Fuzzifikasi setiap data dalam sejarah data menjadi himpunan *fuzzy*. Jika milik data $u_1$, dimana $1 \leq i \leq n$, kemudian data difuzzifikasi ke $A_i$.
4. Membuat *fuzzy logic relationship* didasarkan pada fuzzifikasi data historis yang diperoleh *t+1* adalah $A_i$ dan $A_k$, masing-masing kemudian membangun *fuzzy logic relationship* "$A_i \rightarrow A_k$", dengan $A_i$ dan $A_k$ berturut-turut disebut keadaan saat ini dan keadaan berikutnya dari *fuzzy logic relationship*. Berdasarkan pada keadaan saat ini pada *fuzzy logic relationship*, *fuzzy logic relationship* dibagi ke dalam *fuzzy logic relationship group*, di mana *fuzzy logic relationship* yang memiliki keadaan saat ini yang sama dimasukkan ke dalam *fuzzy logic relationship group* yang sama.
5. Menghitung perkiraan / peramalan dengan prinsip berikut
   a. Jika fuzzifikasi nilai dari tahun *t* adalah $A_j$ dan hanya ada satu *fuzzy logic relationship* pada *fuzzy logic relationship group* yang memiliki keadaan saat ini $A_j$ ditunjukkan sebagai berikut:

      $A_i \rightarrow A_k$,

      Kemudian peramalan nilai pada tahun *t+1* adalah $m_k$, dimana $m_k$ adalah

titik tengah dari interval $u_k$ dan nilai keanggotaan maksimum dari himpunan fuzzy $A_k$ terjadi pada interval $u_k$.

b. Jika fuzzifikasi nilai dari tahun *t* adalah $A_j$ dan ada *fuzzy logic relationship* berikut dalam *fuzzy logic rellationship group* yang memiliki keadaan sekarang $A_j$, ditunjukkan sebagai berikut:

$A_i \rightarrow A_{k1}\ (x_1), A_{k2}\ (x_2), \ldots A_{kp}\ (x_p)$.

Kemudian peramalan nilai dari tahun *t+1* dihitung sebagai berikut:

$$\frac{x_1 m_{k1} + x_2 m_{k2} \ldots + x_{1p} m_{kp}}{x_1 + x_2 \ldots + x_p}$$ …. **Rumus 2.7** Rumus Peramalan

Dimana $x_i$ menggambarkan angka dari *fuzzy logic relationship* "$A_j \rightarrow A_{ki}$" pada *fuzzy logic relationship group*, $1 \leq i \leq p$; $m_{k1}$, $m_{k2}$...dan $m_{kp}$ adalah titik tengah dari interval-interval $u_{k1}$, $u_{k2}$, ... dan $u_{kp}$ berturut-turut, dan nilai keanggotaan maksimum dari himpunan fuzzy $A_{k1}$, $A_{k2}$, ... dan $A_{kp}$ terjadi pada interval $u_{k1}$, $u_{k2}$, ... dan $u_{kp}$ berturut-turut.

c. Jika fuzzifikasi nilai dari tahun *t* adalah $A_j$ dan ada *fuzzy logic relationship* dalam *fuzzy logic relationship group* yang memiliki keadaan sekarang $A_j$, yang digambarkan seperti dibawah ini:

$A_i \rightarrow \neq$

Dimana simbol "≠" menunjukkan sebuah nilai yang tidak diketahui, maka perkiraan nilai pada tahun *t+1* adalah $m_j$, dimana $m_j$ adalah titik tengah dari Interval $u_j$ dan nilai keanggotaan maksimal dari himpunan fuzzy $A_j$ terjadi pada $u_j$.

### 2.1.4.8 *Fuzzy Time Series*

*Fuzzy time series* (FTS) adalah metode peramalan data yang menggunakan prinsip-prinsip *fuzzy* sebagai dasarnya. Sistem peramalan dengan FTS menangkap pola dari data yang telah lalu kemudian digunakan untuk memproyeksikan data yang akan datang. Pertama kali dikembangkan oleh Q. Song dan B.S Chissom pada tahun 1993. Metode ini sering kali digunakan oleh para peneliti untuk menyelesaikan masalah peramalan. Metode peramalan *fuzzy time series* adalah metode peramalan yang menggunakan prinsip-prinsip *fuzzy* sebagai dasarnya. Konsep dasar *fuzzy time series* yang diperkenalkan oleh Song dan Chissom (1993a, 1993b, 1994) dengan nilai *fuzzy time series* direpresentasikan dengan himpunan *fuzzy* (Chen 1998): Didefenisikan $U$ adalah semesta pembicaraan dengan $U = \{ u_1, u_2, u_3, \dots , u_n \}$

Sebuah himpunan fuzzy dalam semesta pembicaraan $U$ dapat direpresentasikan sebagai berikut:

$A = f_A(u_1)/u_1 + f_A(u_2)/u_2 + \dots + f_A : U \rightarrow [0, 1]$, $f_A(u_i)$ merupakan tingkat keanggotaan $u_i$ dalam himpunan fuzzy A, dan $1 \leq i \geq n$.

(Anwary & Ahmad, 2011) dalam penelitiannya untuk meramal kurs Rupiah terhadap terhadap Dollar Amerika menggunakan metode *fuzzy time series*. Dalam peramalan tersebut dilakukan upaya untuk memprediksi besarnya kurs untuk satu hari ke depan. Permasalahan yang dihadapi adalah cara untuk memprediksi besarnya kurs yaang menghasilkan nilai prediksi dengan tingkat kesalahan minimal. Penelitian ini menggunakan *fuzzy time series* (FTS) untuk memprediksi jumlah penumpang pesawat udara. Hasilnya berupa jumlah penumpang pesawat

udara yang terprediksi sampai satu bulan kedepan. Tingkat keakuratan hasil prediksi diukur dengan nilai AFER (*Average Forecasting Error*).

Algoritma *fuzzy time series* dalam menyelesaikan masalah prediksi adalah sebagai berikut :

1. Menentukan semesta pembicaraan $U$. $U$ = [$Dmin - D1$, $Dmax + D2$] dengan $Dmin$ dan adalah nilai terkecil dan nilai terbesar dari semesta pembicaraan $U$, sedangkan $D1$ dan $D2$ merupakan bilangan positive yang ditentukan oleh user.
2. Membagi semesta pembicaraan $U$ menjadi beberapa interval dengan panjang yang sama.
3. Mendefinisikan himpunan fuzzy $A$ dari semesta pembicaraan $U$ berdasarkan interval partisi yang telah ditentukan. Sebuah himpunan fuzzy $A$ dalam semesta pembicaraan $U$ dapat dinyatakan dengan persamaan :

   $$A = (u1)/u1 + (u2)/u2 + \cdots + (un)/un$$

   dimana $fA$ adalah fungsi keanggotaan dari himpunan fuzzy $A$, dan $fA : U \rightarrow [0,1]$;$fA(u1)$ merupakan derajat keanggotaan dari $ui$ dalam himpunan fuzzy $A$, dan $1 \leq i \leq n$.
4. Fuzzifikasi data dengan menemukan derajat keangotaan di setiap himpunan fuzzy $Ai$ ($1 \leq i \leq n$). Setiap data difuzzifikasi berdasarkan derajat keanggotaan yang tertinggi.
5. Menentukan *fuzzy logical relationship* dari hasil fuzzifikasi data yang selanjutnya dikelompokan menjadi *fuzzy logical relationship groups* (FLRG).

6. Menentukan peramalan dalam bentuk himpunan *fuzzy* ($t$) dan defuzzifikasikan hasil peramalan, dengan prinsip sebagai berikut :

   Prinsip 1. Jika himpunan *fuzzy* sekarang $Ai$ dan ada *fuzzy logic relationship* $Ai \rightarrow Aj$, maka hasil peramalanya adalah $mj$ atau titik tengah interval $uj$.

   Prinsip 2. Jika himpunan fuzzy sekarang $Ai$ dan fuzzy logic relationship dalam *fuzzy logic relationship group* adalah $Ai \rightarrow A_1,A_2,\ldots,Ak$, maka nilai peramalanya adalah $m1+m2+\cdots+mk/k$ dengan $m1,m2,\ldots,mk$ secara berturut-turut adalah titik tengah dari interval $u1,u2,\ldots,uk$, dengan kata lain nilai ramalanya adalah rata rata dari titik tengah intervalnya.

   Prinsip 3. Jika himpunan fuzzy sekarang $Ai$ dan fuzzy logic relationship adalah kosong, misalkan ($Ai \rightarrow \neq$), maka hasil peramalan adalah titik tengah interval $ui$ atau nilai maksimum *ui*.

### 2.1.2.9 *Mean Absolute Percentage Error* (MAPE)

*Mean absolute percentage error* (MAPE) menghitung deviasi antara data aktual dengan nilai peramalan kemudian dihitung persen rata-ratanya. Menurut Zainun dan Majid dalam penelitian (Endaryati & Kurniawan, 2015), suatu model mempunyai kinerja sangat bagus jika nilai MAPE dibawah 10% dan mempunyai kinerja bagus jika nilai MAPE berada di antara 10% dan 20%. Mape dapat dihitung dengan rumus sebagai berikut:

MAPE= $\left( \frac{1}{n} \sum_{t=1}^{n} \left| \frac{y_t - V_t}{y_t} \right| \times 100\% \right)$ …. **Rumus 2.8** MAPE

Dimana n adalah jumlah sampel, $y_t$ adalah nilai peramalan pada periode ke-t, $v_t$ adalah nilai aktual pada periode ke-t

## 2.2 Variabel

Dalam penelitian ini variabel penelitian terdiri dari 2 jenis variabel yaitu:

1. Variabel Masukan ialah berupa data jumlah penumpang tiap tahun selama 10 tahun yang di inisialisasi dengan hurup $d_1$, $d_2$,...,$d_n$.
2. Variabel Keluaran (Target) data jumlah penumpang pertahun dari periode tahun sebelumnya, tahun ke- t+1 dijadikan sebagai variabel keluaran atau lebih dikenal dengan nama target dalam peramalan.

## 2.3 Software Pendukung

### 2.3.1 *Microsoft Excel*

*Microsoft Excel* adalah sebuah aplikasi lembar kerja *spreadsheet* keluaran Microsoft Corporation yang dapat dijalankan pada *Microsoft Windows* dan *Mac* OS. Aplikasi ini mempunyai fitur kalkulasi dan pembuatan grafik yang mumpuni sehingga *Microsoft Excel* menjadi salah satu aplikasi yang paling populer digunakan di dalam komputer mikro sampai dengan sekarang. Aplikasi ini juga merupakan aplikasi *spreadsheet* yang paling banyak digunakan oleh semua kalangan karena memiliki banyak fitur unggulan dan kompatibel dengan *platform* PC berbasis *Windows* maupun *platform Macintosh* berbasis OS.

*Microsoft Excel* fungsinya lebih banyak ke menghitung dan mengolah data. Dalam mengolah data biasanya menggunakan *formulas*, secara umum *formulas* adalah kumpulan fungsi-fungsi rumus tertentu pada *Microsoft excel*, seperti penjumlahan dasar, perkalian, pengurangan dll. Dalam menu ini fungsi-fungsi dikelompokkan berdasarkan kriteria, seperti gambar 2.5 berikut ini :

**Gambar 2.5** Menu *Excel* Tab *Formulas*

Berdasarkan gambar 2.5 diatas terdapat 10 *toolbar* dasar yang semuanya memiliki fungsi untuk meng-eksekusi sebuah fungsi. Berikut penjelasannya sesuai dengan nomor urut pada gambar diatas (Jarot S. Shenia A. Sudarma S, 2012) :

1. *Insert Function* adalah untuk mengedit formula dalam sel dengan memilih fungsi dan mengedit *argument*.
2. *Auto Sum* adalah untuk menampilkan hasil perhitungan sederhana seperti penjumlahan, pengurangan, rata-rata dll.
3. *Recently Used* adalah untuk menggunakan formula yang pernah atau sering digunakan, rumus-rumus yang sering digunakan akan tersimpan di sini.
4. *Financial* adalah untuk melakukan perhitungan dengan rumus-rumus *financial*, seperti menghitung nilai konversi mata uang asing dan perhitungan yang berhubungan dengan *finance*.

5. *Logical* adalah untuk melakukan perhitungan dengan rumus-rumus logika, seperti *AND, OR, IF* dll.
6. *Text* adalah untuk memodifikasi format teks pada sel, seperti membuat huruf kecil menjadi *capital*, menghilangkan teks dalam sel dll.
7. *Date & Time* adalah untuk memodifikasi nilai dalam sel menjadi format waktu, seperti tanggal, bulan, tahun, jam, menit dan detik.
8. *Lookup & Reference* adalah untuk memasukkan formula yang berkaitan dengan referensi atau menyorot suatu file dalam sel.
9. *Math & Trig* adalah untuk melakukan perhitungan dengan rumus-rumus *trigonometri*, seperti menghitung sinus, cosinus dll.
10. *More Function* adalah untuk melakukan perhitungan dengan fungsi lainnya sesuai dengan kebutuhan.

## 2.4 Penelitian Terdahulu

Berikut merupakan penelitian sebelumnya yang digunakan untuk dapat dijadikan bahan pertimbangan dan diharapkan sehingga dapat membantu dalam penelitian ini adalah:

Dalam penelitian (Wang & Liu, 2015) mengusulkan model peramalan *fuzzy* baru yang dirancang dengan menggunakan dua teknik utama, yaitu klasifikasi *clustering* dan *axiomatic fuzzy set* (AFS). Pertama, algoritma *clustering* digunakan untuk menghasilkan interval berbasis *clustering*. Kedua, tren *fuzzy* berlabel data *training set* dibangun berdasarkan hubungan logika *fuzzy* dan *tren fuzzy* dari sampel historis. Kemudian, klasifikasi AFS dieksploitasi untuk menghasilkan

interpretasi semantik setiap *tren fuzzy*. Model yang diusulkan tidak hanya memprediksi nilainya tetapi juga dapat menangkap tren yang ada dalam deret waktu, dan mendapatkan interpretasi semantiknya.

Penelitian (Liu, Wang, & Pedrycz, 2015) mengusulkan sebuah pendekatan baru untuk pengelompokkan data. Tujuan dari pengelompokan ini adalah untuk mematuhi beberapa prinsip pengelompokan yang dilakukan oleh manusia saat menentukan struktur data. Dibandingkan dengan pendekatan *clustering* lainnya, pendekatan yang diusulkan menawarkan wawasan lebih rinci mengenai struktur cluster dan proses pengambilan keputusan yang mendasarinya. Ini berkontribusi terhadap peningkatan hasil interpretasi melalui kemampuan representasi teori AFS. Studi eksperimental telah menunjukkan bahwa metode *clustering fuzzy* yang diusulkan dapat menemukan *cluster* dalam data dan membantu menentukannya dalam beberapa aturan *fuzzy* yang komprehensif.

Penelitian (Indrawati & Girsang, 2016) ini mengusulkan kombinasi pendekatan inferensi sistem inferensi *neuro fuzzy* adaptif (ANFIS) dan optimasi partikel partikel (PSO) untuk memperkirakan kebutuhan listrik. PSO digunakan untuk mengoptimalkan fungsi keanggotaan ANFIS dengan mengubah parameter premis dan parameter konsekuen pada ANFIS.

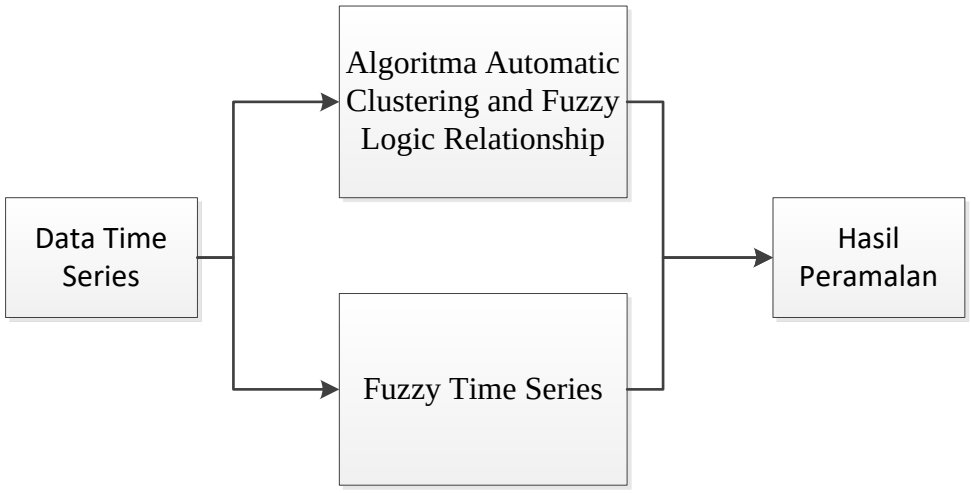
Penelitian (Sitohang, Girsang, & Suharjito, 2017) data yang digunakan adalah deret waktu jumlah penumpang selama sebelas tahun. Data ini akan membentuk pola yang mengindikasikan jumlah penumpang setiap bulan dalam setahun sebagai input data dan jumlah penumpang tahun depan sebagai target prediksi. Setelah data masukan dikelompokkan menjadi tiga jenis dengan

menggunakan *fuzzy* C-*means* (FCM), data diolah menggunakan sistem inferensi *neuro fuzzy* adaptif (ANFIS) untuk mendapatkan data prediksi.

Penelitian (Endaryati & Kurniawan, 2015) *Automatic clustering and fuzzy logic relationship* (ACFLR) digunakan untuk memprediksi data *time series* dengan tujuan membandingkan ACFLR dengan metode *Smoothing eksponensial* tunggal. Hasilnya ACFLR memiliki nilai MSE dan MAPE yang lebih rendah dari *Smoothing Exponensial*

## 2.5 Kerangka Pikir

Kerangka pikir penelitian analisis perbandingan ACFLR dan FTZ dalam memprediksi jumlah data penumpang pesawat udara tertera pada gambar 2.6 dibawah ini:




**Gambar 2.6** Kerangka Pikir

Berdasarkan kerangka pikir gambar 2.6 diatas, data *time series* jumlah penumpang pesawat akan terlebih dahulu diklusterisasi secara otomatis dengan menggunakan *algoritma automatic clustering*. Hasilnya akan didapatkan data

jumlah penumpang pesawat yang sudah terklusterisasi selanjutnya akan dipelajari dan dilatih, Demikian halnya dengan metode *fuzzy time series* sehingga akan tercipta suatu mesin *inference*. Dari mesin *inference* akan menghasilkan hasil peramalan. Hasil peramalan jumlah data penumpang pesawat udara dengan metode ACFLR dan FTZ akan dibandingkan besar errornya sehingga didapatlah kesimpulan metode mana yang terbaik dari kedua metode peramalan ini dalam meramalkan jumlah penumpang pesawat udara.