BAB II

TINJAUAN PUSTAKA

2.1 Peramalan

Seiring dengan senjang waktu antara kesadaran akan peristiwa atau kebutuhan mendatang dengan peristiwa itu sendiri, adanya waktu tenggang ini merupakan alasan utama bagi perencanaan dan peramalan. Jika waktu tenggang ini panjang dan akhir peristiwa tergantung pada faktor yang dapat diketahui, maka perencanaan dapat memegang peranan penting. Perencanaan merupakan kebutuhan yang besar, karena waaktu tenggang untuk mengambil keputusan dapat berkisar dibeberapa tahun (untuk kasus penanaman modal) sampai beberapa hari atau bahkan beberapa jam (untuk menjadwalkan produksi dan transportasi), peramalan merupakan alat bantu yang penting dalam perencanaan yang efektif dan efisien.

Beberapa jenis peramalan yang telah dikenal yaitu *causal* dan *time series.* Peramalan masih digunakan hingga saat ini.

Peramalan merupakan alat bantu yang penting dalam perencanaan yang efektif dan efisien. Peramalan dapat membantu untuk mengurangi ketidakpastian dalam melakukan perencanaan. Oleh karena itu peramalan memegang peranan penting dalam perencanaan dan pengambilan keputusan diberbagai bidang.

2.2 Pendekatan Peramalan

2.2.1 Fungsi Peramalan

Peramalan adalah menduga atau memperkiraan suatu keadaan dimasa yang akan datang berdasarkan keadaan masa lalu dan sekarang yang diperlukan untuk menetapkan kapan suatu peristiwa akan terjadi, sehingga tindakan yang tepat dapat dilakukan (Makridakis *et.al.,* 1999).

Fungsi peramalan adalah sebagai dasar bagi perencanaan kapasitas, anggaran, perencanaan penjualan, perencanaan produksi, dan inventori, perencanaan sumber daya, serta perencanaan pembelianbahan baku. Ada dua hal pokok yang harus diperhatikan dalam proses peramalan yang akurat dan bermanfaat:

1. Pengumpulan data yang relevan yang berupa informasi yang dapat menghasilkan peramalan yang akurat.
2. Pemilihan teknik peramalan yang tepat yang akan memanfaatkan informasi data yang diperoleh semaksimal mungkin.

Peramalan yang baik adalah peramalan yang dilakukan dengan langkah-langkah penyusunan yang teratur. Umumnya langkah-langkah tersebut dapat dibagi menjadi tiga bagian yaitu :

1. Menganalisa data masa lalu.
2. Menentukan metode yang akan digunakan.
3. Memproyeksikan data yang lalu dengan metode yang dipakai dan

mempertimbangkan adanya faktor-faktor perubahannya.

2.2.1.1 Pendekatan Kualitatif dan Kuantitatif

Pada dasarnya pendekatan peramalan dapat diklasifikasikan menjadi dua pendekatan, yaitu:

1. Pendekatan kualitatif

Metode peramalan kualitatif digunakan ketika data historis tidak tersedia. Metode peramalan kualitatif ini adalah metode subyektif. Hal ini meliputi metode pencatatan faktor-faktor yang dianggap akan mempengaruhi produksi terhadap hasil produksi tersebut, ataupun mengikuti pendapatan para pakar yang ahli terhadap produk yang hendak diprediksi. Dengan dasar informasi tersebut kita dapat memprediksi kejadian-kejadian dimasa yang akan datang. Yang termasuk pendekatan kualitatif antara lain *Market Research*, *Costumer Surveys*, *Delphi Method, Sales Force Composite, Excecutive Opinions, Historical Analogy dan Panel Consencuss.*

2. Pendekatan kuantitatif

Metode peramalan kuantitatif dapat dibagi menjadi dua tipe, *causal* dan *time series.* Metode peramalan *causal* meliputi faktor-faktor yang berhubungan dengan variable yang diprediksi. Sebaliknya peramalan *time series* merupakan metode kuantitatif untuk menentukan data masa lampau yang telah dikumpulkan secara teratur. Data lampau tersebut dapat dijadikan acuan untuk peramalan data dimasa yang akan datang (Makridakis *et.al.,* 1999).

2.2.1.2 Tinjauan Metode Kuantitatif

Terdapat dua macam metode kuantitatif yaitu model time series dan juga asosiatif (Heizer & Render, 2004).

Model time series adalah model prediksi dengan asumsi bahwa masa depan merupakan funsi masa lalu. Dengan kata lain, mereka melihat apa yang terjadi selama kurun waktu tertentu, dan menggunakan data masa lalu tersebut untuk melakukan peramalan. Jika kita memeperkirakan penjualan mingguan mesin pemotong rumput, kita menggunakan data masa lau untuk membuat ramalan.

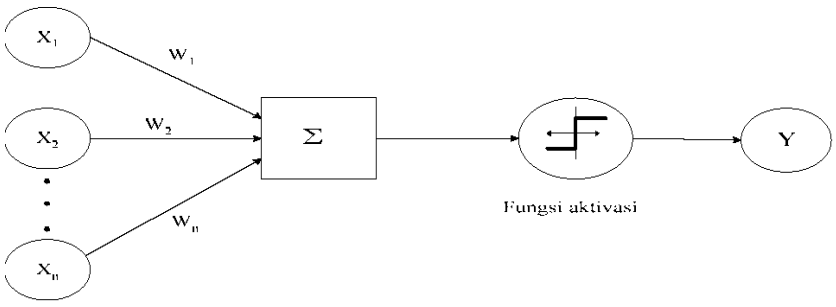
Model asosiatif (atau hubungan sebab akibat), seperti regresi linear, menggabungkan variable atau faktor yang mungkin mempengaruhi kuantitas yang sedang diramalkan. Sebagai contoh, model asosiatif dari penjualan mesin pemotong rumput mungkin memasukkan faktor seperti adanya perumahan baru, anggaran iklan, dan harga pesaing.

2.2.2 Jaringan Syaraf Tiruan

Jaringan syaraf tiruan merupakan sistem pengelola informasi yang memiliki karakter seperti jaringan biologis, yaitu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah buatan disini digunakan karena jaringan syaraf ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran Kusumadewi (2003).

Komponen dari jaringan syaraf tiruan terdiri atas:

1. Neuron : sel syaraf yang berfungsi membawa pesan informasi. Setiap neuron akan memiliki satu inti sel, inti sel ini nanti yang akan bertugas untuk melakukan pemrosesan informasi.
2. Bobot : bobot digunakan untuk mengatur jaringan dan menyelesaikan persoalan dengan menggunakan nilai tertentu berupa nilai matematis.
3. Aktivasi : fungsi yang menggambarkan hubungan antara tingkat aktivasi internal. Aktivasi merupakan fungsi dari *input* yang diterima. Seperti terlihat pada gambar 2.1.
4. Output : hasil pemahaman jaringan terhadap data *input* yang diberikan.



Gambar 2.1 Fungsi aktivasi pada Jaringan Syaraf Tiruan.

Keuntungan dari jaringan syaraf tiruan meliputi :

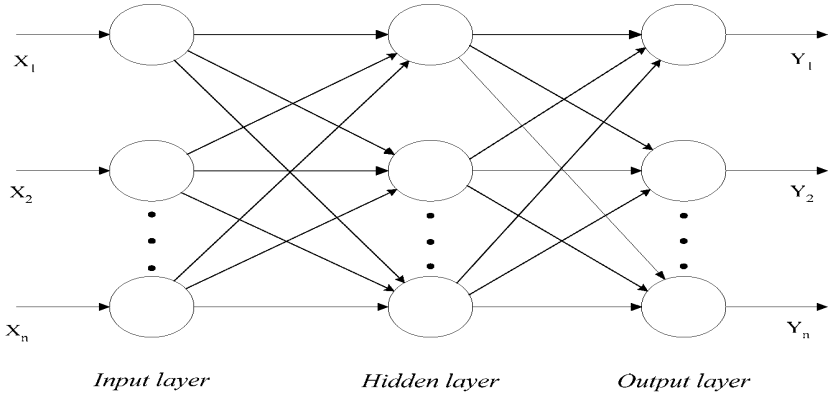
1. *Adaptive Learning*, yaitu kemampuan untuk belajar bagaimana melakukan tugas berdasarkan data yang telah diberikan untuk latihan atau pengalaman pertama.

2. *self-organization,* yaitu sebuah jaringan syaraf tiruan dapat membuat organisasi sendiri atau merepresentasi informasi yang diterimanya pada saat

proses pembelajaran.

3. *Real time operation,* yaitu: *input*, proses dan *output* (hasil) pada saat itu juga.
4. *Fault Tolerance via redundance Information Coding*, yaitu toleransi kesalahan melalui pengkodean informasi yang berlebihan.
5. Dapat mengartikan data yang rumit.
6. Dapat juga digunakan untuk mengubah pola yang terlalu kompleks.

Jaringan syaraf tiruan terdiri atas lapisan masukan (*layer input)* dan lapisan keluaran *(layer output)*. Tetapi ada juga yang mempunyai lapisan tersembunyi *(hidden layer)* di antara lapisan masukan dan keluaran. Unit yang ada pada lapisan masukan disebut dengan unit masukan. Pada unit masukan tidak memproses suatu informasi tetapi hanya menyebarkan atau menyalurkan ke unit lain. Sedangkan unit yang ada pada lapisan tersembunyi dan lapisan keluaran menghasilkan keluaran. Gambar 2.2 menunjukkan JST dengan 3 lapisan, terdiri atas lapisan masukan, satu lapisan tersembunyi dan lapisan keluaran.



Gambar 2.2 Jaringan Syaraf Tiruan dengan tiga lapisan

Pada gambar 2.2 lapisan jaringan syaraf tiruan terdiri dari tiga lapisan jaringan yaitu :

a. Unit masukan (*Input*: X_1, X_2, X_n) : *Nodes* yang terdapat dalam lapisan ini disebut sebagai unit *input* yang mengubah suatu *input* yang dimasukkan ke dalam bentuk sinyal yang dapat dimengerti sistem dan diteruskan ke dalam jaringan untuk diproses.
b. Unit tersembunyi (*Hidden* : Z_1, Z_2, Z_n) : *Nodes* yang berada pada lapisan ini disebut sebagai *hidden unit*, yaitu unit-unit yang tidak berhubungan secara langsung dengan dunia luar (misal : informasi masukan). Lapisan inilah yang membuat jaringan memiliki sifat non *linear* karena terjadi proses komputasi.
c. Unit keluaran (*Output* : Y_1, Y_2, Y_n) : Unit ini merupakan sebutan untuk *nodes* yang berada di dalam lapisan ini yang keluar dari proses sehingga dapat ditafsirkan sesuai dengan kasus yang dikehendaki.

2.2.2.1 Pelatihan Jaringan Syaraf Tiruan

Untuk dapat menyelesaikan suatu permasalahan, jaringan syaraf buatan melakukan algoritma belajar, yaitu bagaimana sebuah konfigurasi jaringan syaraf buatan dapat dilatih untuk mempelajari data historis yang ada. Dengan pelatihan ini, pengetahuan yang terdapat pada data dapat diserap dan direpresentasikan oleh harga – harga bobot koneksinya. Ada dua jenis algoritma belajar, yaitu Erwin (2005) :

1. *Supervised Learning* (proses belajar terawasi), pada pembelajaran ini target tersedia. Contohnya antara lain *Delta Rule*, algoritma backpropagation. Dalam

proses belajar yang terawasi, cara pelatihan jaringan tersebut adalah dengan memberikan data yang disebut data training atau *training vectors*. Training data terdiri atas pasangan *input-output* yang diharapkan dan merupakan *associativeMemory*. Data tersebut biasanya didapat dari pengalaman atau pengetahuan seseorang dalam menyelesaikan persoalan. Setelah jaringan dilatih, *associative memory* akan mengingat suatu pola. Jika jaringan diberi *input* baru, jaringan dapat mengeluarkan *output* seperti yang diharapkan (*desired* atau *target output*) berdasarkan pola yang sudah ada.

2. *Unsupervised Learning* (proses belajar tak terawasi). Algoritma ini sama sekali tidak menggunakan data target (tanpa target). Pada algoritma belajar ini, tidak membutuhkan target untuk keluarannya. Oleh karena itu tak ada perbandingan yang dilakukan dengan respon ideal yang ditetapkan sebelumnya. Rangkaian pelatihan hanya berisi vektor masukan saja.

2.2.2.2 Arsitektur Jaringan Syaraf Tiruan

Berdasarkan jumlah layer jaringan dapat dibedakan *single layer* dan *multilayer*. Jaringan yang berbentuk *single layer*, yaitu jaringan yang mempunyai satu layer saja. Jaringan yang berbentuk *multilayer*, yaitu jaringan yang terdiri lebih dari satu layer. Dalam jaringan *single layer*, hanya terdapat satu lapisan dengan bobot-bobot terhubung. *Neuron-neuron* pada jaringan *single layer* dapat dikelompokkan menjadi dua bagian, yaitu unit *input* dan unit *output*. Unit *input* menerima masukan dari luar,

kemudian masukan tersebut langsung diolah menjadi keluaran tanpa harus melalui lapisan tersembunyi. Semua unit *input* akan langsung berhubungan dengan unit *output*.

Jaringan *multilayer*, selain ada unit *input* dan *output*, juga terdapat unit yang tersembunyi (*hidden units*) diantara lapisan *input* dan lapisan *output*. Jumlah *hidden units* tersebut tergantung pada kebutuhan. Semakin kompleks jaringan, *hidden units* yang dibutuhkan makin banyak, demikian pula jumlah *layer* nya. Jaringan *multilayer* sering dipakai untuk persoalan yang lebih rumit karena pelatihan untuk hal yang komplek akan lebih berhasil jika menggunakan jaringan *multilayer* dibandingkan menggunakan jaringan *single layer*.

2.2.2.3 Jaringan Syaraf Tiruan Backpropagation

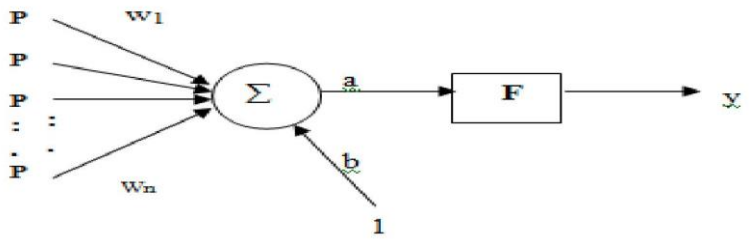
JST backpropagation atau rambat balik adalah metode yang paling sederhana dan mempunyai konsep belajar yang mudah dipahami dibandingkan metode-metode yang lain. JST backpropagation pertama kali diperkenalkan oleh Rumelhart, Hinton dan William pada tahun 1986, kemudian Rumelhart dan Mc Clelland mengembangkannya metode ini pada tahun 1988. JST backpropagation akan mengubah bobot biasnya untuk mengurangi perbedaan antara *output jaringan* dan target *output*. Pelatihan akan terus dilakukan hingga bobot yang dicapai pada jaringan tersebut dianggap ideal atau telah mencapai *minimum error*. Setelah pelatihan selesai, dilakukan pengujian terhadap jaringan yang telah dilatih. Pembelajaran algoritma jaringan syaraf membutuhkan perambatan maju dan diikuti dengan perambatan mundur (bolak-balik), hal ini menyebabkan proses

pembelajarannya menjadi lama karena memakan waktu. Backpropagation merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh perceptron dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyi. Secara umum konsep pelatihan dari JST backpropagation adalah :

1. Belajar dari kesalahan.
2. Memasukan secara umpan maju (*feed forward*) pola-pola masukan.
3. Menghitung dan backpropagation kesalahan yang besangkutan.
4. Mengatur bobot-bobot koneksi.

2.2.2.4 Arsitektur Jaringan Syaraf Tiruan Backpropagation

Sebuah *neuron* dengan jumlah masukan P dan bobot w pada tiap koneksi *input* ditunjukkan pada gambar 2.3 Jumlah masukan bobot dan bias menghasilkan masukan total ke fungsi aktivasi *f*. Neuron-neuron dapat menggunakan sembarang fungsi aktivasi *f* yang ada diferensialnya untuk menghasilkan keluaran.



Gambar 2.3 Satu *layer* jaringan sebagai penyusun *multilayer.*

Keterangan :

p = Masukan (*input*)

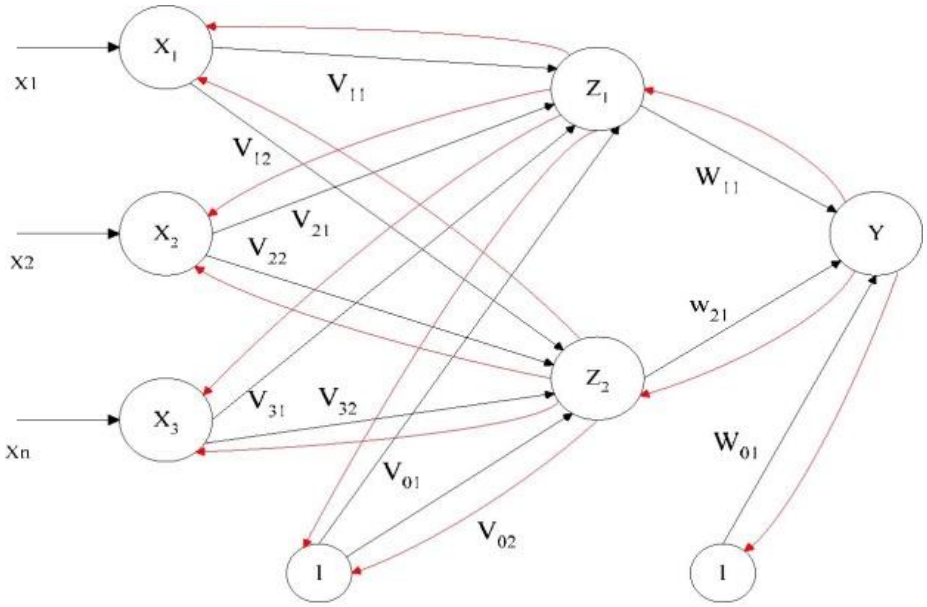
w = Bobot pada lapisan keluaran

b = Bias

F = Fungsi aktivasi

y = Keluaran (*output*) hasil

Arsitektur jaringan yang paling umum digunakan untuk jaringan syaraf tiruan metode backpropagation adalah jaringan *multilayer*. Pada lapisan tersembunyi, dimungkinkan untuk menggunakan lebih dari satu lapisan.



Gambar 2.4 Arsitektur jaringan backpropagation

Gambar 2.4 merupakan salah satu contoh arsitektur jaringan syaraf tiruan metode

backpropagation. Jaringan terdiri atas 3 unit (neuron) pada lapisan *input,* yaitu X_1, X_2, dan X_3; 1 lapisan tersembunyi dengan 2 neuron, yaitu Z_1 dan Z_2; serta 1 unit pada lapisan *output,* yaitu Y. Bobot yang menghubungkan X_1, X_2, dan X_3 dengan neuron pertama pada lapisan tersembunyi, adalah V_{11}, V_{21}, dan V_{31} Sedangkan bobot yang menghubungkan X_1, X_2, dan X_3 dengan neuron kedua pada lapisan tersembunyi, adalah V_{12}, V_{22}, dan V_{32} (V_{ij} : bobot yang menghubungkan neuron *input* ke-i ke neuron ke-j pada lapisan tersembunyi). V_{01} dan V_{02} adalah bobot bias yang menuju ke neuron pertama dan kedua pada lapisan tersembunyi. Bobot yang menghubungkan Z_1 dan Z_2 dengan neuron pada lapisan *output,* adalah W_1 dan W_{21} (W_{jk} : bobot yang menghubungkan neuron lapisan tersembunyi ke-j ke neuron ke-k pada lapisan *output).* Bobot bias W_{01} menghubungkan lapisan tersembunyi dengan lapisan *output.* Algoritma backpropagation menggunakan error *output* untuk mengubah nilai bobot-bobotnya dalam arah mundur *(backward)* pada gambar ditunjukkan dengan panah berwarna merah. Sedangkan untuk mendapatkan error tahap perambatan maju *{feed forward)* harus dikerjakan terlebih dahulu, pada gambar ditunjukkan dengan panah berwarna hitam. Pada saat perambatan maju, neuron-neuron diaktifkan dengan fungsi aktivasi, perhitungan bobot-bobot neuron pada langkah *feed forward* hanya didasarkan pada vektor masukan, sedangkan pada *backward* bobot-bobot diperhalus dengan memperhitungkan nilai target. Fungsi aktivasi yang digunakan, antara lapisan *input* dan lapisan tersembunyi, dan antara lapisan tersembunyi dengan lapisan *output* adalah fungsi aktivasi sigmoid biner (tidak diperlihatkan pada gambar).

2.2.2.5 Fungsi Aktivasi Jaringan Syaraf Tiruan Backpropagation

Karakteristik yang harus dimiliki oleh fungsi aktivasi Jaringan Syaraf Tiruan adalah kontinu, diferensiabel dan tidak menurun secara monoton. Fungsi aktivasi pada Jaringan Syaraf Tiruan Backpropagation terdiri dari beberapa metode:

a. *Sigmoid biner* :

$$f(x) = \frac{1}{1+e^{-x}} \quad \text{.....(2.1)}$$

dengan turunan

$$f'(x) = f(x)[1 - f(x)]$$

b. *Sigmoid bipolar :*

$$g(x) = 2f(x) - 1 = \frac{2}{1+e^{-x}} - 1$$

$$= \frac{1-e^{-x}}{1+e^{-x}} \quad \text{.....(2.2)}$$

dengan turunan

$$g'(x) = \frac{1}{2}[1 + g(x)][1 - g(x)]$$

c. *Tangen Hiperbolik* :

$$y = f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

$$= \frac{1-e^{-2x}}{1+e^{-2x}} \quad \text{.....(2.3)}$$

dengan turunan

$$f'(x) = [1 + f(x)][1 - f(x)]$$

Fungsi aktivasi *sigmoid biner* mempunyai nilai keluaran berkisar antara 0 dan 1,

sedang *sigmoid bipolar* memiliki range antara -1 dan 1.

2.2.2.6 Algoritma Pelatihan Jaringan Syaraf Tiruan Backpropagation

Algoritma pelatihan pada jaringan syaraf tiruan backpropagation adalah sebagai berikut kusumadewi (2003) :

Langkah 0 : Inisialisasi nilai bobot (diatur pada nilai acak yang kecil).

Langkah 1 : Selama kondisi berhenti masih tidak terpenuhi, lakukan langkah 2-9.

Langkah 2 : Untuk setiap pasangan pelatihan, lakukan langkah 3-8.

Perambatan maju *(Feedforward)* :

Langkah 3 :Tiap unit masukan (X_i, i=1,…,n) menerima sinyal masukan x_i dan menyebarkan sinyal itu ke semua unit pada lapisan di atasnya (lapisan tersembunyi).

Langkah 4 : Setiap unit lapisan tersembunyi (Z_j, j=1,…,p) dihitung nilai masukan dengan menggunakan nilai bobotnya :

$$z\_in_j = \upsilon_{0j} + \sum_{i=1}^{n} x_i \upsilon_{ij} \quad \text{.....(2.4)}$$

V0j = bias pada unit tersembunyiy, kemudian dihitung nilai keluaran dengan menggunakan fungsi aktivasi yang dipilih :

$$z_j = f(z\_in_j) \quad \text{.....(2.5)}$$

Hasil fungsi tersebut dikirim ke semua unit pada lapisan di atasnya (unit keluaran).

Langkah 5 : Tiap unit keluaran *(Y_k k=1,..,m)* dihitung nilai masukan dengan menggunakan nilai bobotnya :

$$y\_in_k = w_{0k} + \sum_{j=1}^{p} z_j w_{jk} \quad \text{.....(2.6)}$$

w_{0k} = bias pada unit keluaran *k,* kemudian dihitung nilai keluaran dengan menggunakan fungsi aktivasinya :

$y_k = f(y\_in_k)$ ..... (2.7)

Perambatan balik *(Backpropagation)* :

Langkah 6 : Tiap unit keluaran *(Y_k , k=1,..,m)* menerima pola target yang berhubungan

dengan pola masukan pelatihan, dan kemudian dihitung kesalahan

informasinya :

$\delta_k = (t_k - y_k) f'(y\_in_k)$ ..... (2.8)

Kemudian dihitung koreksi nilai bobotnya yang kemudian akan digunakan untuk memperbaharui nilai bobot w_{jk} :

$\Delta w_{jk} = \alpha \delta_k z_j$ ..... (2.9)

Hitung koreksi nilai biasnya yang kemudian akan digunakan untuk memperbaharui nilai w_{0k} :

$$\Delta w_{0k} = \alpha\delta_k \quad \text{..... (2.10)}$$

dan kemudian nilai δ_k dikirim ke unit pada lapisan di bawahnya.

Langkah 7 : Setiap unit lapisan tersembunyi (Z_j, $j=1, ...,p$) dihitung perubahan masukan yang dari unit-unit pada lapisan di atasnya :

$$\delta_{in_j} = \sum_{k=1}^{m} \delta_k w_{jk} \quad \text{.....(2.11)}$$

Kemudian nilai tersebut dikalikan dengan nilai turunan dari fungsi aktivasinya untuk menghitung informasi kesalahannya :

$$\delta_j = \delta\_in_j f'(z\_in_j) \quad \text{..... (2.12)}$$

Hitung koreksi nilai bobot yang kemudian digunakan untuk memperbaharui nilai v_{ij} :

$$\Delta vij = \alpha\delta_j X_i \quad \text{.....(2.13)}$$

dan hitung nilai koreksi bias yang kemudian digunakan untuk memperbaharui o_{OJ} :

$$\Delta v_{oj} = \alpha\delta_j \quad \text{.....(2.14)}$$

Memperbaharui nilai bobot dan bias :

Langkah 8 : Tiap unit *(Y_k , k=1,…,m)* keluaran diperbaharui nilai bias dan bobotnya *(j=0,...,p):*

$$w_{jk}(baru) = w_{jk}(lama) + \Delta w_{jk} \quad \text{.....(2.15)}$$

dan pada tiap unit lapisan tersem tersembunyi (Zj, j = 1,…,p) diperbaharui bias dan bobotnya (i = 1, …,n):

$$v_{ij}(baru) = v_{ij}\,(lama) + \Delta v_{ij} \quad \text{..... (2.16)}$$

Langkah 9 : Menguji apakah kondisi berhenti sudah terpenuhi. Jika kondisi berhenti telah terpenuhi, pelatihan jaringan dapat dihentikan. Untuk menentukan kondisi berhenti terdapat dua cara yang biasa dipakai, yaitu :

a. Pertama, dengan membatasi iterasi yang ingin dilakukan.
b. Kedua, dengan membatasi *error*. Pada metode Backpropagation, dipakai metode *Mean Square Error* untuk menghitung rata-rata *error* antara *output* yang dikehendaki pada *training* data dengan *output* yang dihasilkan oleh jaringan. Cara memeriksa kondisi berhenti dengan *Mean Square Error* adalah sebagai berikut : *Mean Square Error* :

$$\widehat{\mathrm{MSE}}(\hat{\theta}) - \frac{1}{n}\sum_{j=1}^{n}(\theta_j - \theta)^2 \quad \text{.....(2.17)}$$

Suatu jangka waktu (*epoh*) adalah satu set putaran vektor-vektor pelatihan. Beberapa *epoh* diperlukan untuk pelatihan sebuah jaringan syaraf tiruan backpropagation. Dalam algoritma ini dilakukan perbaikan bobot setelah masing-masing pola pelatihan dimasukan. Setelah pelatihan selesai bobot-bobot yang telah diperbaiki disimpan.

2.2.2.7 Prosedur Pengujian Jaringan Syaraf Tiruan Backpropagation

Setelah pelatihan, sebuah jaringan syaraf backpropagation diaplikasikan hanya pada fase umpan maju *{feedforward).* Prosedur aplikasinya adalah sebagai berikut:

Langkah 0 : Inisialisasi bobot awal (hasil dari pelatihan).

Langkah 1 : Untuk tiap vektor masukan, lakukan langkah 2-4.

Langkah 2: Untuk setiap unit masukan, distribusikan masukan *xi* ke setiap unit diatasnya(unit dalam).

Langkah 3 : Untuk j = 1,...,p ; digunakan persamaan berikut :

$$z\_in_j = v_{0j} + \sum_{i=1}^{n} x_i v_{ij} \quad \text{.....(2.17)}$$

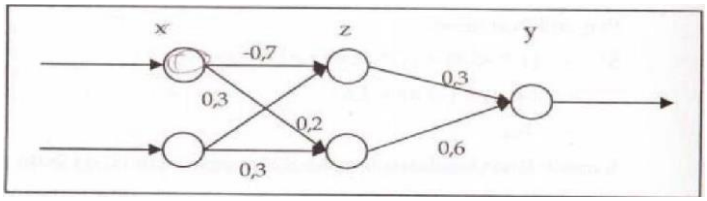
$$z_j = f(z\_in_j) \quad \text{.....(2.18)}$$

Langkah 4 : Untuk k = 1,...,m ; digunakan persamaan sebagai berikut :

$$y\_in_k = w_{0k} + \sum_{j=1}^{p} z_j w_{jk} \quad \text{.....(2.19)}$$

$$y_k = f(y\_in_k) \quad \text{.....(2.20)}$$

Contoh :



Gambar 2.5 contoh jaringan syaraf tiruan single layer

Diketahui :

- Pola input = 1 1
- Output yang
- diinginkan 1
- Konstanta belajar 0,4
- Fungsi aktivasi : = $\frac{1}{1+e^{-x}}$

Hitung perubahan bobot setelah iterasi 1 dilakukan.

Jawab : tiap unit tersembunyi menjumlahkan isyarat masukan terbobot pada input ke1

$$Z_{in} = - 0,7 \times 1 + 0,3 \times 1 = 0,4$$

Fungsi aktivasi hitung :

$$z = \frac{1}{1+e^{-0,5}} = 0,622$$

Tiap unit tersembunyi menjumlahkan isyarat masukan terbobot pada input ke 2

$$Z_{in2} = 0,2 \times 1 + 0,3 \times 1 = 0,5$$

Fungsi aktivasi hitung :

$$z = \frac{1}{1+e^{-0,5}} = 0,622$$

Menjumlahkan isyarat masukan berbobot pada tiap unit keluaran :

$$Y_{in} = (0,3 \times 0,401) + (-0,6 \times 0,622) = 0,2529$$

Fungsi aktivasi hitung :

$$Y = \frac{1}{1+e^{0,2529}} = 0,437$$

Tiap unit keluaran menerima pola sasaran berkaitan dengan pola pelatihan masukannya.

Hitung galat informasi :

$$\delta_y = (t - y) f'(Y_{in})$$

= (1- 0,437)* f (-0,2529) (1- f (-0,2529))

= 0,138

Hitung koreksi bobot dan prasikapnya :

Δt = 0,138 x 0,4 x 0,401 = 0,0221

Δt = 0,138 x 0,4 x 0,622 = 0,0343

Tiap unit tersembunyi menjumlahkan delta masukannya (dari unit lapisan di atasnya):

δinz_1= 0,3 x 0,318 = - 0,041

δz_1 = 0,041 f (-0,4) (l-f(-0,4)) = 0,009

δinz_2= -0,6 x 0,318 = - 0,08

δz_2= 0,08 f (-0,5) (l-f(-0,5)) = -0,01

Tiap unit keluaran memperbaharui bobot dan prasikap :

Δt 1 = 0,009 x 0,4 x 1 = 0,0036

Δt 2= 0,009 x 0,4 x l = 0,0036

Δt 3= -0,01 x 0,4 x 1 = -0,004

Δt 4= 0,01 x 0,4 x 1 = -0,004

2.3 Saham

Saham dapat didefinisikan tanda penyertaan atau kepemilikan seseorang atau badan dalam suatu perusahaan atau perseroan terbatas. Wujud saham adalah selembar kertas yang menerangkan bahwa pemilik kertas tersebut adalah pemilik perusahaan yang menerbitkan surat berharga tersebut. Porsi kepemilikan ditentukan oleh seberapa besar penyertaan yang ditanamkan di perusahaan tersebut (Darmadji dan Fakhruddin, 2001). Suatu perusahaan dapat menjual hak kepemilikannya dalam bentuk saham. Suatu perseroan terbatas mengeluarkan sertifikat saham kepada pemiliknya sebagai bukti investasi mereka dalam usaha. Satuan dasar dari modal saham adalah lembar saham. Suatu perseroan terbatas mengeluarkan sertifikat saham untuk sejumlah lembar saham yang diinginkan. Saham yang ada ditangan pemegang saham disebut saham beredar. Total jumlah saham dalam peredaran pada tiap waktu mewakili seratus persen kepemilikan perseroan terbatas disebut modal saham.

Ada beberapa sudut pandang untuk membedakan saham (Darmadji dan Fakhruddin, 2001) :

2.3.1 Ditinjau dari segi kemampuan dalam hak tagih atau klaim

2.3.1.1 Saham Biasa

Mewakili klaim kepemilikan pada penghasilan dan aktiva yang dimiliki perusahaan. Pemegang saham biasa memiliki kewajiban yang terbatas. Artinya, jika perusahaan bangkrut, kerugian maksimum yang ditanggung oleh pemegang

saham adalah sebesar investasi pada saham tersebut.

2.3.1.2 Saham Preferen

Saham yang memiliki karakteristik gabungan antara obligasi dan saham biasa, karena bisa menghasilkan pendapatan tetap (seperti bunga obligasi), tetapi juga bisa tidak mendatangkan hasil, seperti yang dikehendaki investor. Serupa saham biasa karena mewakili kepemilikan ekuitas dan diterbitkan tanpa tanggal jatuh tempo yang tertulis di atas lembaran saham tersebut; dan membayar deviden. Persamaannya dengan obligasi adalah adanya klaim atas laba dan aktiva sebelumnya, devidennya tetap selama masa berlaku dari saham, dan memiliki hak tebus dan dapat dipertukarkan (*convertible*) dengan saham biasa.

2.3.2 Ditinjau Dari Cara Peralihannya

2.3.2.1 Saham Atas Unjuk

Pada saham atas unjuk (bearer stock) tersebut tidak tertulis nama pemiliknya, agar mudah dipindahtangankan dari satu investor ke investor lainnya. Secara hukum, siapa yang memegang saham tersebut, maka dialah diakui sebagai pemiliknya dan berhak untuk ikut hadir dalam RUPS.

2.3.2.2 Saham Atas Nama

Saham atas nama atau *registered stock* Merupakan saham yang ditulis dengan jelas siapa nama pemiliknya, di mana cara peralihannya harus melalui prosedur tertentu.

2.3.3 Ditinjau dari kinerja perdagangan

2.3.3.1 *Blue – Chip Stocks*

Saham biasa dari suatu perusahaan yang memiliki reputasi tinggi, sebagai *leader* di industri sejenis, memiliki pendapatan yang stabil dan konsisten dalam membayar dividen.

2.3.3.2 *Income Stocks*

Saham dari suatu emiten yang memiliki kemampuan membayar dividen lebih tinggi dari rata – rata dividen yang dibayarkan pada tahun sebelumnya. Emiten seperti ini biasanya mampu menciptakan pendapatan yang lebih tinggi dan secara teratur membagikan dividen tunai. Emiten ini tidak suka menekan laba dan tidak mementingkan potensi.

2.3.3.3 Growth Stocks

2.3.3.3.1 (*Well - Known*)

Saham – saham dari emiten yang memiliki pertumbuhan pendapatan yang tinggi, sebagai *leader* di industri sejenis yang mempunyai reputasi tinggi.

2.3.3.3.2 (*Lesser – Known*)

Saham dari emiten yang tidak sebagai leader dalam industri, namun memiliki ciri *growth stock*. Umumnya saham ini berasal dari daerah dan kurang populer di kalangan emiten.

2.3.3.3 *Speculative Stock*

Saham suatu perusahaan yang tidak bisa secara konsisten memperoleh penghasilan dari tahun ke tahun, akan tetapi mempunyai kemungkinan penghasilan yang tinggi di masa mendatang, meskipun belum pasti.

2.3.3.4 *Counter Cyclical Stocks*

Saham yang tidak terpengaruh oleh kondisi ekonomi makro maupun situasi bisnis secara umum. Pada saat resesi ekonomi, harga saham ini tetap tinggi, di mana emitennya mampu memberikan dividen yang tinggi sebagai akibat dari kemampuan emiten dalam memperoleh penghasilan yang tinggi pada masa resesi.

Dan yang terbaru jenis saham yang diperdagangkan di BEI , yaitu ETF (*Exchange Trade Fund*) adalah gabungan reksadana terbuka dengan saham dan pembelian di bursa seperti halnya saham di pasar modal bukan di Manajer Investasi (MI) . ETF dibagi 2, yaitu: ETF index dan close and ETF. ETF *index* adalah menginvestasikan dana kelolanya dalam sekumpulan portofolio efek yang terdapat pada satu indeks tertentu dengan proporsi yang sama. Sedangkan *Close* and ETFs yaitu *fund* yang diperdagangkan dibursa efek yang berbentuk perusahaan investasi tertutup dan dikelola secara aktif.

2.4 Flow – Chart

Bagan alir program (*program flow-chart*) adalah suatu bagan yang menggambarkan arus logika dari data yang akan diproses dalam suatu program dari awal sampai akhir. Bagan alir program merupakan alat yang berguna bagi *programmer* untuk mempersiapkan program yang rumit. Bagan alir terdiri dari

simbol-simbol yang mewakili fungsi-fungsi langkah program dan garis alir (*flowlines*) menunjukkan urutan dari simbol-simbol yang akan dikerjakan.

Berikut ini adalah simbol-simbol program *flowchart* menurut ANSI(*American National standard Institute*) :

- Simbol terminal (*terminal symbol*)

digunakan untuk menunjukkan awal dan akhir dari program

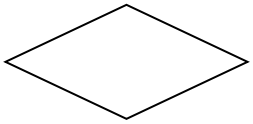
- Simbol persiapan (*preparation symbol*)

digunakan untuk memberikan nilai awal pada suatu variabel atau counter.

- Simbol pengolahan (*processing symbol*)

digunakan untuk pengolahan aritmatika dan pemindahan data.

- Simbol keputusan (*dacision symbol*)

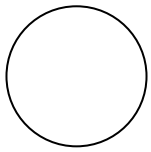


digunakan untuk mewakili operasi perbandingan logika.

- Simbol proses terdefinisi(*predefined process symbol*)

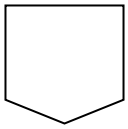
digunakan untuk proses yang detilnya dijelaskan terpisah, misalnya dalam bentuk subroutine.

- Simbol penghubung (*connector symbol*)



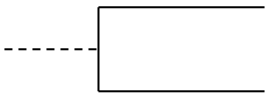
digunakan untuk menunjukkan hubungan arus proses yang terputus masih dalam halaman yang sama.

- Simbol penghubung halaman lain (*off page connector symbol*)



digunakan untuk menunjukkan hubungan arus proses yang terputus sudah berbeda halaman.

- Simbol penjelasan (*annotation flag symbol*)

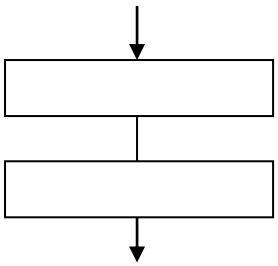


digunakan untuk memberikan keterangan-keterangan guna memperjelas simbol-simbol yang lain.

Berikut ini adalah beberapa bentuk dasar struktur logika yang diwakili oleh bagan alir :

1. Struktur urut sederhana (*simple sequence structure*)

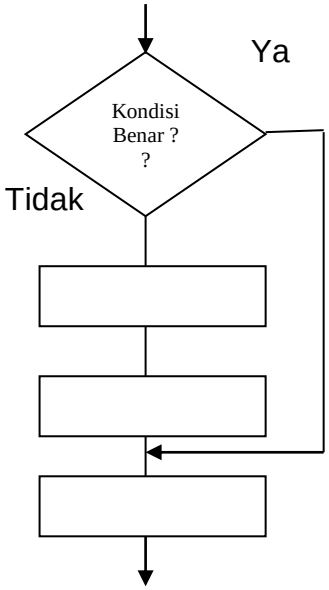
   Struktur ini semata-mata hanya berisi langkah-langkah yang urut saja, satu diikuti yang lainnya.



Gambar 2.6 Struktur urut sederhana

2. Struktur loncat (*branch structure*)

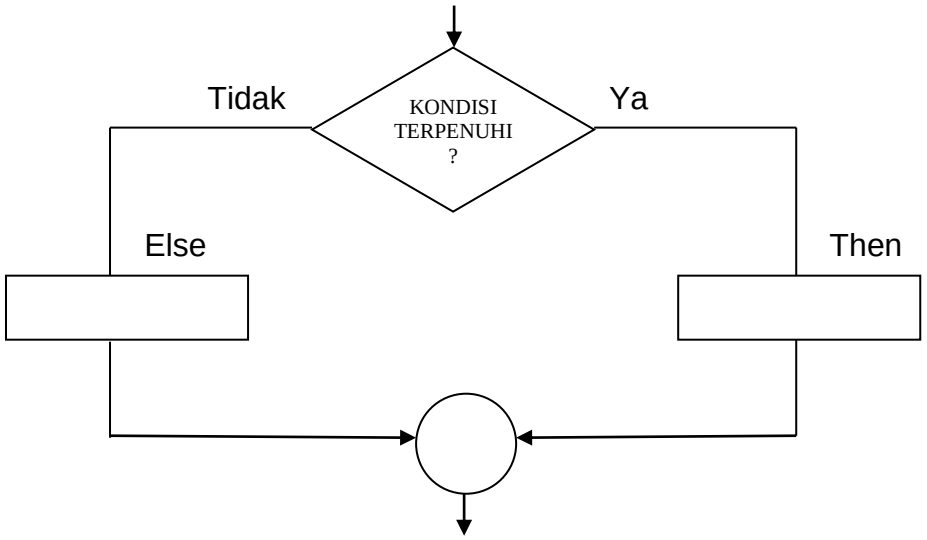
   Struktur berisi suatu loncatan ke proses tertentu oleh statemen GOTO atau statemen IF.



Gambar 2.7 Struktur loncat

3. Struktur seleksi (*selection structure*)

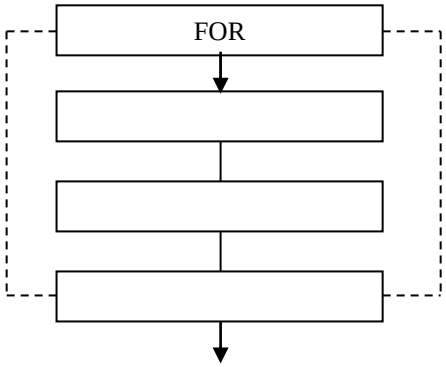
   Structure ini merupakan penyelesaian kondisi yang menggunakan statemen IF-THEN-ELSE.



Gambar 2.8 Struktur seleksi

4. Struktur perulangan FOR (*FOR loop structure*)

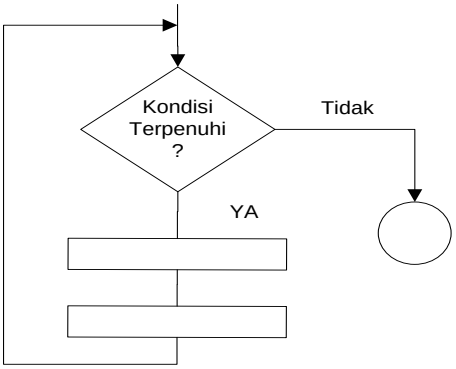
   Struktur ini merupakan perulangan beberapa blok statemen yang dibentuk dengan statemen FOR.



Gambar 2.9 Struktur perulangan FOR

5. Struktur perulangan DO WHILE (*DO WHILE loop structure*)

   Struktur ini menunjukkan suatu blok statemen akan dikerjakan (DO) berulang-ulang selama (WHILE) kondisi yang diseleksi masih terpenuhi dan akan keluar dari lingkungan *loop* bila kondisi sudah tidak terpenuhi.

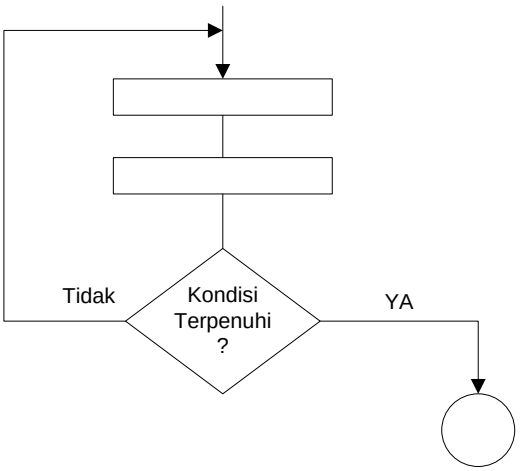


Gambar 2.10 Struktur perulangan DO WHILE

6. Struktur perulangan DO UNTIL (*DO UNTIL loop structure*).

   Struktur ini menunjukkan suatu blok statemen akan dikerjakan (DO) sampai

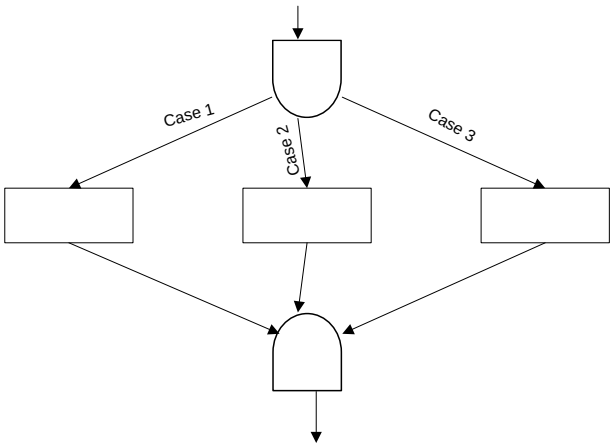
(UNTIL) kondisi yang diseleksi tidak terpenuhi.



Gambar 2.11 Struktur perulangan DO UNTIL

7. Struktur CASE (*Case strukture*)

   Struktur ini akan memproses sebuah blok statemen pada salah satu kondisi case yang terpenuhi dari sejumlah case yang ada.



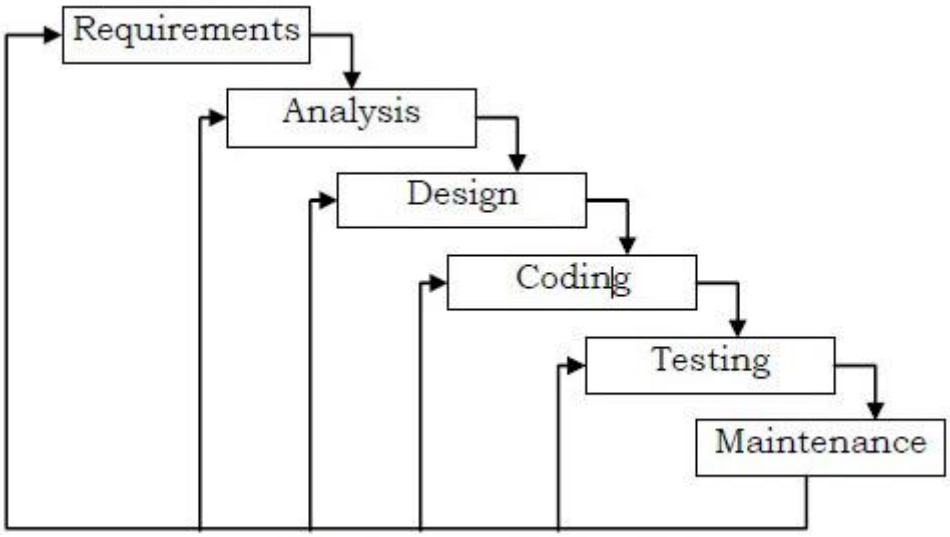
Gambar 2.12 Struktur CASE

2.5 Metode Waterfall

Dalam perancangan aplikasi pada tugas akhir ini penulis menggunakan metode *Waterfall*. Metode *Waterfall* adalah metode yang menyarankan sebuah

pendekatan yang sistematis dan sekuensial melalui tahapan-tahapan yang ada pada SDLC untuk membangun sebuah perangkat lunak.

Gambar menjelaskan bahwa metode Waterfall menekankan pada sebuah keterurutan dalam proses pengembangan perangkat lunak. Metode ini adalah sebuah metode yang tepat untuk membangun sebuah perangkat lunak yang tidak terlalu besar dan sumber daya manusia yang terlibat dalam jumlah yang terbatas.



Gambar 2.13 Diagram Waterfall Model

Dari gambar 2.13 dapat dilihat bahwa tahapan pada metode *Waterfall* diawali oleh tahap analisis kebutuhan yang merupakan tahap awal pembangunan sebuah perangkat lunak. Tahap ini didefinisikan sebagai sebuah tahap yang menghasilkan sebuah kondisi yang diperlukan oleh pengguna untuk menyelesaikan permasalahan ataupun mencapai sebuah tujuan. Tahap ini bertujuan untuk mengumpulkan kebutuhan-kebutuhan pengguna dan kemudian mentransformasikan ke dalam sebuah deskripsi yang jelas dan lengkap.

Tahapan kedua adalah tahap analisis sistem yang bertujuan untuk menjabarkan segala sesuatu yang nantinya akan ditangani oleh perangkat lunak.

Tahapan ini adalah tahapan dimana pemodelan merupakan sebuah representasi dari object di dunia nyata. Untuk memahami sifat perangkat lunak yang akan dibangun, analis harus memahami domain informasi, dan tingkah laku yang diperlukan.

Tahap ketiga adalah tahap perancangan perangkat lunak yang merupakan proses multi langkah dan berfokus pada beberapa atribut perangkat lunak yang berbeda yaitu struktur data, arsitektur perangkat lunak, dan detil algoritma. Proses ini menerjemahkan kebutuhan ke dalam sebuah model perangkat lunak yang dapat diperkirakan kualitasnya sebelum dimulainya tahap implementasi.

Tahap implementasi adalah tahap yang mengkonversi apa yang telah dirancang sebelumnya ke dalam sebuah bahasa yang dimengerti komputer. Kemudian komputer akan menjalankan fungsi-fungsi yang telah didefinisikan sehingga mampu memberikan layanan-layanan kepada penggunanya.

Tahap selanjutnya adalah tahap pengujian. Terdapat dua metode pengujian perangkat lunak yang umum digunakan, yaitu metode *black-box* dan *white-box*. Pengujian dengan metode *black-box* merupakan pengujian yang menekankan pada fungsionalitas dari sebuah perangkat lunak tanpa harus mengetahui bagaimana struktur di dalam perangkat lunak tersebut. Sebuah perangkat lunak yang diuji menggunakan metode *black-box* dikatakan berhasil jika fungsi-fungsi yang ada telah memenuhi spesifikasi kebutuhan yang telah dibuat sebelumnya. Sedangkan metode *white-box* menguji struktur internal perangkat lunak dengan melakukan pengujian pada algoritma yang digunakan oleh perangkat lunak.

Tahap akhir dari metode *Waterfall* adalah tahap perawatan. Tahap ini dapat diartikan sebagai tahap penggunaan perangkat lunak yang disertai dengan

perawatan dan perbaikan. Perawatan dan perbaikan suatu perangkat lunak diperlukan, termasuk di dalamnya adalah pengembangan, karena dalam prakteknya ketika perangkat lunak tersebut digunakan terkadang masih terdapat kekurangan ataupun penambahan fitur-fitur baru yang dirasa perlu.

2.6 MATLAB (*Matrix Laboratory*)

Bahasa pemrograman sebagai media untuk berinteraksi antara manusia dan komputer saat dibuat semakin mudah dan cepat. Sebagai contoh, dapat dilihat dari perkembangan bahasa pemrograman Pascal yang terus memunculkan varian baru sehingga akhirnya menjadi Delphi, demikian pula dengan Basic dengan Visual Basicnya serta C dengan C + + Buildernya. Pada akhirnya semua bahasa pemrograman akan semakin memberikan kemudahan pemakainya (*programmer*) dengan penambahan fungsi-fungsi baru yang sangat mudah digunakan bahkan oleh pemakai tingkat pemula.

MATLAB muncul di dunia bahasa pemrograman yang cenderung dikuasai oleh bahasa yang telah mapan. Tentu saja sebagai bahasa pemrograman yang baru MATLAB akan sukar mendapat hati dari pemakai. Namun MATLAB hadir tidak dengan fungsi dan karakteristik yang umumnya ditawarkan bahasa pemrograman lain yang biasanya hampir seragam. MATLAB dikembangkan sebagai bahasa pemrograman sekaligus alat visualisasi, yang menawarkan banyak kemampuan untuk menyelesaikan berbagai kasus yang berhubungan langsung dengan disiplin keilmuan matematika. MATLAB memiliki kemampuan mengintegrasikan komputasi, visualisasi, dan pemrograman dalam sebuah lingkungan yang tunggal dan mudah digunakan. MATLAB menyediakan beberapa

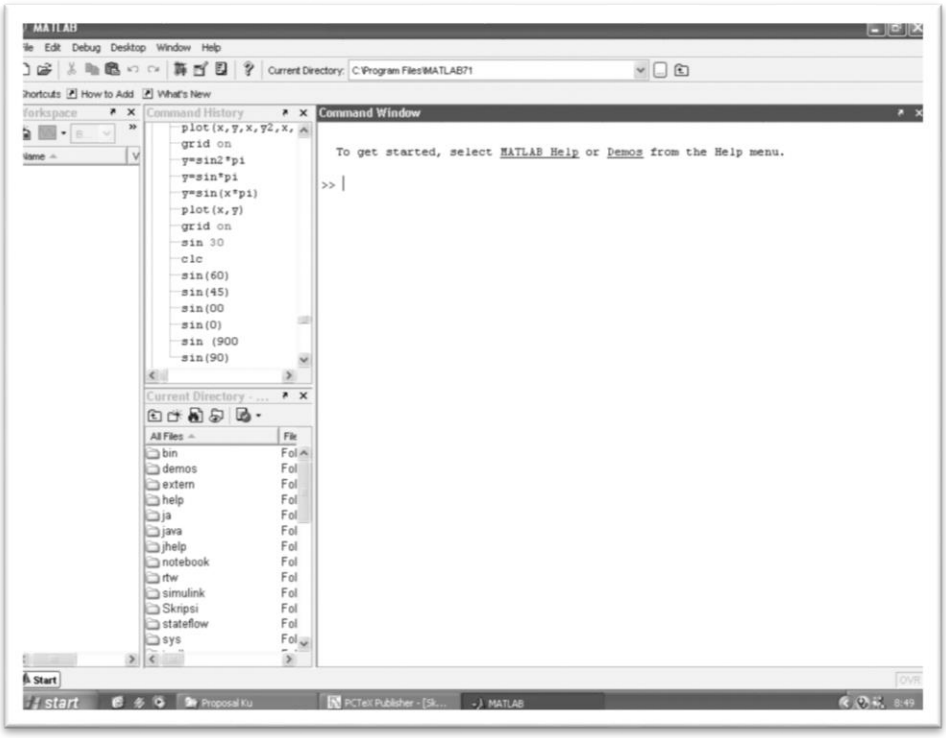
pilihan untuk dipelajari, mempelajari metode visualisasi saja, pemrograman saja, atau kedua-duanya. MATLAB adalah bahasa pemrograman level tinggi yang dikhususkan untuk komputasi teknis. Bahasa ini mengintegrasikan kemampuan komputasi, visualisasi, dan pemrograman dalam sebuah lingkungan yang tunggal dan mudah digunakan. MATLAB memberikan sistem interaktif yang menggunakan konsep *array* sebagai standar variabel elemennya tanpa membutuhkan pendeklarasian *array* seperti pada bahasa pemrograman lain.

Kehadiran MATLAB sebagai bahasa pemrograman memberikan jawaban sekaligus tantangan. MATLAB menyediakan beberapa pilihan untuk dipelajari, mempelajari metoda visualisasi saja, pemrograman saja, atau kedua-duanya. MATLAB memang dihadirkan bagi orang-orang yang tidak ingin disibukkan dengan rumitnya sintaks dan alur logika pemrograman, sementara pada saat yang sama membutuhkan hasil komputasi dan visualisasi yang maksimal untuk mendukung pekerjaannya. Selain itu, MATLAB juga memberikan kemudahan bagi *programmer/developer* program yaitu untuk menjadi pembanding yang sangat handal, hal tersebut dapat dilakukan karena kekayaannya akan fungsi matematika, fisika, statistika, dan visualisasi.

2.6.1 Lingkungan Kerja MATLAB

Sebagaimana bahasa pemrograman lainnya, MATLAB juga menyediakan lingkungan kerja terpadu yang sangat mendukung dalam membangun sebuah aplikasi. Pada setiap versi MATLAB terbaru, lingkungan terpadunya akan semakin dilengkapi. Lingkungan tepadu ini terdiri dari beberapa *form* yang memiliki kegunaan masing-masing. Setiap pertama kali membuka aplikasi MATLAB, maka

akan diperoleh beberapa *form*. MATLAB akan menyimpan *mode/setting* terakhir lingkungan kerja yang digunakan sebagai *mode/setting* lingkungan kerja pada saat membuka aplikasi MATLAB diwaktu berikutnya.



Gambar 2.14 Window Utama MATLAB

2.6.2 GUIDE MATLAB

GUIDE atau *GUI Builder* merupakan sebuah *Graphical User Interface* (GUI) yang dibangun dengan objek grafis seperti tombol (*pushbutton*), *edit*, *slider*, *text*, *combo*, sumbu (*axes*), maupun *menu* dan lain-lain untuk kita gunakan. Sebagai contoh, ketika menggerakkan slider, maka kita dapat melihat perubahan sebuah nilai. Kemudian ketika kita menekan tombol OK, maka aplikasi akan dijalankan. Aplikasi yang menggunakan GUI umumnya lebih mudah dipelajari dan digunakan

karena orang yang menjalankannya tidak perlu mengetahui perintah yang ada dan bagaimana perintah bekerja.

Jika membicarakan pemrograman berorientasi visual, yang paling dikenal adalah sederetan bahasa pemrograman seperti Visual Basic, Delphi, Visual C, Visual Fox Pro, dan yang lainnya yang memang didesain untuk itu. MATLAB merintis ke arah pemrograman yang menggunakan GUI dimulai dari MATLAB versi 5, yang terus disempurnakan hingga sekarang.

Tidak seperti bahasa pemrograman lainnya, GUIDE MATLAB memiliki banyak keunggulan tersendiri, antara lain:

1. GUIDE MATLAB cocok untuk aplikasi-aplikasi berorientasi sains.
2. MATLAB memiliki banyak fungsi *built-in* yang siap digunakan dan pemakai tidak perlu repot membuatnya sendiri.
3. Ukuran file, baik Fig-file maupun M-file yang dihasilkan relatif kecil.
4. Kemampuan grafisnya cukup handal dan tidak kalah dengan bahasa pemrograman lainnya.