# BAB II

# LANDASAN TEORI

## 2.1 Pengertian Pemrograman Linear

Pemrograman Linear merupakan suatu model yang dapat digunakan dalam pemecahan masalah pengalokasian sumber-sumber yang terbatas secara optimal dengan menggunakan model matematika, istilah linear menunjukkan bahwa seluruh fungsi matematika yang ada di dalam model harus merupakan suatu fungsi linear, sedangkan *programming* atau pemrograman pada hakekatnya adalah sinonim dengan perencanaan .

Jadi Pemrograman Linear mencakup perencanaan kegiatan-kegiatan untuk mencapai suatu hasil yang optimal, yaitu suatu hasil yang mencerminkan tercapainya sasaran tertentu yang paling baik diantara alternatif-alternatif yang mungkin dengan menggunakan fungsi linear, biasanya Pemrograman Linear disingkat dengan PL, untuk memahami Pemrograman Linear dimulai dengan formulasi umum permasalahan Pemrograman Linear yang merupakan perumusan dari permasalahan Pemrograman Linear, formulasi umum tersebut terdiri dari fungsi tujuan yang akan dicari solusi optimalnya berdasarkan alternatif - alternatif yang tersedia yang dirumuskan dalam fungsi pembatas .

Model matematika Pemrograman Linear dapat ditulis dalam bentuk formulasi umum (2.1.1) sebagai berikut:

Fungsi tujuan : Optimalkan (1) $Z = \sum_{j=1}^{n} b_j x_j$

Pembatas :

$$(2) \sum_{j=1}^{n} A_{ij} x_j \leq c_i \text{ , atau}$$

$$(3) \sum_{j=1}^{n} A_{ij} x_j \geq c_i \text{ , atau} \qquad \ldots (2.1.1)$$

$$(4) \sum_{j=1}^{n} A_{ij} x_j = c_i$$

$$(5) x_j \geq 0$$

Untuk semua : $i = 1, 2, 3, \ldots, m$

$j = 1, 2, 3, \ldots, n$

Dari formulasi umum (2.1.1) Pemrograman Linear diatas, terdapat dua katagori permasalahan utama yaitu : (1) masalah maksimisasi dan (2) masalah minimisasi, dua katagori permasalahan utama tersebut masing – masing dijelaskan dalam formulasi umum (2.1.2) dan (2.1.3) sebagai berikut :

(1) . Masalah Maksimisasi

Fungsi tujuan : Maksimumkan (1) $Z = \sum_{j=1}^{n} b_j x_j$

Pembatas :

$$(2) \sum_{j=1}^{n} A_{ij} x_j \leq c_i$$

$$(3) x_j \geq 0 \qquad \ldots (2.1.2)$$

Untuk semua : $i = 1, 2, 3, \ldots, m$

$j = 1, 2, 3, \ldots, n$

(2) . Masalah Minimisasi

Fungsi tujuan : Minimumkan (1) $Z = \sum_{j=1}^{n} b_j x_j$

Pembatas : (2) $\sum_{j=1}^{n} A_{ij} x_j \ \geq \ c_i$

(3) $x_j \geq 0$ ... (2.1.3)

Untuk semua : $i = 1, 2, 3, \ldots, m$

$j = 1, 2, 3, \ldots, n$

Dibawah ini adalah 4 metode klasik (metode biasa) untuk menyelesaikan permasalahan Pemrograman Linear .

### 2.1.1 Metode Grafik

Metode grafik akan efektif jika digunakan untuk menyelesaikan masalah Pemrograman Linear yang hanya memiliki dua variabel keputusan, jika melebihi dua variabel keputusan akan sangat sulit untuk dicari solusi optimalnya.

### 2.1.2 Metode Simplex

Apabila suatu masalah Pemrograman Linear hanya mengandung dua variabel keputusan, maka dapat diselesaikan dengan metode grafik. Tetapi bila melibatkan lebih dari dua variabel keputusan, maka metode grafik tidak efektif untuk digunakan lagi , sehingga diperlukan metode simplex.

#### 2.1.2.1 Langkah – langkah tabel metode simplex

Tabel metode simplex biasanya hanya disebut dengan istilah metode simplex saja, dibawah ini diuraikan langkah – langkah metode simplex sebagai berikut :

***Langkah 1 : Merubah fungsi tujuan dan fungsi pembatas***

Fungsi tujuan dan fungsi pembatas dari formulasi umum (2.1.1) diubah menjadi fungsi implisit yang disebut formulasi umum standar metode simplex sebagai berikut :

1. Masalah Maksimisasi :

Fungsi Tujuan : Maksimumkan (1) $Z - \sum_{j=1}^{n} b_j x_j - 0S_i^* + MR_i^* = 0$

Pembatas :

$$\begin{aligned} &(2) \left(\sum_{j=1}^{n} A_{ij} x_j\right) + S_i = c_i \\ &(3) \left(\sum_{j=1}^{n} A_{ij} x_j\right) - S_i + R_i = c_i \qquad \ldots (2.1.2.1) \\ &(4) \left(\sum_{j=1}^{n} A_{ij} x_j\right) + R_i = c_i \\ &(5)\ x_j \geq 0 \end{aligned}$$

2. Masalah Minimisasi :

Fungsi Tujuan : Minimumkan (1) $Z - \sum_{j=1}^{n} b_j x_j - 0S_i^* - MR_i^* = 0$

Pembatas :

$$\begin{aligned} &(2) \left(\sum_{j=1}^{n} A_{ij} x_j\right) + S_i = c_i \\ &(3) \left(\sum_{j=1}^{n} A_{ij} x_j\right) - S_i + R_i = c_i \qquad \ldots (2.1.2.2) \\ &(4) \left(\sum_{j=1}^{n} A_{ij} x_j\right) + R_i = c_i \\ &(5)\ x_j \geq 0 \end{aligned}$$

Keterangan :

- S menyatakan variabel *slack* dan R menyatakan variabel *artificial*

-Indeks i pada $S_i^*$ dan $R_i^*$ menyatakan letak penempatan dari suatu pembatas

*Langkah 2 : Menyusun persamaan – persamaan ke dalam Tabel Simplex*

Tabel 2.1.2.1 Tabel Simplex dalam bentuk simbol

| VD | Z | $x_1$ | $x_2$ | ... | $x_n$ | $S_1$ | $S_2$ | ... | $S_m$ | $R_1$ | $R_2$ | ... | $R_m$ | NK |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Z | 1 | $-b_1$ | $-b_2$ | ... | $-b_n$ | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| $S_1$ | 0 | $A_{11}$ | $A_{12}$ | ... | $A_{1n}$ | ... | ... | ... | ... | ... | ... | ... | ... | $c_1$ |
| $S_2$ | 0 | $A_{21}$ | $A_{22}$ | ... | $A_{2n}$ | ... | ... | ... | ... | ... | ... | ... | ... | $c_2$ |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| $S_m$ | 0 | $A_{m1}$ | $A_{m2}$ | ... | $A_{mn}$ | ... | ... | ... | ... | ... | ... | ... | ... | $c_m$ |
| $R_1$ | 0 | $A_{11}$ | $A_{12}$ | ... | $A_{1n}$ | ... | ... | ... | ... | ... | ... | ... | ... | $c_1$ |
| $R_2$ | 0 | $A_{21}$ | $A_{22}$ | ... | $A_{2n}$ | ... | ... | ... | ... | ... | ... | ... | ... | $c_2$ |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| $R_m$ | 0 | $A_{m1}$ | $A_{m2}$ | ... | $A_{mn}$ | ... | ... | ... | ... | ... | ... | ... | ... | $c_m$ |

(Sumber : **Subagyo Pangestu, Drs., Asri Marwan, Drs. dan Handoko T. Hani, Dr.,** *Dasar dasar Operations Research*, Yogyakarta : Penerbit BPFE, 1992)

Keterangan : - **VD** menyatakan Variabel Dasar.

- **NK** menyatakan Nilai Kanan.

*Langkah 3 : Menyelesaikan Tabel Simplex*

Pada langkah ini dilakukan perbaikan – perbaikan untuk memperoleh kondisi optimal, perbaikan yang terjadi harus berhenti setelah nilai koefisien variabel pada baris $Z \geq 0$ untuk kasus maksimisasi dan nilai koefisien variabel pada baris $Z \leq 0$ untuk kasus minimisasi.

### 2.1.3 Metode Primal – Dual

Metode Primal – Dual di dalam PL sangat penting sekali untuk difahami, karena pada dasarnya setiap bentuk primal di dalam PL akan mempunyai bentuk dualnya, sehingga antara primal dan dual sangat berkaitan erat dalam pengambilan suatu keputusan, untuk lebih jelasnya akan diuraikan hubungan antara primal dan dual, sebagai berikut :

### 2.1.3.1 Formulasi Umum Primal Permasalahan Pemrograman Linear

Fungsi tujuan : Maksimumkan (1) $Z = \sum_{j=1}^{n} b_j x_j$

Pembatas : (2) $\sum_{j=1}^{n} A_{ij} x_j \leq c_i$ ... (2.1.3.1)

(3) $x_j \geq 0$

Untuk semua : $i = 1, 2, 3, \ldots, m$

$j = 1, 2, 3, \ldots, n$

### 2.1.3.2 Formulasi Umum Dual Permasalahan Pemrograman Linear

Fungsi tujuan : Minimumkan (1) $Z = \sum_{i=1}^{m} c_i y_i$

Pembatas : (2) $\sum_{i=1}^{m} A_{ij} y_i \geq b_j$ ... (2.1.3.2)

(3) $y_i \geq 0$

Untuk semua : $i = 1, 2, 3, \ldots, m$

$j = 1, 2, 3, \ldots, n$

### 2.1.3.3 Ketentuan – ketentuan pada Metode Primal – Dual :

1. Koefisien fungsi tujuan primal menjadi konstanta pada dual
2. Konstanta pada primal menjadi koefisien fungsi tujuan pada dual
3. Fungsi tujuan maksimal pada primal menjadi fungsi tujuan minimal pada dual
4. Setiap kolom pada primal berkorespondensi dengan baris pada dual
5. Setiap baris pada primal berkorespondensi dengan kolom pada dual
6. Tanda ketidaksamaan bergantung pada fungsi tujuan .

Pada pembahasan selanjutnya, contoh kasus pada permasalahan Pemrograman Linear dibedakan menjadi 2 katagori dalam penggunaan notasi :

1. Untuk kasus maksimisasi, notasi pada variabel keputusan menggunakan huruf x (sesuai dengan Formulasi Umum Primal).
2. Untuk kasus minimisasi, notasi pada variabel keputusan menggunakan huruf y (sesuai dengan Formulasi Umum Dual).

**Contoh 2.1.3.1** Kasus Primal Pemrograman Linear

Fungsi tujuan : Maksimumkan (1) $Z = 2x_1 + 4x_2 - 3x_3$

Pembatas : (2) $x_1 + 3x_2 - 2x_3 \leq 30$

(3) $x_1 + x_2 + x_3 \leq 24$

(4) $3x_1 + 5x_2 + 3x_3 \leq 60$

(5) $x_1 \geq 0$

(6) $x_2 \geq 0$

(7) $x_3 \geq 0$

Contoh 2.1.3.1 di atas merupakan kasus primal dari permasalahan Pemrograman Linear, sehingga bentuk dualnya dapat disusun sebagai berikut :

**Contoh 2.1.3.2** Kasus Dual Pemrograman Linear

Fungsi tujuan : Minimumkan (1) $Z = 30y_1 + 24y_2 + 60y_3$

Pembatas : (2) $y_1 + y_2 + 3y_3 \geq 2$

(3) $3y_1 + y_2 + 5y_3 \geq 4$

(4) $-2y_1 + y_2 + 3y_3 \geq -3 \longrightarrow$ (4) $2y_1 - y_2 - 3y_3 \leq 3$

(5) $y_1 \geq 0$

(6) $y_2 \geq 0$

(7) $y_3 \geq 0$

Implementasi langkah – langkah tabel metode simplex untuk menyelesaikan permasalahan pemrograman linear yang telah dibawa ke dalam bentuk dual (contoh 2.1.3.2), dijelaskan sebagai berikut :

***Langkah 1.***

Berdasarkan formulasi umum (2.1.2.2), maka contoh 2.1.3.2 dapat disusun dalam bentuk standar, sebagai berikut :

Fungsi tujuan : Minimumkan (1) $Z + (4M - 30)y_1 + (2M-24)y_2 + (8M-60)y_3 - MS_1 - MS_2 - 0S_3 = 6M$

Pembatas : (2) $y_1 + y_2 + 3y_3 - S_1 + R_1 = 2$

(3) $3y_1 + y_2 + 5y_3 - S_2 + R_2 = 4$

(4) $2y_1 - y_2 - 3y_3 + S_3 = 3$

(5) $y_1 \geq 0$

(6) $y_2 \geq 0$

(7) $y_3 \geq 0$

***Langkah 2.***

Tabel 2.1.3.1 Inputan Nilai ke dalam Tabel Simplex

| VD | Z | $y_1$ | $y_2$ | $y_3$ | $S_1$ | $S_2$ | $S_3$ | $R_1$ | $R_2$ | NK |
|---|---|---|---|---|---|---|---|---|---|---|
| Z | 1 | (4M-30) | (2M-24) | (8M-60) | -M | -M | 0 | 0 | 0 | 6M |
| $R_1$ | 0 | 1 | 1 | 3 | -1 | 0 | 0 | 1 | 0 | 2 |
| $R_2$ | 0 | 3 | 1 | 5 | 0 | -1 | 0 | 0 | 1 | 4 |
| $S_3$ | 0 | 2 | -1 | -3 | 0 | 0 | 1 | 0 | 0 | 3 |

***Langkah 3.***

Tabel 2.1.3.2 dibawah ini menunjukkan perubahan – perubahan yang dimulai dari keadaan tabel awal , tabel hasil perubahan pertama, tabel hasil perubahan kedua dan tabel hasil perubahan ketiga, yang menghasilkan nilai fungsi tujuan yang optimal, sebagai berikut :

Tabel 2.1.3.2 Keadaan optimal Dual

| VD | Z | $y_1$ | $y_2$ | $y_3$ | $S_1$ | $S_2$ | $S_3$ | $R_1$ | $R_2$ | NK |
|---|---|---|---|---|---|---|---|---|---|---|
| Z | 1 | (4M-30) | (2M-24) | (8M-60) | -M | -M | 0 | 0 | 0 | 6M |
| $R_1$ | 0 | 1 | 1 | 3 | -1 | 0 | 0 | 1 | 0 | 2 |
| $R_2$ | 0 | 3 | 1 | 5 | 0 | -1 | 0 | 0 | 1 | 4 |
| $S_3$ | 0 | 2 | -1 | -3 | 0 | 0 | 1 | 0 | 0 | 3 |
| Z | 1 | (4/3M-10) | (-2/3M-4) | 0 | (5/3M-20) | -M | 0 | (-8/3M+20) | 0 | (2/3M+40) |
| $y_3$ | 0 | 1/3 | 1/3 | 1 | -1/3 | 0 | 0 | 1/3 | 0 | 2/3 |
| $R_2$ | 0 | 4/3 | -2/3 | 0 | 5/3 | -1 | 0 | -5/3 | 1 | 2/3 |
| $S_3$ | 0 | 3 | 0 | 0 | -1 | 0 | 1 | 1 | 0 | 5 |
| Z | 1 | 6 | -12 | 0 | 0 | -12 | 0 | -M | (-M+12) | 48 |
| $y_3$ | 0 | 9/15 | 3/15 | 1 | 0 | -1/5 | 0 | 0 | 1/5 | 12/15 |
| $S_1$ | 0 | 4/5 | -2/5 | 0 | 1 | -3/5 | 0 | -1 | 3/5 | 2/5 |
| $S_3$ | 0 | 19/5 | -2/5 | 0 | 0 | -3/5 | 1 | 0 | 3/5 | 27/5 |
| Z | 1 | 0 | -9 | 0 | -30/4 | -15/2 | 0 | (-M+30/4) | (-M+15/2) | 45 |
| $y_3$ | 0 | 0 | 1/2 | 1 | -3/4 | 5/20 | 0 | 3/4 | -1/4 | 1/2 |
| $y_1$ | 0 | 1 | -2/4 | 0 | 5/4 | -3/4 | 0 | -5/4 | 3/4 | 1/2 |
| $S_3$ | 0 | 0 | 15/10 | 0 | -19/4 | 45/20 | 1 | 19/4 | -45/20 | 7/2 |

Dari Tabel 2.1.3.2 diperoleh solusi optimal sebagai berikut :

$[\, y_1, y_2, y_3, Z \,] = [\, 1/2, 0, 1/2, 45 \,]$.

### 2.1.4 Metode Dual Simplex

Apabila dalam permasalahan PL telah diperoleh kondisi optimal yaitu untuk kasus maksimisasi nilai koefisien variabel pada baris $Z \geq 0$ dan untuk kasus minimisasi nilai koefisien variabel pada baris $Z \leq 0$, tetapi belum fisibel (ada pembatas non negatif yang tidak terpenuhi), maka hal ini dapat diatasi dengan metode dual simplex dengan ketentuan :

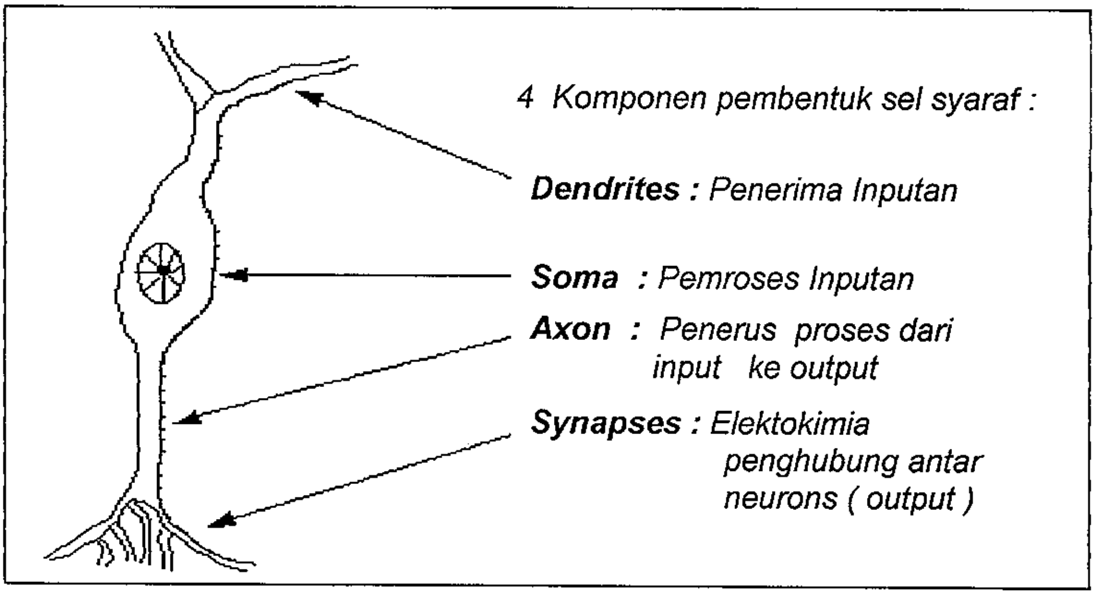
1. Semua pembatas merupakan ketidaksamaan $\leq$.

2. *Leaving Variabel (LV)* ditentukan dengan cara memilih harga *variabel baris* yang memiliki harga negatif terbesar , jika semua *variabel baris* tercapai harga positif atau nol maka kondisi ini adalah fisibel.
3. *Entering Variabel (EV)* ditentukan dengan mencari rasio Z dengan *Leaving Variabel* dipilih yang negatif dan abaikan yang positif atau nol , jika untuk setiap penyebut memiliki *Leaving Variabel* berharga $\geq 0$ maka persoalan tidak memiliki solusi fisibel.
4. Untuk kasus maksimisasi dipilih rasio absolut terkecil.

   Untuk kasus minimisasi dipilih rasio terkecil.

## 2.2 Jaringan Syaraf Biologis Manusia

Di bawah ini adalah gambaran umum sel syaraf biologis pada manusia :




(Sumber : http://www.Artificial Neural Networks.htm)

Gambar 2.2 Sel Syaraf Biologis Manusia

Berdasarkan Gambar 2.2, dapat difahami bahwa sel syaraf biologis manusia (dalam ilmu biologi, sel syaraf sering disebut dengan istilah *neuron*) terdiri dari 4 komponen utama yaitu :

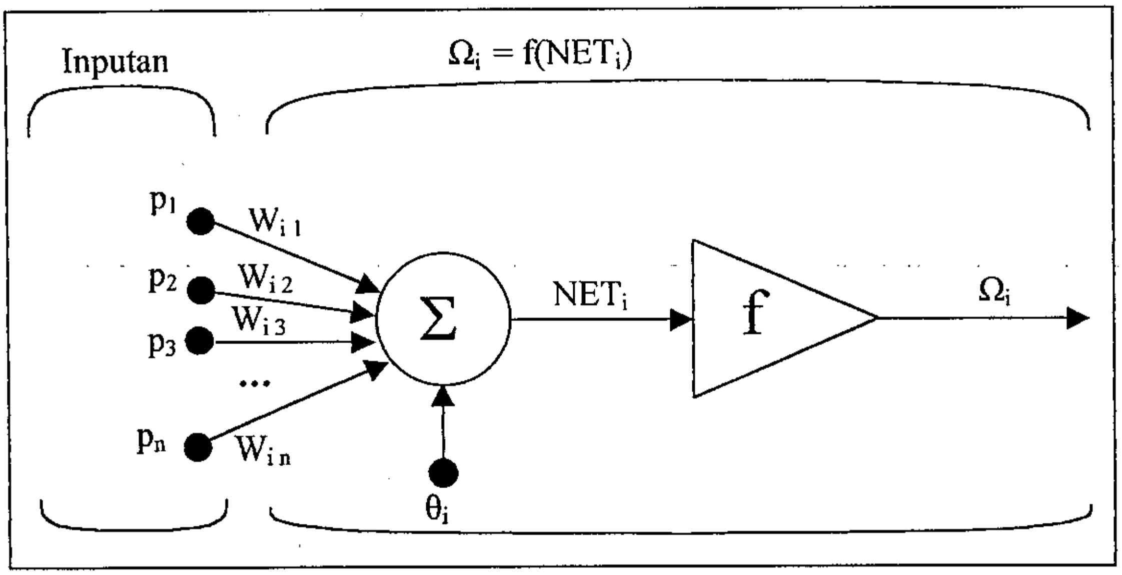
1. *Dendrites* , berfungsi untuk menerima inputan dari sel syaraf yang lain
2. *Soma* , berfungsi sebagai *Processor* ( Pemroses ) dari inputan
3. *Axon* , berfungsi untuk meneruskan proses dari input ke output atau dengan kata lain yaitu sebagai fungsi transfer .
4. *Synapses* , berfungsi untuk meneruskan output ke sel syaraf yang lain dan pada akhirnya , output dari *synapses* ini akan menjadi input bagi *dendrites* sel syaraf yang lain, dan membentuk suatu keterhubungan yang sangat kompleks, dengan demikian dapat difahami bahwa sekumpulan sel – sel syaraf pada manusia yang saling berhubungan dan sangat kompleks yang membentuk suatu jaringan disebut dengan jaringan syaraf biologis *(biological neural networks ).*

### 2.2.1 Jaringan Syaraf Tiruan

Jaringan syaraf tiruan *(artificial neural networks)* merupakan suatu sistem pemrosesan informasi yang diciptakan berdasarkan jaringan syaraf biologis manusia, sehingga jaringan syaraf tiruan memiliki karakteristik tampilan mirip dengan jaringan syaraf biologis manusia. Pada jaringan syaraf tiruan, sebuah sel syaraf akan terhubung dengan banyak sel syaraf yang lain sehingga akan membentuk suatu jaringan syaraf, yang dimaksud dengan jaringan syaraf atau *neural networks* pada jaringan syaraf tiruan yaitu elemen-elemen pembawa informasi yang terdiri dari sekumpulan sel – sel syaraf yang saling berhubungan antara sel syaraf yang satu dengan sel syaraf lainnya, sehingga jaringan syaraf ini membentuk suatu sistem yang menjalankan suatu fungsi tertentu.

Jaringan syaraf tiruan merupakan bidang penelitian dari *Artificial Intelligence* atau Kecerdasan Buatan yang terus diteliti dan dikembangkan oleh para ilmuan untuk membantu dalam menyelesaikan permasalahan yang ada dalam kehidupan manusia. Implementasi dari jaringan syaraf tiruan yaitu berupa suatu perangkat lunak yang dibangun oleh suatu bahasa pemrograman tertentu, dengan tujuan menjadikan komputer memiliki suatu kecerdasan tertentu yang dikehendaki oleh manusia.

Di bawah ini adalah gambaran umum sel syaraf tiruan sebagai berikut :




(Sumber : http://www.Artificial Neural Networks.htm)

Gambar 2.2.1 Sel Syaraf Tiruan

Berdasarkan Gambar 2.2.1, dapat difahami bahwa sel syaraf tiruan dari sel syaraf biologis manusia, terdiri dari 4 komponen utama yaitu :

1. $p_1$ , $p_2$ , $p_3$ , ..., $p_n$ merupakan inputan sel syaraf , inputan ini masing – masing merupakan pengali dengan suatu faktor penimbang (bobot) tertentu yang analog (serupa) dengan tegangan *synapses* pada sel syaraf biologis , faktor penimbang tersebut dilambangkan dengan $W_{i1}$ , $W_{i2}$ , $W_{i3}$ , ... , $W_{in}$ . Simbol $\theta_i$ melambangkan konstanta bias , yaitu konstanta yang bersama – sama merupakan komponen inputan dari sel syaraf .
2. Lambang $\sum$ memiliki arti bahwa , semua masukan tertimbang dan konstanta bias dijumlahkan secara bersama – sama dengan tujuan untuk menentukan tingkat aktivasi suatu sel syaraf , penjumlahan tersebut dinamakan dengan sinyal $NET_i$, yaitu $NET_i = \left(\sum_{j=1}^{n} W_{ij} p_j\right) + \theta_i$ ,Untuk $i = 1,2,...,m$
3. f merupakan lambang dari fungsi transfer ( biasa juga disebut dengan fungsi aktivasi ) , fungsi transfer ini berguna untuk memproses lebih lanjut sinyal $NET_i$ dengan tujuan untuk menghasilkan sinyal output sel syaraf .
4. $\Omega_i$ merupakan lambang dari sinyal output , pada Gambar 2.2.1 terlihat bahwa fungsi transfer f membawa $NET_i$ ke $\Omega_i$ .

### 2.2.2 Perbandingan antara Jaringan Syaraf Biologis Manusia dengan Jaringan Syaraf Tiruan

Tabel 2.2.2 Perbandingan antara Jaringan Syaraf Biologis Manusia dengan Jaringan Syaraf Tiruan

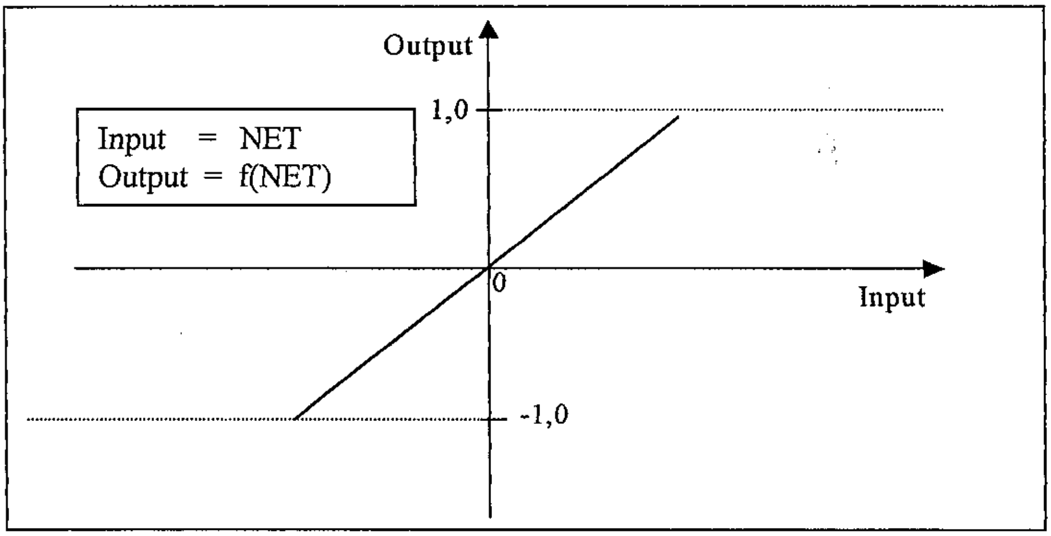
| Perbandingan / Kinerja | Jaringan Syaraf Biologis ( Komponen ) | Jaringan Syaraf Tiruan ( Simbol ) |
|---|---|---|
| 1. Inputan | Dendrites | $p_j$ , $W_{ij}$ , $\theta_i$ *dengan* $i = 1,2,\ldots,m$; $j = 1,2,\ldots,n$ |
| 2. Processor (pemroses) | Soma | $\sum$ |
| 3. Penerus proses | Axon | f |
| 4. Output | Synapses | $\Omega_i = f(NET_i)$ |

(Sumber : http://www.Artificial Neural Networks.htm)

### 2.2.3 Fungsi Transfer

Fungsi Transfer berguna untuk memproses lebih lanjut suatu sinyal yaitu sinyal NET$_i$ $=\left(\sum_{j=1}^{n} W_{ij}p_j\right)+\theta_i$ , Untuk $i=1,2,\ldots,m$ dengan tujuan untuk menghasilkan sinyal output sel syaraf yang disimbolkan dengan $\Omega_i$ , fungsi transfer yang digunakan dalam jaringan syaraf tiruan dinamis nonlinear Khanh V. Nguyen yaitu fungsi update linear.

Fungsi update berarti bahwa, nilai fungsi yang sekarang merupakan penjumlahan nilai fungsi sebelumnya dengan definisi fungsinya, pada jaringan syaraf tiruan Khanh V. Nguyen definisi fungsinya yaitu $f(\text{NET})=\alpha(\text{NET})$, dengan $\alpha$ = dt yaitu suatu tetapan selang waktu, artinya $f(\text{NET})=dt(\text{NET})$, sehingga fungsi transfer pada jaringan syaraf Khanh V. Nguyen didefinisikan sebagai $f(\text{NET})\text{baru}=f(\text{NET})\text{ lama}+dt(\text{NET})$, grafik dari suatu fungsi linear sederhana $f(\text{NET})=\text{NET}$ berbentuk garis lurus, sebagai berikut :




(Sumber : http://www.Artificial Neural Networks.htm)

Gambar 2.2.3. Fungsi Linear

### 2.3 Pengenalan Bahasa Pemrograman Matlab versi 5.3.1

Matlab merupakan bahasa pemrograman yang mengkapsulasi algoritma-algoritma matematis dalam bentuk yang dapat dengan mudah diterapkan pada berbagai bidang, misalnya dalam bidang pemrosesan sinyal digital, teori kontrol, aljabar linear, sinyal dan sistem, metode dan analisa numerik, matematika terapan, dan pada penerapan – penerapan lainnya, dalam Tugas Akhir ini digunakan Matlab versi 5.3.1 untuk mensimulasikan jaringan syaraf tiruan dinamis nonlinear Khanh V. Nguyen.

#### 2.3.1 Pengenalan Toolbox Matlab versi 5.3.1

Matlab disebut sebagai bahasa pemrograman karena Matlab memiliki fasilitas *M – File*, yaitu fasilitas untuk mengetikkan sintaks bahasa pemrograman Matlab, dan mengeksekusinya seperti bahasa tingkat tinggi lainnya, sedangkan Matlab dapat juga disebut sebagai bahasa pemrograman aplikasi karena Matlab dilengkapi dengan bermacam – macam *Toolbox*, yaitu kumpulan function untuk menyelesaikan permasalahan – permasalahan yang bersifat khusus yaitu permasalahan – permasalahan tertentu dalam bidang rekayasa maupun sains, Matlab dapat mensimulasikan suatu permasalahan dengan menggunakan antarmuka grafis ( *Graphical User Interface* ) sehingga solusi dari suatu permasalahan dapat dimengerti dengan jelas, contoh-contoh *Toolbox* yang sudah disediakan Matlab versi 5.3.1 adalah :

1. *Signal Processing Toolbox* : kumpulan function untuk mengolah data sinyal digital yang berhubungan dengan permasalahan sonar, radar, audio video, telekomunikasi, dan sebagainya.

2. *System Identification Toolbox* : kumpulan function untuk membangun model dari data pengamatan yang hasilnya berupa sistem dan sinyal.
3. *Optimization Toolbox* : kumpulan function untuk mengoptimalkan fungsi linear dan fungsi nonlinear.
4. *Simulink Toolbox* : kumpulan function yang berisi fasilitas untuk memudahkan dalam pembuatan block diagram.
5. *Fuzzy Logic Controller Toolbox* : kumpulan function yang berisi tentang kontrol otomatis, pengenalan pola, dan sebagainya yang berdasarkan pada konsep logika samar.
6. *Spline Toolbox* : kumpulan function yang berhubungan dengan permasalahan *polynomial.*

### 2.3.2 Membuat Tampilan Grafis *( Graphical User Interface )*

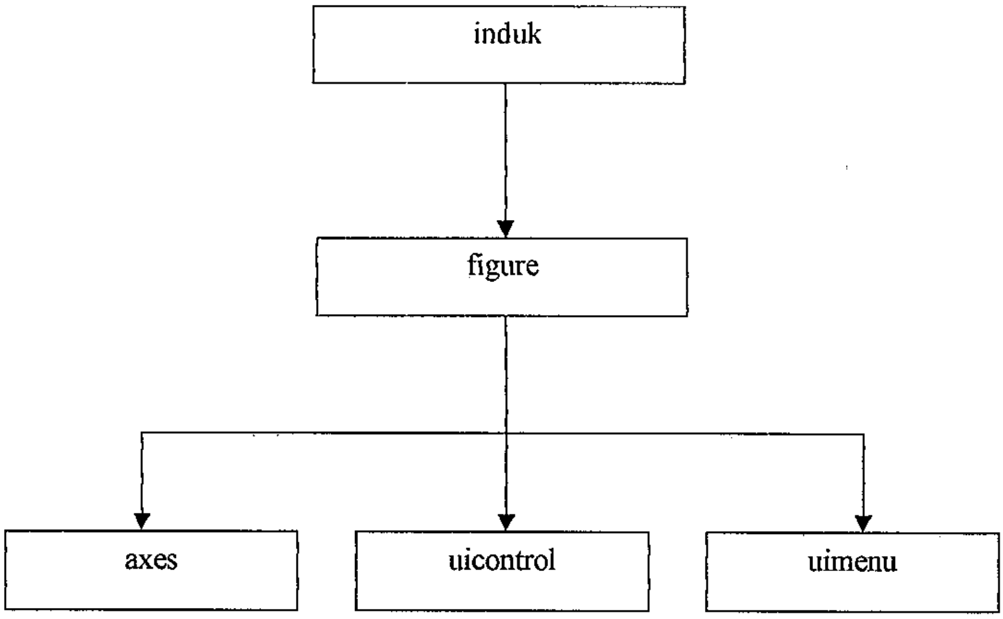
Semua komponen tampilan mempunyai nama *property* dan nilai atau sifat *property*. Untuk mengatur *property* tampilan digunakan perintah *set*, sedangkan untuk memperoleh *property* tampilan digunakan perintah *get*. Komponen sentral untuk tampilan grafis dalam program Matlab versi 5.3.1 adalah fungsi *figure*. Fungsi *figure* digunakan untuk membuat obyek gambar atau tampilan.

Fungsi *figure* mempunyai 3 anak *(children)*, yaitu :

1. fungsi *axes* : untuk membuat obyek sumbu
2. fungsi *uicontrol* : untuk membuat obyek kontrol

   Dalam Matlab versi 5.3.1 ada 9 gaya *(style)* untuk *uicontrol*, yaitu : *pushbutton,checkbox,listbox,slider,edit,popupmenu,text,radiobutton,frame*
3. fungsi *uimenu* : untuk membuat obyek *menu* (misalnya *file, help*).

Obyek *axes*, *uicontrol* dan *uimenu* menempel pada obyek *figure*, sehingga dapat dikatakan bahwa obyek *figure* adalah induk *(parent)* untuk obyek *axes*, *uicontrol* dan *uimenu*. Sedangkan obyek *figure* merupakan anak *(children)* dari obyek induk *(root object)*, yaitu obyek yang menangani grafik yang berhubungan dengan layar komputer, untuk lebih jelasnya dapat dilihat dalam struktur *graphical user interface* Matlab versi 5.3.1, sebagai berikut :




(Sumber : **Laboratorium Dasar Teknik Elektro Universitas Diponegoro,** *Modul Pelatihan Matlab 5.3.1*, 2001)

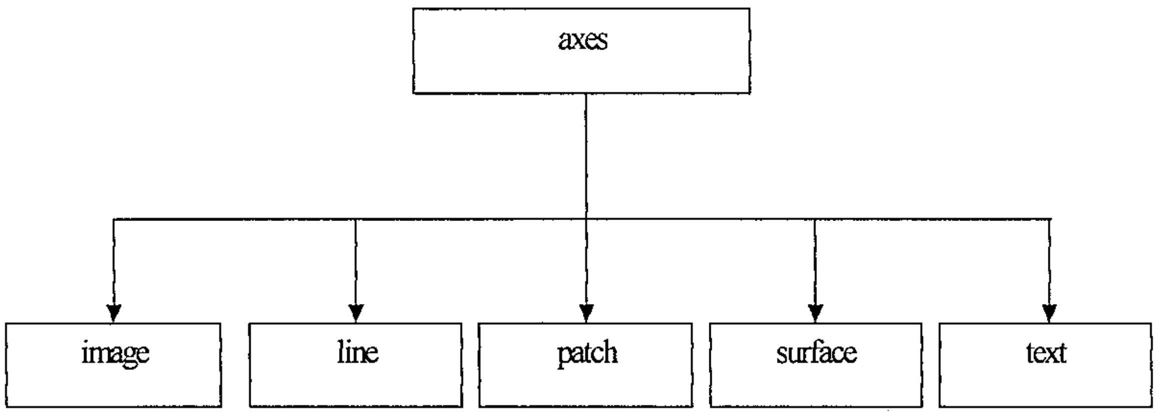
Gambar 2.3.2.1 Struktur *Graphical User Interface* Matlab versi 5.3.1

Obyek *axes* mempunyai 5 anak *(children)*, yaitu :

1. Fungsi *image* : untuk menampilkan gambar (fungsi level tinggi) atau untuk membuat obyek gambar (fungsi level rendah).
2. Fungsi *line* : merupakan fungsi level rendah untuk membuat obyek garis, fungsi level tinggi yaitu *plot* ( menghapus obyek *axes*).

3. Fungsi *patch* : merupakan fungsi grafis level rendah untuk membuat obyek 2 dimensi dan mewarnainya, fungsi level tinggi yaitu *fill* (menghapus obyek *axes*).
4. Fungsi *surface* : merupakan fungsi level rendah untuk membuat obyek gambar permukaan, fungsi level tinggi yaitu *mesh* atau *surf* ( menghapus obyek *axes* ).
5. Fungsi *text* : menambah teks pada *plot* ( fungsi level tinggi ) atau membuat obyek teks (fungsi level rendah).

Untuk lebih jelasnya, struktur obyek *axes* dalam Matlab versi 5.3.1, dapat digambarkan sebagai berikut :




(Sumber : **Laboratorium Dasar Teknik Elektro Universitas Diponegoro,** *Modul Pelatihan Matlab 5.3.1*, 2001)

Gambar 2.3.2.2 Struktur obyek *axes* Matlab versi 5.3.1