BAB II

KAJIAN PUSTAKA

A. Pariwisata

Pariwisata adalah suatu perjalanan yang dilakukan untuk sementara waktu, yang diselenggarakan dari suatu tempat ke tempat lain, dengan maksud bukan untuk mencari nafkah di tempat yang dikunjungi, tetapi semata-mata untuk menikmati perjalanan tersebut guna pertamasyaan dan rekreasi atau untuk memenuhi keinginan yang beraneka ragam.Menurut Morgenroth, pariwisata dalam arti sempit adalah lalu lintas orang-orang yang meninggalkan tempat kediamannya untuk sementara waktu, untuk berpesiar ke tempat lain, semata-mata sebagai konsumen dari buah hasil perekonomian dan kebudayaan, guna memenuhi kebutuhan hidup dan budayanya atau keinginan yang beraneka ragam dari pribadinya (Yoeti 1983: 107).

Secara umum wisatawan (tourist) adalah setiap orang yang bepergian dari tempat tinggalnya untuk berkunjung ke tempat lain dengan menikmati perjalanan dan kunjungannya itu. Menurut Norwal seorang wisatawan adalah seorang yang memasuki wilayah negeri asing dengan maksud tujuan apapun, asalkan bukan untuk tinggal permanen dan mengeluarkan uangnya di negeri yang dikunjungi, uang tersebut diperoleh bukan dari negeri yang dikunjungi, tetapi dari negeri lain (Yoeti 1983: 129).

Melihat sifat perjalanan dan ruang lingkup dimana perjalanan wisata itu dilakukan, maka kita dapat mengklasifikasikan wisatawan sebagai berikut (Yoeti 1983: 131):

1. Wisatawan asing *(foreign tourist)*

   Adalah orang asing yang melakukan perjalanan wisata, yang datang memasuki suatu negara lain yang bukan merupakan negara dimana ia biasanya tinggal. Dalam rangka meningkatkan tambahan penghasilan devisa negara, maka jenis wisatawan ini yang perlu ditingkatkan jumlahnya, karena uang yang dibelanjakannya merupakan devisa bagi negara yang menjadi *tourist receiving countries.*

2. *Domestic foreign tourist*

   Adalah orang asing yang berdiam atau bertempat tinggal pada suatu negara, yang melakukan perjalanan wisata di wilayah negara dimana ia tinggal. Orang tersebut bukan warga negara dimana ia berada, tetapi warga negara asing yang karena tugasnya atau kedudukannya menetap dan tinggal pada suatu negara, dengan memperoleh penghasilan dengan mata uang negara aslinya atau dengan mata uang negara dimana ia tinggal tetapi dalam jumlah yang berimbang.

1. *Domestic tourist*

   Adalah seorang warga negara suatu negara yang melakukan perjalanan wisata dalam batas wilayah negaranya sendiri tanpa melewati perbatasan negaranya.

2. *Indigenous foreign tourist*

Adalah warga negara suatu negara tertentu, yang karena tugasnya atau jabatannyaberada di luar negeri, pulang ke negara asalnya dan melakukan perjalanan wisata di wilayah negaranya sendiri.

3. *Transit tourist*

Adalah wisatawan yang sedang melakukan perjalanan wisata ke suatu negara tertentu, yang menumpang kapal udara atau kapal laut ataupun transportasi umum lain, yang terpaksa mampir atau singgah pada suatu bandara atau pelabuhan ataupun tempat transit transportasi umum lain bukan atas kemauannya sendiri.

4. *Business tourist*

Adalah orang yang melakukan perjalanan untuk tujuan lain bukan wisata, tetapi perjalanan wisata akan dilakukannya setelah tujuannya yang utama selesai.

B. Analisis *Time Series*

Dalam memprediksi nilai suatu variabel di waktu yang akan datang, perlu diperhatikan dan dipelajari terlebih dahulu sifat dan perkembangan variabel tersebut di waktu yang lalu. Nilai dari suatu variabel dapat diprediksi jika diketahui dahulu sifat dan variabel di waktu sekarang dan diwaktu yang lalu.Analisis data *time series* digunakan untuk melakukan analisis data yang mempertimbangkan pengaruh waktu. Data-data yang dikumpulkan secara periodik berdasarkan urutan waktu, bisa dalam jam,

hari, minggu, bulan, kuartal dan tahun, dapat dilakukan analisis menggunakan metode analisis data *time series*.

C. Autokorelasi

Autokorelasi adalah asosiasi atau ketergantungan bersama antara nilai-nilai suatu *time series* yang sama pada periode waktu yang berlainan dan digunakan untuk menentukan koefisien korelasi pada *time series*. Autokorelasi merupakan korelasi dari sebuah data *time series* untuk selang waktu (*lag*) yang berlainan. Autokorelasi dapat digunakan untuk menentukan ada tidaknya faktor musiman (*seasonality*) beserta panjang musim dalam deret tersebut (Makridakis et al, 1999: 512). Selain itu, autokorelasi dapat digunakan untuk menentukan kestasioneran suatu data.

Dalam suatu proses stasioner Y_t, rata-rata $E(Y_t) = \mu$ dan$var(Y_t) = E(Y_t - \mu)^2 = \sigma^2$adalahkonstan, dankovarians$Cov(Y_t, Y_{t+k})$antara$Y_t$dannilainyapada periodewaktu lain Y_{t+k} disebutautokovarianpada*lag*k, didefinisikan sebagai(Wei, 2006: 10):

$$\gamma_k = Cov(Y_t, Y_{t+k}) = E(Y_t - \mu)(Y_{t+k} - \mu) \tag{2.1}$$

Nilai-nilai γ_k pada saat k = 1,2,…. disebut fungsi autokovarian. Koefisien autokorelasi pada *lag*k (ρ_k) antara pengamatan Y_t dan Y_{t+k} pada populasi dinyatakan dalam bentuk (Montgomery et al, 2008:30):

$$\rho_k = \frac{E[(Y_t-\mu)((Y_{t+k}-\mu)]}{\sqrt{E[(Y_t-\mu)^2]E[(Y_{t+k}-\mu)^2]}} = \frac{Cov(Y_t,Y_{t+k})}{var(Y_t)} = \frac{\gamma_k}{\gamma_0} \tag{2.2}$$

dimana $var(Y_t) = var(Y_{t+k}) = \gamma_0$

dengan

γ_k : fungsi kovarians pada lag-k
ρ_k : fungsi autokorelasi pada lag-k
t : waktu pengamatan, t = 1,2,3,…
Y_t : pengamatan pada saat t
Y_{t+k} : pengamatan pada saat $t + k$

Nilai-nilai ρ_k pada saat k =1, 2,…n disebut fungsi autokorelasi (*Autocorrelation Function*/ ACF). Fungsi autokorelasi dapat diperkirakan dari fungsi autokorelasi sampel yang didefinisikan dengan (Montgomery et al, 2008:30)

$$r_k = \frac{c_k}{c_0} \tag{2.3}$$

dengan c_k adalah perkiraan fungsi autokovarian sampel yang didefinisikan sebagai

$$c_k = \frac{1}{n}\sum_{t=1}^{n-k}(Y_t - \bar{Y})((Y_{t+k} - \bar{Y}) \tag{2.4}$$

dengan

r_k : autokorelasi pada lag k
Y_t : pengamatan pada saat t
Y_{t+k} : pengamatan pada saat $t + k$
$\bar{Y}$: nilai rata–rata dari pengamatan

Nilai autokorelasi berkisar antar -1 sampai 1. Jika nilai autokorelasi tepat ± 1 atau mendekati, dapat disimpulkan terdapat hubungan yang tinggi antara data *time series* tersebut dalam *lag* yang berlainan. Jika nilai autokorelasi adalah 0, maka tidak terdapat hubungan dari data *time series* tersebut. Untuk mengetahui suatu autokorelasi signifikan atau tidak dapat menggunakan suatu pengujian dengan hipotesis sebagai berikut:

$H_0 : \rho_k = 0$ (koefisien autokorelasi *lagk* tidak signifikan)

$H_1 : \rho_k \neq 0$ (koefisien autokorelasi *lagk* signifikan)

Statistik uji yang digunakan adalah

$$t_{hitung} = \frac{r_k}{SE(r_k)} \tag{2.5}$$

dengan SE adalah standart *error* yang didefinisikan (Hanke & Wichern, 2005: 64)

$$SE(r_k) = \sqrt{\frac{1+2\sum_{i=1}^{k-1} r_i^2}{n}} \tag{2.6}$$

dengan

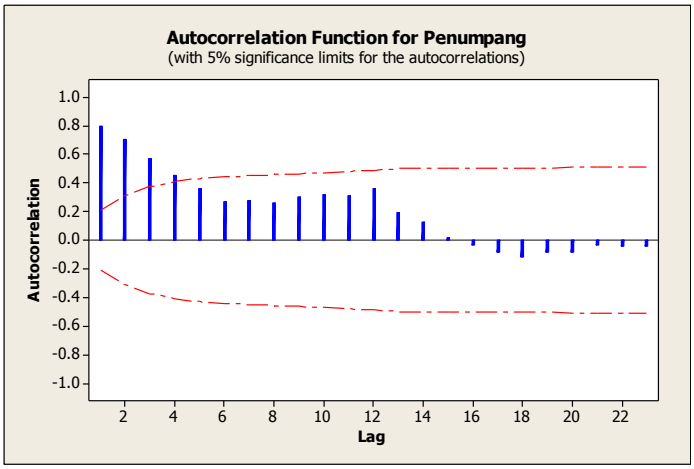
$SE(r_k)$: standar *error* koefisien autokorelasi pada *lag k*
r_k : koefisien autokorelasi pada *lag k*
n : banyak pengamatan.

Kriteria keputusan dari pengujian ini adalah autokorelasi signifikan jika $|t_{hitung}| > t_{\frac{\alpha}{2}}$ dengan derajat bebas *n*-1.

Signifikansi autokorelasi juga dapat dilihat dengan selang kepercayaan r_k dengan pusat 0. Selang kepercayaan r_k dapat dihitung dengan rumus:

$$0 \pm t_{n-1}(\alpha/2) \times SE(r_k) \tag{2.7}$$

Selang kepercayaan r_k dapat direpresentasikan dalam sebuah plot autokorelasi dengan bantuan program Minitab 16. Contoh plot autokorelasi dapat dilihat dari Gambar 2.5. Selang kepercayaan direpresentasikan dengan garis putus-putus merah. Kriteria autokorelasi pada suatu *lag* dikatakan signifikan jika nilai autokorelasi melewati garis putus-putus merah. Pada Gambar 2.1 autokorelasi*lag* yang signifikan adalah *lag* 1 sampai dengan *lag* 4.



Gambar 2.1 Contoh Plot Autokorelasi

D. Himpunan Klasik *(Crisp Set)*

Himpunan klasik adalah kumpulan objek yang tegas. Pada teori himpunan klasik, keberadaan suatu elemen dalam himpunan *A* hanya terdapat dua kemungkinan keanggotaan, yaitu menjadi anggota *A* atau bukan anggota *A* (Lin & Lee, 1996:12). Nilai keanggotaan atau derajat keanggotaan adalah nilai yang menunjukan besar tingkat keanggotaan elemen x dalam himpunan A dan dinotasikan dengan $\mu_A(x)$. Pada himpunan klasik nilai keanggotaannya menggunakan logika biner, yaitu 0 atau 1 untuk menyatakan keanggotaannya. Jika $\mu_A(x) = 1$, maka *x* merupakan anggota *A*. Jika $\mu_A(x) = 0$, maka *x* bukan anggota *A*.

$$\mu_A(x) = \begin{cases} 1, x \in A, untuk\ semua\ nilai\ x \\ 0, x \notin \text{A}, untuk\ semua\ nilai\ x \end{cases} \quad (2.8)$$

Contoh 2.1Jika diketahui $S = \{1,2,3,4,5,6\}$ adalah himpunan semesta dan

$A = \{2,5,6\}$ maka dapat diketahui bahwa:

1) Nilai keanggotaan 1 pada himpunan *A*, $\mu_A\, 1 = 0$ karena $1 \notin A$.
2) Nilai keanggotaan 2 pada himpunan *A*, $\mu_A\, 2 = 1$ karena $1 \in A$.
3) Nilai keanggotaan 3 pada himpunan *A*, $\mu_A\, 3 = 0$ karena $1 \notin A$.
4) Nilai keanggotaan 4 pada himpunan *A*, $\mu_A\, 4 = 0$ karena $1 \notin A$.
5) Nilai keanggotaan 5 pada himpunan *A*, $\mu_A\, 5 = 1$ karena $1 \in A$.
6) Nilai keanggotaan 6 pada himpunan *A*, $\mu_A\, 6 = 1$ karena $1 \in A$.

Himpunan klasik sesuai untuk berbagai macam aplikasi dan telah terbukti sebagai alat yang penting dalam matematika dan sains komputer, tetapi himpunan klasik tidak dapat menggambarkan konsep pemikiran manusia yang cenderung abstrak dan tidak tepat (Jang, Sun & Mizutani 1997: 13). Berdasarkan konsep pemikiran tersebut, munculah konsep himpunan *fuzzy* yang menjadi dasar dari logika *fuzzy*.

E. Logika Fuzzy

Logika *fuzzy* pertama kali diperkenalkan pada tahun 1965 oleh Zadeh seorang Profesor di bidang ilmu komputer, Universitas California, Berkeley. Zadeh beranggapan bahwa logika benar salah tidak dapat mewakili setiap pemikiran manusia. Penggunaan logika *fuzzy* akhir-akhir ini sangat diminati diberbagai bidang karena logika *fuzzy* dapat mempresentasikan setiap keadaan atau mewakili pemikiran manusia. Perbedaan mendasar dari logika *crisp* dan logika *fuzzy* adalah keanggotan elemen dalam suatu himpunan, jika dalam logika *crisp* suatu elemen mempunyai dua pilihan yaitu terdapat dalam himpunan atau bernilai 1 dan tidak pada himpunan atau bernilai 0. Keanggotaan elemen pada logika *fuzzy* berada di selang [0,1] (Sri Kusumadewi, 2010 : 158).

Logika *Fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang *input* ke dalam suatu ruang *output*. Beberapa proses logika *fuzzy* seperti himpunan *fuzzy*, fungsi keanggotaan, operasi dasar dalam himpunan *fuzzy* dan penalaran dalam himpunan *fuzzy*. Logika *fuzzy* menjadi alternatif dari berbagai sistem yang ada dalam

pengambilan keputusan karena logika *fuzzy* mempunyai kelebihan sebagai berikut (Sri Kusumadewi, 2010 : 154):

1) Konsep logika *fuzzy* 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 mesistemkan fungsi-fungsi nonlinear 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.

1. Himpunan Fuzzy

Teori himpunan *fuzzy* merupakan perluasan dari teori himpunan klasik. Teori himpunan *fuzzy* diperkenalkan oleh Zadeh pada tahun 1965. Misalkan S adalah himpunan semesta dan $x \in S$. Suatu himpunan *fuzzy A* dalam S didefinisikan sebagai suatu fungsi keanggotaan $\mu_A\ (x)$, yang memetakan setiap objek di S menjadi suatu nilai real dalam interval $[0,1]$.

Menurut Zimmermann (1991:11-12) jika *X* adalah himpunan dari objek-objek yang dinotasikan oleh *x*, maka himpunan *fuzzy A* dalam *X* adalah suatu himpunan pasangan berurutan:

$$A = \{(x, \mu_A(x)) | x \in X\} \quad (2.9)$$

dengan $\mu_A(x)$adalah derajat keanggotaan *x* di *A* yang memetakkan *X* ke ruang keanggotaan *M* yang terletak pada interval [0, 1].

Pada himpunan tegas *(crips)*, derajat keanggotaan suatu elemen x dalam himpunan *A*, memiliki 2 kemungkinan, yaitu:

a. Satu (1), yang berarti bahwa suatu elemen menjadi anggota dalam suatu himpunan, atau
b. Nol (0), yang berarti bahwa suatu elemen tidak menjadi suatu anggota dalam suatu himpunan

Kemiripan antara keanggotaan *fuzzy* dengan probabilitas menimbulkan kerancuan. Keduanya 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 keseringan suatu hasil bernilai benar dalam jangka panjang. Misalnya, jika derajat keanggotaan suatu himpunan *fuzzy* A adalah 0,7, maka tidak perlu dipermasalahkan berapa seringnya nilai itu diulang secara individual untuk mengharapkan suatu hasil yang hampir pasti muda. Di lain pihak, nilai probabilitas 0,7 berarti 30% dari himpunan tersebut diharapkan tidak A.

Himpunan *fuzzy* memiliki 2 atribut,yaitu:

a. Linguisik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti: RENDAH, SEDANG, TINGGI.
b. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti: 40, 25, 50, dsb.

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

a. Variabel *fuzzy*

   Variabel *fuzzy* merupakan variabel yang akan dibahas dalam suatu sistem *fuzzy*.

   Contoh 2.1 Variabel *fuzzy* yang akan dibahas dalam model ini adalah kedatangan wisatawan mancanegara.

b. Himpunan *fuzzy*

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

   Contoh 2.2 Banyaknya himpunan *fuzzy* yang akan dibahas dalam model ini adalah 6, yaitu himpunan *fuzzy*yang diberi namaA1, A2, A_3, A_4, A_5 dan A_6.

c. Himpunan universal (semesta pembicaraan)

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

bilangan positif maupun negatif. Adakalanya nilai semesta pembicaraan ini tidak dibatasi batas atasnya.

Contoh 2.3 Semesta pembicaraan untuk variabel *Input*kedatangan wisatawan mancanegara[1254 , 10860].

d. Domain

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

Contoh 2.4 Domain untuk kedatangan wisatawan mancanegara sebagai berikut :

Tabel 2.1 Domain pada kedatangan wisatawan mancanegara

Himpunan *fuzzy*	Domain
A1	[-667,2 , 3735]
A2	[1254 , 5096]
A3	[3735 , 7018]
A4	[5096 , 8939]
A5	[7018 , 10860]
A6	[10860 , 12780]

Himpunan *fuzzy* juga dapat dituliskan sebagai berikut (Lin & Lee, 1996:13):

$A = \mu_1/x_1 + \mu_2/x_2 + \cdots + \frac{\mu_n}{x_n} = \sum_{i=1}^{n} \frac{\mu_i}{x_i}$untuk himpunan diskrit (2.10)

$A = \int_S \frac{\mu_A(x)}{x}$untuk himpunan kontinu (2.11)

Simbol " / " bukan merupakan operasi pembagian, begitu juga simbol "+" bukan merupakan operasi penjumlahan.

Contoh 2.5Misalkan W adalah himpunan berat badan dalam kg pada orang-orang yang mempunyai tinggi badan 170 cm. Anggota W adalah

$W = \{40, 50, 55,65,70,80\}$

Fungsi keanggotaan pada variabel berat badan diberikan sebagai berikut:

$$\mu_{kurus}(w) = \begin{cases} 1; & w \leq 40 \\ \frac{60-w}{20}; & 40 \leq w \leq 60 \\ 0; & w \geq 60 \end{cases}$$

$$\mu_{langsing}(w) = \begin{cases} 0; & w \leq 50 \; atau \; w \geq 70 \\ \frac{w-50}{10}; & 50 \leq w \leq 60 \\ \frac{70-w}{10}; & 60 \leq w \leq 70 \end{cases}$$

$$\mu_{gemuk}(w) = \begin{cases} 0; & w \leq 60 \\ \frac{x-60}{20}; & 60 \leq w \leq 80 \\ 1; & w \geq 80 \end{cases}$$

Berdasarkan fungsi keanggotaan tersebut diperoleh derajat keanggotaan variable berat badan pada Tabel 2.2.

Tabel 2.2 Derajat Keanggotaan pada Variabel Berat Badan

Berat Badan	Kurus $(\mu_{kurus}(w))$	Langsing $(\mu_{langsing}(w))$	Gemuk $(\mu_{gemuk}(w))$
40	1	0	0
50	0,5	0	0
55	0,25	0,5	0
65	0	0,5	0,25
70	0	0	0,5
80	0	0	1

Himpunan *fuzzy* untuk kurus (*K*) dapat dinotasikan dituliskan sebagai berikut:

Himpunan pasangan berurutan:

$$K = \{(40;1)\ ,\ (50;0{,}5)\ ,\ (55;0{,}25)\ ,\ (65;0)\ ,\ (70;0)\ ,\ (80;0)\}$$

Himpunan diskrit:

$$K = \frac{1}{40} + \frac{0{,}5}{0{,}5} + \frac{0{,}25}{55} + \frac{0}{65} + \frac{0}{70} + \frac{0}{80}$$

2. Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data ke dalam nilai keanggotaannya (Sri Kusumadewi & Sri Hartati,2010: 22). Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Nilai keanggotaan *fuzzy* (derajat keanggotaan) memiliki interval antara 0 sampai 1.

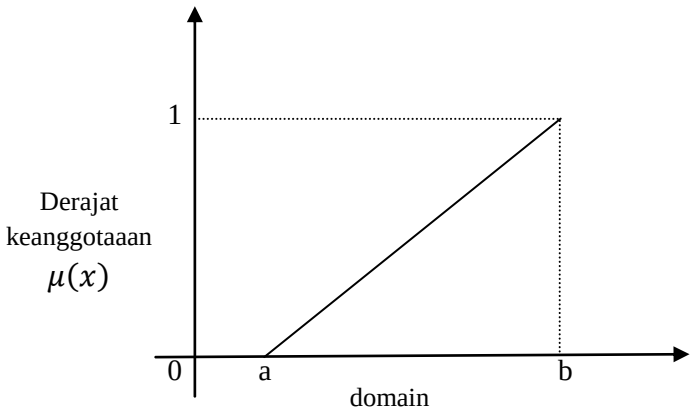
Fungsi keanggotaan yang dapat dibangun dan digunakan untukmempresentasikan himpunan *fuzzy* antara lain (Sri Kusumadewi, 2010):

a. Representasi Linear

Pada representasi linear, pemetaan *Input* ke derajat angggotanya digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas. Terdapat 2 keadaan pada himpunan *fuzzy* yang linear, yaitu:

1) Representasi linear naik

Representasi linear naik dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan yang lebih tinggi seperti pada gambar 2.2di bawah ini:



Gambar 2.2 Representasi Linear Naik

dengan fungsi keanggotaan kurva representasi linear naik:

$$\mu(x) = \begin{cases} 0; & x \leq a \\ \frac{(x-a)}{(b-a)}; & a < x \leq b \\ 1; & x \geq b \end{cases} \quad (2.12)$$

Contoh 2.6Salah satu himpunan *fuzzy* nilai kedatangan wismanadalah A2 dengan himpunan universal $U = [1254 \quad 10855]$ yang mempunyai fungsi keanggotaan:

$$\mu_{A2}(x) = \begin{cases} 0 & ; x \leq 1254 \\ \dfrac{x - 1254}{1921} & ; 1254 < x \leq 3175 \\ 1 & ; x > 3175 \end{cases}$$

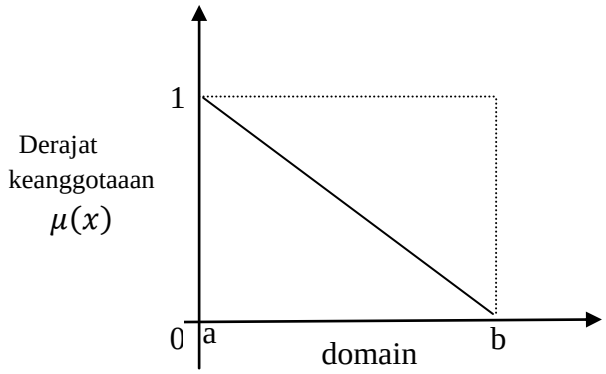
Berdasarkan fungsi keanggotaan tersebut, sebagai contoh menentukan derajat keanggotaan nilai kedatangan wismansebesar 2672 sehingga dapat dilakukan perhitungan:

$$\mu_{A2}(2672) = \frac{2672 - 1254}{1921} = 0{,}7381$$

Dapat diperoleh kesimpulan bahwa derajat keanggotaan nilai kedatangan wismansebesar 2672 adalah 0,7381 pada himpunan *fuzzy*A2. Sehingga nilai kedatangan wisman sebesar 2672merupakan anggota himpunan *fuzzy*A2 dengan nilai keanggotaan sebesar 0,7381.

2) Representasi linear turun

Representasi nilai 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 dengan derajat keanggotaan yang lebih rendah. Seperti pada gambar 2.3 di bawah ini:



Gambar 2.3 Representasi Linear Turun

dengan fungsi keanggotaan kurva representasi linear turun:

$$\mu(x) = \begin{cases} \frac{(b-x)}{(b-a)}\,; & a \le x \le b \\ 0\,; & x \ge b \end{cases} \qquad (2.13)$$

Contoh 2.7Salah satu himpunan *fuzzy* nilai kedatangan wismanadalah A2 dengan himpunan universal $U = [1254\,, 10855]$ yang mempunyai fungsi keanggotaan:

$$\mu_{A2}(x) = \begin{cases} 1 & ; x \leq 3175 \\ \dfrac{5096 - x}{1921} & ; 3175 < x \leq 5096 \\ 0 & ; x > 5096 \end{cases}$$

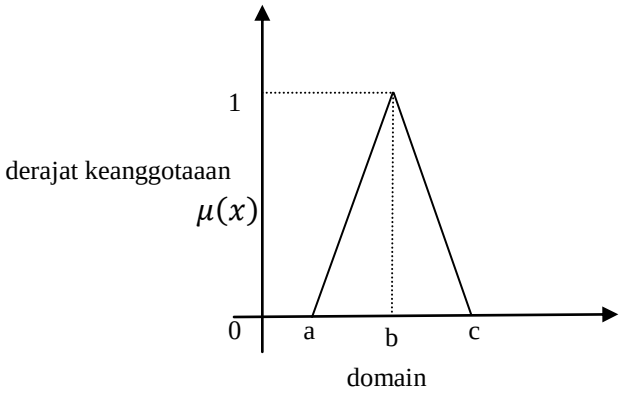
Berdasarkan fungsi keanggotaan tersebut, sebagai contoh menentukan derajat keanggotaan nilai kedatangan wisman sebesar 4253, sehingga dapat dilakukan perhitungan:

$$\mu_{A2}(4253) = \frac{5096 - 4253}{1921} = 0{,}4388$$

Dapat diperoleh kesimpulan bahwa derajat keanggotaan nilai kedatangan wismansebesar 4253 adalah 0,4388 pada himpunan *fuzzy*A2. Sehingga nilai kedatangan wisman sebesar 4253merupakan anggota himpunan *fuzzy*A2 dengan nilai keanggotaan sebesar 0,4388.

b. Representasi Kurva Segitiga

Representasi kurva segitiga pada dasarnya terbentuk dari gabungan 2 garis linear, yaitu linear naik dan linear turun. Kurva segitiga hanya memiliki satu nilai x dengan derajat keanggotaan tertinggi [1], hal tersebut terjadi ketika $x = b$. Nilai yang tersebar dipersekitaran b memiliki perubahan derajat keanggotaan menurun dengan menjauhi 1. Seperti pada gambar 2.4 di bawah ini:



Gambar 2.4 Kurva Segitiga

dengan fungsi keanggotaan Kurva Segitiga:

$$\mu(x) = \begin{cases} 0; & x \leq adanx > c \\ \frac{(x-a)}{(b-a)}; & a < x \leq b \\ \frac{(c-x)}{(c-b)}; & b < x \leq c \end{cases} \quad (2.14)$$

Contoh 2.7Salah satu himpunan *fuzzy* nilai kedatangan wismanadalah A2dengan himpunan universal $U = [1254\ 10855]$ yang mempunyai fungsi keanggotaan:

$$\mu_{A2}(x) = \begin{cases} 0 & x \leq 1254\ ataux \geq 5096 \\ \frac{x - 1254}{1921}, & 1254 \leq x \leq 3175 \\ \frac{5096 - x}{1921}, & 3175 \leq x \leq 5096 \end{cases}$$

Berdasarkan fungsi keanggotaan tersebut, sebagai contoh menentukan derajat keanggotaan untuk nilai kedatangan wisman sebesar 3488 sehingga dapat dilakukan perhitungan:

$$\mu_{A2}(3488) = \frac{5096 - 3488}{1921} = 0{,}8370$$

Dapat diperoleh kesimpulan bahwa derajat keanggotaan nilai kedatangan wismansebesar 3488 adalah 0,8370 pada himpunan *fuzzy* A2. Sehingga nilai kedatangan wisman sebesar 3488 merupakan anggota himpunan *fuzzy*A2 dengan nilai keanggotaan sebesar 0,8370.

3. Operator-Operator Fuzzy

Terdapat dua model operator *fuzzy*, yaitu operator dasar yang dikemukakan oleh Zadeh dan operator alternatif yang dikembangkan dengan konsep transformasi tertentu.

Beberapa operator dasar yang diciptakan oleh Zadeh (Sri Kusumadewi & Hartati, 2010 : 175) yaitu:

a. Operator-Operator Dasar Zadeh

Terdapat beberapa operasi yang didefinisikan secara khusus untuk mengkombinasi dan memodifikasi himpunan *fuzzy*. Nilai keanggotaan sebagai hasil dari operasi dua himpunan disebut *fire strength* atau α-predikat. Terdapat tiga operator dasar yang dikemukakan oleh Zadeh, yaitu:

1) Operator AND

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

2) Operator OR

Operator OR berhubungan dengan operasi union pada himpunan. α-predikat merupakan hasil dari operasi OR yang diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan yang bersangkutan.

3) Operator NOT

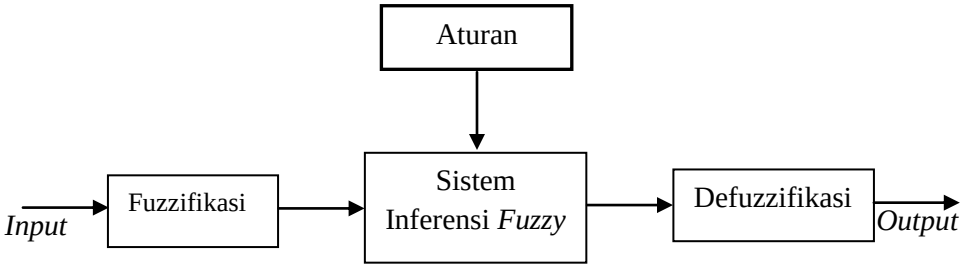
Operator NOT berhubungan dengan operasi komplemen himpunan. α-predikat merupakan hasil operasi dengan operator NOT yang diperoleh dengan mengurangkan nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1.

b. Operator Alternatif

Terdapat 2 tipe operator alternatif yaitu operator alternatif yang didasarkan pada transformasi aritmetika dan operator alternatif yang didasarkan pada transformasi fungsi yang lebih kompleks.

4. Susunan Sistem *Fuzzy*

Susunan sistem fuzzy dapat digambarkan pada gambar 2.5 berikut ini :



Gambar 2.5 Sistem Fuzzy (Li-Xin Wang, 1997 : 7)

Menurut Wang (1997 : 7) sistem *fuzzy* terdiri dari 3 tahapan, yaitu :

a. Fuzzifikasi

Fuzzifikasi merupakan tahap pertama dari perhitungan *fuzzy*, yaitu mengubah *input*yang bernilai *crisp* menjadi derajat keanggotaan yang bernilai *fuzzy*. Sehingga, tahap ini mengambil nilai-nilai *crisp* dan menentukan derajat dimana nilai-nilai tersebut menjadi anggota dari setiap himpunan *fuzzy* yang sesuai.

b. Inferensi

Inferensi adalah melakukan penalaran menggunakan *fuzzyinput* dan aturan *fuzzy* yang telah ditentukan sehingga menghasilkan *fuzzyoutput*. Secara sintaks, suatu aturan *fuzzy* dituliskan sebagai berikut :

IF anteseden THEN konsekuen

c. Defuzzifikasi

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 *crips* tertentu sebagai *output*.

5. Sistem Inferensi *Fuzzy*

Salah satu metode yang sering digunakan dalam inferensi adalah metode Mamdani. Metode Mamdani sering juga dikenal dengan nama Metode Max-Min. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Pada sistem ini untuk mendapatkan *output* diperlukan 4 tahap, antara lain (Sri Kusumadewi, 2003):

a. Pembentukan himpunan *fuzzy*

Pada metode Mamdani, variabel *input*dan variabel *output*dibagi menjadi satu atau lebih himpunan *fuzzy*.

b. Aplikasi fungsi implikasi

Pada metode ini, fungsi implikasi yang digunakan adalah Min.

c. Komposisi aturan

Ada 3 metode yang digunakan dalam melakukan inferensi sistem *fuzzy*, yaitu :

1) Metode max (maksimum)

Pada metode max,solusi himpunan *fuzzy* diperoleh dengan cara mengambil nilai maksimum aturan yang kemudian digunakan untuk memodifikasi daerah *fuzzy* dan mengaplikasikannya ke *output*dengan menggunakan operator *OR* (union/gabungan). Jika semua proposisi telah dievaluasi maka *output*akan berisi suatu himpunan *fuzzy* yang menggambarkan kontribusi dari tiap-tiap proposisi. Secara umum dapat dituliskan :

$$\mu_{sf}[x_i] \leftarrow \max(\mu_{sf}[x_i], \mu_{kf}[x_i]) \tag{2.15}$$

dengan
$\mu_{sf}[x_i]$: derajat keanggotaan solusi *fuzzy* sampai aturan ke-i.
$\mu_{kf}[x_i]$: derajat keanggotaan konsekuen *fuzzy* sampai aturan ke-i.

2) Metode *additive* (sum)

Pada metode ini, solusi himpunan *fuzzy* diperoleh dengan cara melakukan *bounded-sum* terhadap semua *output* daerah *fuzzy*. Secara umum dituliskan :

$$\mu_{sf}[x_i] \leftarrow \min(1, \mu_{sf}[x_i] + \mu_{kf}[x_i]) \tag{2.16}$$

dengan
$\mu_{sf}[x_i]$: derajat keanggotaan solusi *fuzzy* sampai aturan ke-i.
$\mu_{kf}[x_i]$: derajat keanggotaan konsekuen *fuzzy* sampai aturan ke-i.

3) Metode probabilistik OR (probor)

Pada metode ini, solusi himpunan *fuzzy* diperoleh dengan cara melakukan *product* terhadap semua *Output* daerah *fuzzy*. Secara umum dituliskan :

$$\mu_{sf}[x_i] \leftarrow (\mu_{sf}[x_i] + \mu_{kf}[x_i]) - (\mu_{sf}[x_i] * \mu_{kf}[x_i]) \quad (2.17)$$

dengan
$\mu_{sf}[x_i]$: derajat keanggotaan solusi *fuzzy* sampai aturan ke-i.
$\mu_{kf}[x_i]$: derajat keanggotaan konsekuen *fuzzy* sampai aturan ke-i.

d. Defuzzifikasi (Penegasan)

Terdapat beberapa metode defuzzifikasi pada komposisi aturan Mamdani:

1) Metode *Centroid*

Pada metode ini, solusi tegas diperoleh dengan cara mengambil titik pusat (z*) daerah *fuzzy*, secara umum dirumuskan:

$$z^* = \frac{\int_Z z\mu(z)dz}{\int_Z \mu(z)dz}; \text{ untuk semesta kontinu} \quad (2.18)$$

$$z^* = \frac{\sum_{j=1}^{n} z_j \mu(z_j)}{\sum_{j=1}^{n} \mu(z_j)}; \text{ untuk semesta diskret} \quad (2.19)$$

2) Metode Bisektor

Pada metode ini, solusi tegas diperoleh dengan cara mengambil nilai pada domain *fuzzy* yang memiliki nilai keanggotaan setengah dari jumlah total nilai keanggotaan pada daerah *fuzzy*. Secara umum dituliskan :

z_p sedemikian hingga $\int_{\Re 1}^{p} \mu(z)dz = \int_{p}^{\Re n} \mu(z)dz$

3) Metode *Mean of Maximum* (MOM)

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

4) Metode *Largest of Maximum* (LOM)

Pada metode ini, solusi tegas diperoleh dengan cara mengambil nilai terbesar dari domain yang memiliki nilai keanggotan maksimum.

5) Metode *Smallest of Maximum* (SOM)

Pada metode ini, solusi tegas diperoleh dengan cara mengambil nilai terkecil dari domain yang memiliki nilai keanggotan maksimum.

F. *Neural Network*

Model untuk menjelaskan hubungan non linear telah berkembang pesat hingga kini. Salah satu model tersebut adalah *Neural Network* (NN). Model NN adalah model yang didesain untuk memodelkan bentuk arsitektur syaraf pada otak manusia. Telah banyak dilakukan penelitian dengan menggunakan model NN. Hal ini karena didorong oleh adanya kemungkinan untuk menggunakan NN sebagai instrumen untuk menyelesaikan berbagai permasalahan aplikasi seperti *pattern recognition, signal processing, processing control* dan peramalan. NN terdiri dari elemen sederhana yang beroperasi secara paralel dan terinspirasi oleh sistem saraf biologis. Seperti di alam, fungsi jaringan ditentukan terutama oleh hubungan antar elemen, dalam NN elemen ini disebut neuron. Umumnya NN disesuaikan, atau dilatih, sehingga masukan tertentu mengarah ke target *output* tertentu. NN berawal

dari memodelkan otak manusia dengan cara berbeda dari *computer digital* konvensional.

Neuron terdiri dari 3 elemen pembentuk sebagai berikut:

1) Himpunan unit-unit yang dihubungkan dengan jalur koneksi.
2) Suatu unit penjumlah yang akan menjumlahkan masukan-masukan sinyal yang sudah dikalikan dengan bobotnya.
3) Fungsi aktivasi yang akan menentukan apakah sinyal dari *input* neuron akan diteruskan ke neuron lain ataukah tidak.

Proses pembelajaran (*learning)* NN dimulai dengan memasukkan informasi yang sebelumnya telah diketahui hasil kebenarannya. Pemasukan Informasi ini dilakukan melalui unit-unit *input*. Bobot-bobot antar koneksi dalam suatu arsitektur diberi nilai awal dan kemudian NN dijalankan. Bagi jaringan sendiri, bobot-bobot ini digunakan untuk belajar dan mengingat suatu informasi yang telah ada. Pengaturan bobot dilakukan secara terus menerus dan dengan menggunakan kriteria tertentu sampai diperoleh hasil yang sesuai dengan kriteria yang telah ditentukan.

Lapisan-lapisan penyusun NN dibagi menjadi 3 yaitu (Siang, 2005: 24):

1) Lapisan *input*

   Node-node di dalam lapisan *input* disebut neuron-neuron *input*. Neuron neuron *input* menerima *input* dari luar, *input* yang diberikan merupakan penggambaran suatu permasalahan.

2) Lapisan tersembunyi

Node di dalam lapisan tersembunyi disebut neuron tersembunyi. *Output* dari lapisan ini tidak dapat diamati secara langsung.

3) Lapisan *output*

Node-node di dalam lapisan *output* disebut neuron-neuron *output*. Keluaran dari lapisan ini merupakan hasil dari NN terhadap suatu permasalahan.

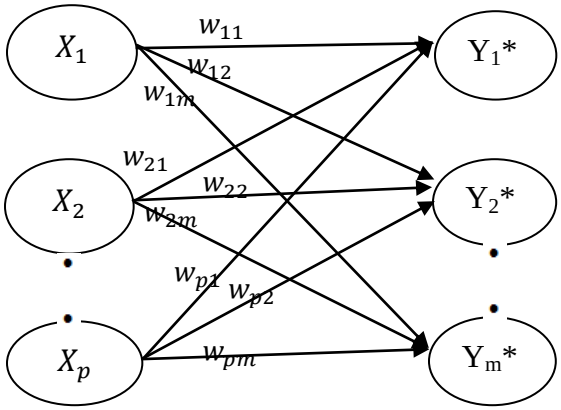
Penggunaan NN dapat diterapkan pada data *time series* dengan fungsi linear maupun non linear dengan berbagai tingkat akurasi yang diinginkan. Kelebihan NN lainnya adalah model NN dihasilkan langsung dari data. Beberapa contoh model NN adalah *perceptron, adaline, kohonen, hopfield, propagasibalik (feedforward), recurrent,* dan lain-lain (Diah Puspitaningrum, 2006: 22).

1. Arsitektur Jaringan

Model-model NN ditentukan oleh arsitektur jaringan serta algoritma pelatihan. Arsitektur biasanya menjelaskan arah perjalanan sinyal atau data di dalam jaringan, sedangkan algoritma belajar menjelaskan bagaimana bobot koneksi harus diubah agar pasangan *input-output* yang diinginkan dapat tercapai. Perubahan bobot koneksi dapat dilakukan dengan berbagai cara, tergantung pada jenis algoritma pelatihan yang digunakan. Dengan mengatur besarnya nilai bobot ini diharapkan bahwa kinerja jaringan dalam mempelajari berbagai macam pola yang dinyatakan oleh setiap pasangan *input-output* akan meningkat.

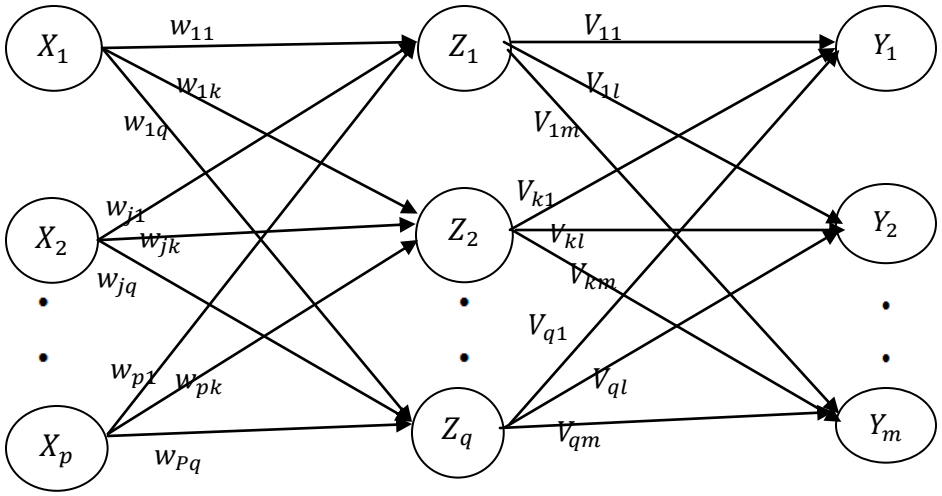
Beberapa arsitektur jaringan yang sering dipakai dalam NN antara lain (Fausett, 1994: 12-15):

a. Jaringan layar tunggal (*single-layer network*) adalah jaringan yang menghubungkan langsung neuron pada input layer dengan neuron pada *output layer*, meskipun dengan bobot yang berbeda-beda. Gambar 2.6 adalah contoh jaringan syaraf dengan lapisan tunggal.



Gambar 2.6 Arsitektur Jaringan Syaraf dengan Layer Tunggal

b. Jaringan layer jamak (*multi-layer network*) adalah jaringan yang lebih kompleks yang terdiri dari *input layer*, beberapa *hidden layer* dan *output layer*. Gambar 2.7 merupakan contoh arsitektur jaringan dengan banyak lapisan (*Multi-layer Net*).



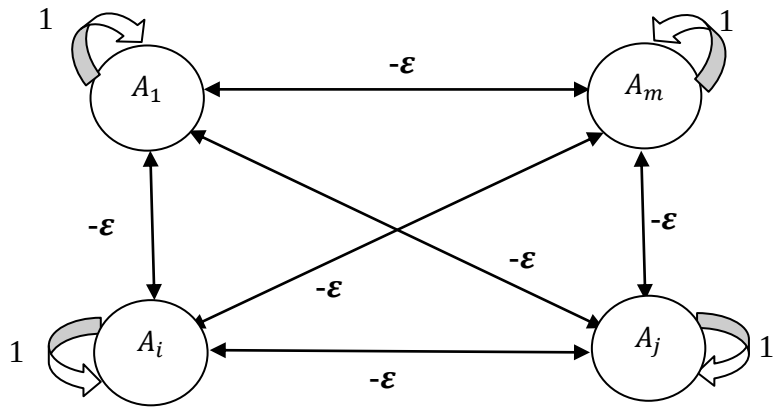
Gambar 2.7 Arsitektur Jaringan Syaraf dengan Layer Jamak

Pada gambar di atas, terdapat lapisan *input* dengan banyaknya *neuroninput*(x_j, j=1, 2, 3…, p). Lapisan tersembunyi ada satu dengan banyaknya *neuron* tersembunyi (z_k, k=1, 2,…, q). Dan lapisan *output* dengan banyaknya *neuron*(y_l, l =1, 2,…, m). Bobot-bobot yang menghubungkan *neuroninput* ke-*j* menuju *neuron* ke-*k* pada lapisan tersembunyi disimbolkan dengan w_{jk}, sedangkan v_{kl} adalah bobot-bobot dari *neuron* ke-*k* pada lapisan tersembunyi yang menuju ke-*l* pada lapisan *output*.

c. Jaringan Syaraf dengan Lapisan Kompetitif (*Competitive Layer Net*)

Bentuk arsitektur jaringan syaraf dengan lapisan kompetitif memiliki bentuk yang berbeda karena antar *neuron* pada jaringan ini dapat saling dihubungkan. Jaringan ini memetakan pola *input* secara tepat. Namun, apabila terlalu banyak membuat NN hanya mampu mengingat set data yang diberikan pada saat proses

pelatihan saja, sedangkan apabila diberikan data *input* yang baru maka jaringan tidak mampu mengeluarkan *output* yang benar atau *overfitting*. Gambar 2.8merupakan salah satu contoh arsitektur jaringan syaraf dengan lapisan kompetitif dengan koneksi dari lapisan tersebut memiliki bobot -ε.



Gambar 2.8 Arsitektur Jaringan Syaraf dengan Lapisan Kompetitif

Setiap jaringan memuat banyak jalur koneksi yang menghubungkan *neuron* dalam tiap *layer*. Penghubung ini memiliki bobot (*weight*) yang memfasilitasi pertukaran informasi antar *neuron*. Metode yang digunakan untuk menentukan bobot koneksi tersebut dinamakan algoritma pelatihan (*training*). Setiap *neuron* mepunyai tingkat aktivasi yang merupakan fungsi dari *input* yang masuk padanya. Aktivasi yang dikirim suatu *neuron* ke *neuron* yang lain berupa sinyal dan hanya dapat mengirim sekali dalam satu waktu, meskipun sinyal tersebut disebarkan pada beberpa *neuron* yang lain.

2. Algoritma Pembelajaran

Pembelajaran dalam NN didefinisikan sebagai suatu proses dimana parameter-parameter bebas NN diadaptasi melalui suatu proses perangsangan berkelanjutan oleh lingkungan dimana jaringan berada. Prosespembelajaran merupakan bagian penting dari konsep NN. Proses pembelajaran bertujuan untuk melakukan pengaturan terhadap bobot yang ada pada NN, sehingga diperoleh bobot akhir yang tepat sesuai dengan pola data yang dilatih (Sri Kusumadewi dan Sri Hartati, 2010:84). Pada proses pembelajaran akan terjadi perbaikan bobot-bobot berdasarkan alogaritma tertentu. Nilai bobot akan naik jika informasi yang diberikan ke suatu *neuron* mampu tersampaikan ke *neuron* yang lain. Sebaliknya, nilai bobot akan berkurang jika informasi yang diberikan ke suatu *neuron* tidak tersampaikan ke *neuron* lainnya. Terdapat 2 metode pembelajaran NN, yaitu (Fausett, 1994:15) :

a. Pembelajaran Terawasi (*Supervised Learning*)

Metode pembelajaran pada NNdisebut terawasi jika *output* yang diharapkan telah diketahui sebelumnya. Tujuan pembelajaran terawasi adalah untuk memprediksi satu atau lebih variabel target dari satu atau lebih variabel *input*. Pada proses pembelajaran, satu pola *input* akan diberikan ke suatu *neuron* pada lapisan *input*. Selanjutnya pola akan dirambatkan sepanjang NNhingga sampai ke *neuron* pada lapisan *output*. Lapisan *output* akan membangkitkan pola *output* yang akan dicocokan dengan pola *output* targetnya. *Error* muncul apabila terdapat perbedaan antara pola *output* hasil pembelajaran dengan pola target sehingga diperlukan pembelajaran lagi.

b. Pembelajaran Tak Terawasi (*Unsupervised Learning*)

Pembelajaran tak terawasi tidak memerlukan target *output* dan jaringan dapat melakukan *training* sendiri untuk mengekstrak fitur dari variabel independen. Pada metode ini, tidak dapat ditentukan hasil *output*nya. Selama proses pembelajaran, nilai bobot disusun dalam suatu *range* tertentu sesuai dengan nilai *input* yang diberikan. Tujuan pembelajaran ini adalah untuk mengelompokkan unit-unit yang hampir sama ke dalam suatu area tertentu.

3. Fungsi Aktivasi

Fungsi aktivasi merupakan fungsi yang mentransformasikan nilai penjumlahan menjadi sebuah nilai yang dapat diproses lebih lanjut. Fungsi ini sangat penting ketika melakukan tahap perhitungan *output*.

Pada algoritma *backpropagation* hanya dapat menggunaan fungsiaktivasi yang dapat didiferensialkan, antara lain sebagai berikut:

1) Fungsi Sigmoid Biner (*Logsig*)

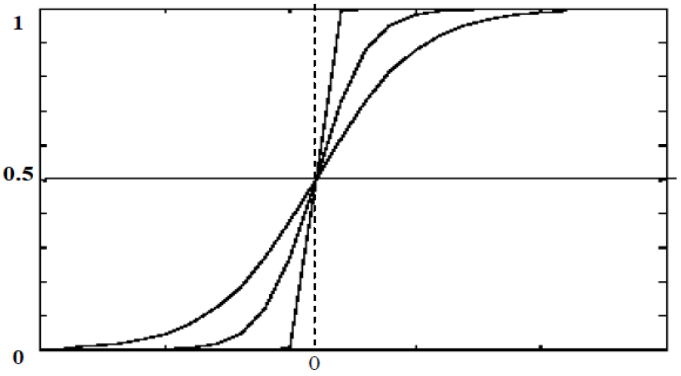
Fungsi sigmoid biner memiliki *range* 0 sampai dengan 1. Oleh karena itu, fungsi ini sering digunakan untuk jaringan yang membutuhkan nilai *output* pada interval 0 sampai 1. Namun, fungsi ini juga bisa digunakan oleh jaringan yang nilai *output*-nya 0 atau 1. Fungsi sigmoid biner sangat baik untuk menyelesaikan permasalahan kompleks dan bersifat *non-linier*. Rumus fungsi bipolar adalah sebagai berikut:

$$y = f(x) = \frac{1}{1+e^{-x}} \qquad -\infty < x < \infty \qquad (2.24)$$

dengan turunan fungsinya adalah:

$$f'(x) = \frac{e^{-x}}{1+2e^{-x}+e^{-2x}} \qquad -\infty < x < \infty \qquad (2.25)$$

Gambar 2.9 berikut merupakan grafik fungsi sigmoid biner (*logsig*):



Gambar 2.9 Fungsi Sigmoid Biner(*Logsig*)

2) Fungsi Linear (*Purelin*)

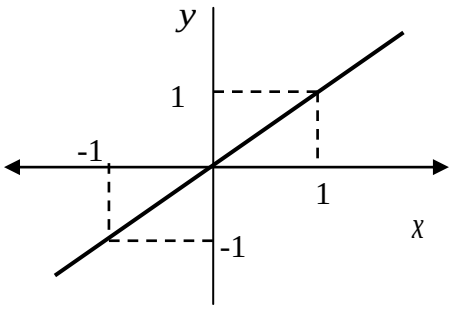
Fungsi linear mempunyai nilai *output* yang sama dengan nilai *input*-nya, sehingga disebut juga fungsi identitas. Rumus fungsi linear adalah sebagai berikut:

$$y = f(x) = x \qquad -\infty < x < \infty \qquad (2.26)$$

dengan turunan fungsinya adalah:

$$f'(x) = 1 \qquad (2.27)$$

Gambar 2.10berikut merupakan grafik fungsi linear (identitas):



Gambar 2.10 Fungsi Linear (Identitas)

3) Fungsi Sigmoid Bipolar (*Tansig*)

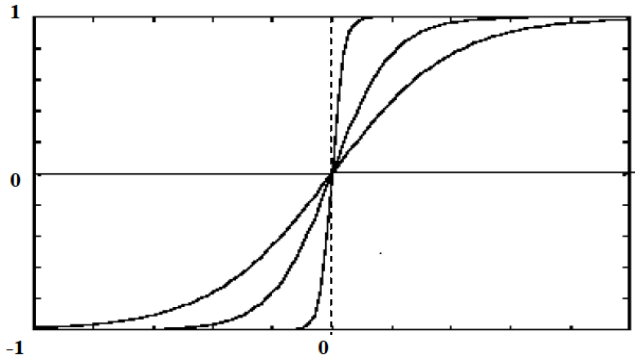
Fungsi sigmoid bipolar hampir sama dengan fungsi sigmoid biner, hanya saja *output* ini memiliki *range* antara 1 sampai -1. Rumus fungsi sigmoid bipolar adalah sebagai berikut:

$$y = f(x) = \frac{1-e^{-2x}}{1+e^{-2x}} \quad -\infty < x < \infty \tag{2.28}$$

dengan turunan fungsinya adalah:

$$f'(x) = \frac{2e^{-x}}{1+2e^{-x}+e^{-2x}} \quad -\infty < x < \infty \tag{2.29}$$

Gambar 2.11berikut merupakan grafik fungsi sigmoid bipolar:



Gambar 2.11 Fungsi Sigmoid Bipolar *(Tansig)*

4. Algoritma *Backpropagation Neural Network*

Jaringan syaraf tiruan dengan algoritma pembelajaran *backpropagation* menggunakan tiga tahap yaitu *feedforward* atau perambatan maju, *backpropagation* atau propagasi balik dari kumpulan kesalahan, dan perubahan atau penyesuaian bobot. Pada saat *feedforward* atau perambatan maju, *input* dihitung maju mulai dari lapisan *input* sampai lapisan *output* menggunakan fungsi aktivasi yang sudah ditentukan. Selanjutnya tahap *backpropagation* atau propagasi balik dari kumpulan kesalahan, pada tahap ini kesalahan yang terjadi dihitung menggunakan selisih antara nilai *output* jaringan dengan target yang diinginkan. Kesalahan tersebut kemudian dipropagasikan balik untuk mendapatkan *error* yang minimal, dimulai dari lapisan *output* sampai ke lapisan *input*. Tahap ketiga adalah perubahan atau penyesuaian bobot dan bias. Tahap ini dilakukan untuk menurunkan bobot yang terjadi. Langkah-langkah algoritma *backpropagation*adalah sebagai berikut (Fausett, 1994: 294-296):

Langkah 0 : Inisiasi bobot-bobot (ambil bobot awal menggunakan nilai *random* yang cukup kecil).

Langkah 1 : Menetapkan parameter pembelajaran seperti maksimum *epoch*, target *error*, dan *learning rate*

Langkah 2 : Kerjakan langkah-langkah berikut selama (*Epoch*< Maksimum *Epoch*) dan (MSE < Target *Error*).

Fase I : ***Feedforward***

Langkah 3 : Setiap *neuron input* (x_j, j = 1, 2, 3… , p) menerima sinyal *input*x_j dan meneruskan sinyal tersebut ke semua *neuron* yang ada di lapisan atasnya (lapisan tersembunyi).

Langkah 4 : Setiap *neuron* pada lapisan tersembunyi (z_k, k= 1, 2, …, q) menjumlahkan sinyal-sinyal *input* terbobot

$$z\_in_k = b_k + \sum_{j=1}^{p} x_j w_{jk} \quad (2.30)$$

Menggunakan fungsi aktivasi untuk menghitung sinyal *output*nya,

$$z_k = f(z\_in_k), \quad (2.31)$$

Mengirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit *output).*

Langkah 5 : Setiap unit *output* (*l*) menjumlahkan sinyal-sinyal *input* terbobot.

$$l\_in = b_0 + \sum_{k=1}^{q} z_k v_k \quad (2.32)$$

Menggunakan fungsi aktivasi untuk menghitung sinyal *output*-nya,

$$l = f\ (l\_in) \quad (2.33)$$

Mengirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit *output).*

Fase II: ***Backpropagation***

Langkah 6 : Setiap unit *output* (*yl*) menerima target pola yang berhubungan dengan pola *input* pembelajaran, hitung informasi *error*-nya:

$$\delta_0 = (t - y)\ f'(l\_in) \quad (2.34)$$

δ_0 adalah unit *error* yang akan dipakai dalam perubahan bobot lapisan di bawahnya. Hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki bobot v_k) dengan laju pembelajaran α.

$$\Delta v_k = \alpha . \delta_0 . z_k, \qquad (k = 1, 2, \ldots, q). \tag{2.35}$$

Hitung koreksi bias (yang nantinya akan digunakan untuk memperbaiki bobot b_0):

$$\Delta b_0 = \alpha . \delta_0 \tag{2.36}$$

Mengirimkan sinyal tersebut ke unit-unit pada lapisan sebelumnya.

Langkah 7 : Setiap unit tersembunyi (z_k , k= 1, 2, …, q) menjumlahkan hasil perubahan *input*-nya dari unit-unit di lapisan atasnya

$$\delta\_in_k = \sum_{k=1}^{q} \delta_0 v_k \tag{2.37}$$

Faktor δ unit tersembunyi:

$$\delta_k = \delta\_in_k . f'(z\_in_k) \tag{2.38}$$

Hitung koreksi bobot (yang nantinya akan dipakai untuk memperbaiki nilai w_{jk})

$$\Delta w_{jk} = \alpha \delta_k x_j, \ (k = 1, 2, \ldots, q; \ j = 1, 2, \ldots, p) \tag{2.39}$$

Hitung koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai b_k)

$$\Delta b_k = \alpha . \delta_k \tag{2.40}$$

Fase III : Perubahan bobot dan bias

Langkah 8 : Setiap unit *output* (l) memperbaiki bias dan bobot-bobotnya (k= 1, 2, …, q)

$$v_k\ (baru\ ) = v_k\ (lama) + \Delta v_k \tag{2.41}$$

Setiap unit tersembunyi (z_k, j = 1, 2, …, q) memperbaiki bias dan bobot-bobotnya (j= 1, 2, …, p)

$$w_{jk}\ (baru\ ) = w_{jk}\ \ (lama) + \Delta w_{jk} \tag{2.42}$$

Langkah 9 : Uji syarat berhenti

Setelah pelatihan jaringan selesai dijalankan, maka jaringan tersebut dapat digunakan sebagai pengenalan pola (menguji data *testing*). Namun, dalam hal ini aplikasi jaringan hanya terdiri dari fase *feedforward* yang termuat dalam persamaan (2.30) sampai dengan persamaan (2.33) saja yang digunakan untuk menentukan *output* jaringan. Sebelum melakukan pembelajaran, terlebih dahulu mengatur parameter-parameter yang akan digunakan dalam proses pembelajaran. Pemilihan parameter untuk jaringan syaraf tiruan *backpropagation* berkaitan dengan pemilihan inisialisasi bobot, kecepatan pembelajaran dan momentum.

a. Inisilisasi Bobot

Inisialisasi bobot awal sangat mempengaruhi jaringan syaraf dalam mencapai minimum global atau lokal terhadap nilai *error*, serta cepat tidaknya proses pelatihan menuju kekonvergenan. Apabila nilai bobot awal terlalu besar maka *input* ke setiap lapisan tersembunyi atau lapisan *output* akan jatuh pada daerah dimana turunan fungsi sigmoidnya akan sangat kecil. Sebaliknya apabila nilai bobot awal terlalu kecil, maka *input* ke setiap lapisan tersembunyi atau lapisan *output* akan sangat kecil, yang akan menyebabkan proses pelatihan akan berjalan sangat lambat. Biasanya bobot awal diinisialisasi secara *random* dengan nilai antara -0,5 sampai 0,5 atau -1 sampai dengan 1 atau interval yang lainnya (Sri Kusumadewi, 2004: 97). Perintah inisialisasi bobot BPNNdalam matlab ditulis dengan :

net=train(net,P,T)

b. Parameter Laju Pembelajaran

Parameter laju pembelajaran (*learning rate*) sangat berpengaruh pada proses pelatihan. Begitu pula terhadap efektivitas dan kecepatan mencapai konvergensi dari pelatihan. Nilai optimum dari *learning rate* tergantung permasalahan yang diselesaikan, prinsipnya dipilih sedemikian rupa sehingga tercapai konvergensi yang optimal dalam proses pelatihan. Nilai *learning rate* yang cukup kecil menjamin penurunan *gradient* terlaksana dengan baik, namun ini berakibat bertambahnya jumlah iterasi. Pada umumnya besarnya nilai laju pembelajaran tersebut dipilih mulai 0,001 sampai 1 selama proses pelatihan. Perintah *learning rate* dalam matlab ditulis dengan :

net.trainParam.lr=LearningRate

Nilai default untuk *learning rate* adalah 0.01.

c. Momentum

Disamping koefisien laju pembelajaran, ada koefisien lain yang bertujuan untuk mempercepat konvergensi dari algoritma *errorbackpropagation*. Penggunaan koefisien momentum ini disarankan apabila konvergensi berlangsung terlalu lama dan juga untuk mencegah terjadinya lokal minimum. Dengan penambahan momentum, bobot baru pada waktu ke-(t+1) didasarkan atas bobot pada waktu t dan t-1 (Purnomo & Kurniawan, 2006: 45-46). Besarnya momentum antara 0 sampai. Apabila nilai momentum = 0, maka perubahan bobot hanya akan dipengaruhi oleh gradiennya. Tetapi, apabila nilai momentum = 1, maka perubahan bobot akan sama dengan perubahan bobot sebelumnya. Perintah momentumdalam matlab ditulis dengan :

net.trainParam.mc=Momentum

Nilai default untuk momentum adalah 0.75.

d. Maksimum *epoch*

Maksimum *epoch* adalah jumlah *epoch* maksimum yang boleh dilakukan selama proses pelatihan. Iterasi akan dihentikan apabila nilai *epoch* melebihi maksimum *epoch.* Perintah maksimum *epoch* dalam matlab ditulis dengan :

net.trainParam.*epochs*=Max*Epoch*

Nilai default untuk maksimum *epoch* adalah 10.

e. Kinerja tujuan

Kinerja tujuan adalah target nilai fungsi kerja. Iterasi akan dihentikan apabila nilai fungsi kurang dari atau sama dengan kinerja tujuan. Perintah kinerja tujuandalam matlab ditulis dengan :

net.trainParam.goal=TargetError

Nilai default untuk kinerja tujuan adalah 0.

f. Rasio kenaikan *learning rate*

Rasio ini berguna sebagai faktor pengali untuk menaikkan*learning rate* yang ada terlalu rendah untuk mencapai kekonvergenan.Perintah kenaikan *learning rate* dalam matlab ditulis dengan :

net.trainParam.lr_inc=IncLearningRate

Nilai default untuk kenaikan *learning rate* adalah 1.05.

g. Rasio penurunan *learning rate*

Rasio ini berguna sebagai faktor pengali untuk menurunkan *learning rate* apabila *learning rate* yang ada terlalu tinggi untuk menuju ketidakstabilan. Perintah penurunan *learning rate* dalam matlab ditulis dengan :

net.trainParam.lr_dec=LearningRate

Nilai default untuk penurunan *learning rate* adalah 0.7.

h. Jumlah *epoch* yang akan ditunjukkan kemajuannya

Menunjukkan jumlah *epoch* berselang yang akan ditunjukkan kemajuannya.Perintah dalam matlab ditulis dengan :

net.trainParam.show=EpochShow

Nilai default untuk jumlah *epoch* yang akan ditunjukkan adalah 25.

i. Maksimum kenaikan kerja

Maksimum kenaikan kerja adalah nilai maksimum kenaikan *error* yang diijinkan, antara *error* saat ini dan *error* sebelumnya.Perintah dalam matlab ditulis dengan :

net.trainParam.max_perf_inc=MaxPerfInc

Nilai default untuk maksimum kenaikan kerja adalah 1.04.

5. Prosedur Pembentukan Model NN

ModelNNyang digunakan sangat mempengaruhinilai *output* yang dihasilkan. Oleh karena itu, perancangan strukturjaringan akan menjadi sifat utama dalam perancangan sistem. Prosedurpembentukan struktur jaringan untuk peramalan adalah sebagai berikut:

a. Pembagian data

Dalam pembagian data, data dibagi menjadi data *training* dandata *testing.* Aspek pembagian data harus ditekankan supayamemperoleh data *training* yang secukupnya yang selanjutnyadigunakan untuk proses pembelajaran dan data *testing* digunakanuntuk menguji proses pembelajaran yang dilakukan data *training*berdasarkan nilai MSE data *training* dan *testing*. Komposisi pembagian data untuk data *training* yang kurang sesuai akan menyebabkan jaringan tidak dapat mempelajari sebaran data dengan baik. Sebaliknya jika data yang digunakan untuk proses pembelajaran cukup banyak akan melambatkan proses pemusatan. Beberapa komposisi data *training* dan *testing* yang sering digunakan adalah 80% untuk data *training* dan 20% untuk data *testing*, 75% untuk data *training* dan 25% untuk data *testing*, atau 50% untuk data *training* dan 50% untuk data *testing*(Hota, Shrivas & Singhai, 2013: 165). Komposisi ini bebas dilakukan sesuai dengan data yang akan diolah.

b. Identifikasi input model

Identifikasi input model dilakukan dengan melihat plot fungsi *autokorelasi* (ACF) sebagai dasar penentuan *input*. Penentuan *input* dapat dilakukan dengan melihat *lag-lag* yang signifikan pada plot ACF.

c. Estimasi model

Estimasi model diperoleh melalui hasil pembelajran pada data *training* dengan membangun model terbaik. Model terbaik diperoleh dengan *trial* dan *error* terhadap beberapa macam arsitektur yang menghasilkan nilai *Mean Absolute Precentage Error*

(MAPE) dan*Mean Square Error* (MSE) terkecil. Estimasi model ini dilakukan dengan menentukan banyak neuron pada *hidden layer* dan membandingkan hasil pembelajaran terbaik dari data *training* dan data *testing.* Setelah terbentuknya arsitektur jaringan dari model terbaik, pada hasil pembelajaran akan diperoleh bobot-bobot yang digunakan sebagai parameter pada model jaringan yang terbangun, dan bobot-bobot tersebut digunakan untuk memprediksi nilai untuk periode selanjutnya. Nilai MAPE yaitu persentase nilai rata-rata *Absolute Error* dari kesalahan prediksi tanpa menghiraukan tanda positif atau negative dan MSE yaitu jumlah kuadrat selisih dari kesalahan prediksi dibagi dengan banyak pengamatan yang dirumuskan (Hanke & Wichern, 2005: 80):

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{y_t-\hat{y}_t}{y_t}\right| x\ 100\% \qquad (2.43)$$

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(y_t-\hat{y}_t)^2 \qquad (2.44)$$

dengan,

y_t : data sebenarnya
$\hat{y}_t$: hasil peramalan dihitung dari model yang digunakan pada waktu *t*
n : banyak pengamatan

d. Prediksi

Setelah proses pemilihan neuron terbaik, langkah selanjutnya adalah memprediksikan data pengamatan berdasarkan struktur jaringan yang telah terbangun. Dengan menggunakan data target dari data *training* dan data *testing.*