

# **TUGAS AKHIR**

## **STUDI KOMPARASI PREDIKSI PENJUALAN MOBIL MENGGUNAKAN JARINGAN SYARAF TIRUAN *BACKPROPAGATION* , *AUTOREGRESSIVE* DAN *MOVING AVERAGE***

**Diajukan sebagai Salah Satu Syarat  
untuk Memperoleh Gelar Sarjana Strata – 1**

**Teknik Industri**



**Oleh**

**Nama : Ulfa Rahmi**

**No. Mahasiswa : 02 522 259**

**JURUSAN TEKNIK INDUSTRI  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
YOGYAKARTA  
2011**



## LEMBAR PERNYATAAN KEASLIAN

Demi Allah, saya akui karya ini adalah hasil kerja saya sendiri kecuali nukilan dan ringkasan yang setiap satunya telah saya jelaskan sumbernya. Jika di kemudian hari ternyata bukti pengakuan saya ini melanggar aturan yang sah dan hak dalam karya tulis intelektual maka saya bersedia ijazah yang telah saya terima untuk ditarik oleh Universitas Islam Indonesia.

Yogyakarta, Juni 2011

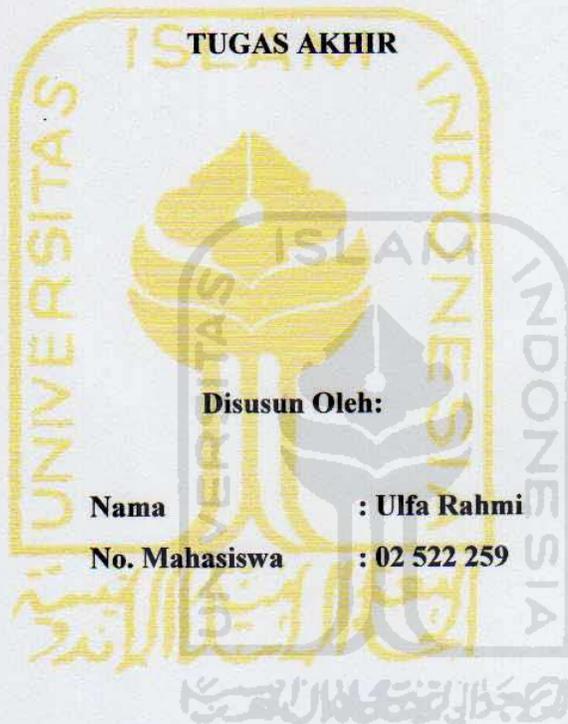


*[Signature]*  
Ulfa Rahmi

02522259

**LEMBAR PENGESAHAN PEMBIMBING**

**STUDI KOMPARASI PREDIKSI PENJUALAN MOBIL MENGGUNAKAN  
JARINGAN SYARAF TIRUAN *BACKPROPAGATION*, *AUTOREGRESSIVE* DAN  
*MOVING AVERAGE***



**TUGAS AKHIR**

**Disusun Oleh:**

**Nama : Ulfa Rahmi**

**No. Mahasiswa : 02 522 259**

**Yogyakarta, Juni 2011**

**Dosen Pembimbing**

A handwritten signature in black ink, appearing to be 'Yuli Agusti Rochman', is written over a horizontal line.

**(Yuli Agusti Rochman, ST., M.Eng)**

**LEMBAR PENGESAHAN PENGUJI**

**STUDI KOMPARASI PREDIKSI PENJUALAN MOBIL MENGGUNAKAN  
JARINGAN SYARAF TIRUAN *BACKPROPAGATION*, *AUTOREGRESSIVE* DAN  
*MOVING AVERAGE***

Oleh

Nama : Ulfa Rahmi

No Mahasiswa : 02 522 259

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat  
untuk Memperoleh Gelar Sarjana Jurusan Teknik Industri

Fakultas Teknologi Industri

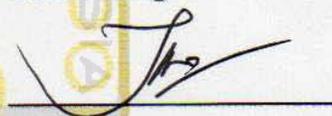
Universitas Islam Indonesia

Jogjakarta, Juni 2011

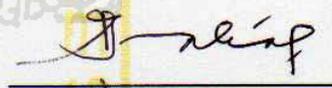
Tim Penguji

Tanda Tangan

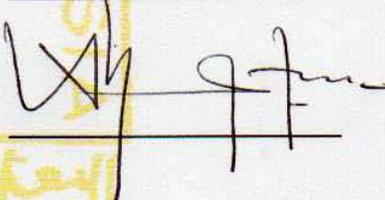
Yuli Agusti Rochman, ST., M.Eng  
Ketua



Ir. Ali Parkhan, MT  
Anggota I



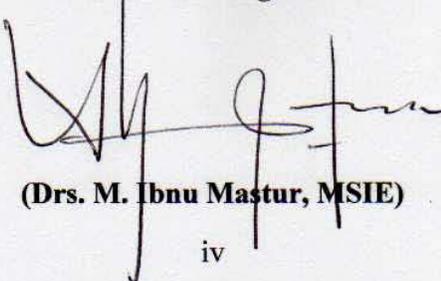
Drs. M. Ibnu Mastur, MSIE  
Anggota II



Mengetahui,

Ketua Jurusan Teknik Industri

Fakultas Teknologi Industri



(Drs. M. Ibnu Mastur, MSIE)

20  
6 2011

## HALAMAN PERSEMBAHAN



Alhamdulillah, atas izin Allah SWT tugas akhir ini dapat terselesaikan.

Kupersembahkan hasil karya kecilku ini kepada orang yang paling berarti dalam

hidupku :

Ayahanda Muchsin dan Ibunda Zuraida tercinta yang tak pernah letih menguntai do'a,

kasih sayang dan telah memberikan segala yang terbaik dalam hidupku.

Keluarga besarku serta Kakakku Bani Rafdinal

yang telah menjadi motivasiku meraih cita-cita.

## MOTTO

إِنَّ مَعَ الْعُسْرِ يُسْرًا ﴿٦﴾

فَإِذَا فَرَغْتَ فَانصَبْ ﴿٧﴾

*“Sesungguhnya sesudah kesulitan itu ada kemudahan. Maka apabila kamu telah selesai (dari sesuatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain”*

(Q.S. Al Insyirah 6 - 7)

ketenteraman hati orang-orang yang beriman serta penubuhannya bagi mereka

الَّذِينَ ءَامَنُوا وَتَطْمَئِنُّ قُلُوبُهُمْ بِذِكْرِ اللَّهِ أَلَا بِذِكْرِ اللَّهِ تَطْمَئِنُّ  
الْقُلُوبُ ﴿٢٨﴾

*“Ingatlah, hanya dengan mengingat Allah-lah hati menjadi tenteram”*

(Q.S. AR Ra'ad 28)

يَا أَيُّهَا الَّذِينَ ءَامَنُوا اسْتَعِينُوا بِالصَّبْرِ وَالصَّلَاةِ إِنَّ اللَّهَ مَعَ الصَّابِرِينَ

*“Hai orang-orang yang beriman, jadikanlah sabar dan shalat sebagai penolongmu, sesungguhnya Allah beserta orang-orang yang sabar”*

(Q.S. Al Baqarah 153)

## KATA PENGANTAR



*Assalamu'alaikum Wr. Wb.*

Alhamdulillah, segala puji dan syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan segala rahmat dan karuniaNya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **Studi Komparasi Prediksi Penjualan Mobil Menggunakan Jaringan Syaraf Tiruan *Backpropagation*, *Autoregressive* Dan *Moving Average*.**

.Laporan Tugas Akhir ini disusun sebagai salah satu syarat memperoleh gelar Sarjana Teknik Industri Jurusan Teknologi Industri Universitas Islam Indonesia. Penulis juga menyadari bahwa terselesaikannya laporan Tugas Akhir ini tidak lepas dari bantuan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak Gumbolo Hadi Susanto, Ir., MSc selaku Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.
2. Bapak Drs. M. Ibnu Mastur, MSIE selaku Ketua Jurusan Prodi Teknik Industri, yang senantiasa mendidik penulis dalam menyelesaikan perkuliahan.
3. Bapak Yuli Agusti Rochman, ST., M.Eng selaku dosen pembimbing yang telah berkenan memberikan bimbingan, saran dan waktunya dalam pembuatan Tugas Akhir ini.
4. Kedua Orang Tua, abang dan adik serta keluarga besar penulis tercinta, yang telah memberikan kasih dan sayangnya, perhatian, do'a restu serta dukungan kepada penulis yang tak ternilai harganya.

5. Pimpinan pengurus Gaikindo serta Bapak Juwono Andrianto selaku pembimbing penelitian di lapangan.
6. Teman-teman dan pihak-pihak lain yang tidak bisa penulis sebutkan satu per satu, yang telah membantu dalam pelaksanaan dan penyusunan Tugas Akhir ini terima kasih atas perhatian dan dukungannya.

Sebagai manusia yang tidak luput dari kesalahan, penulis menyadari bahwa masih banyak kekurangan-kekurangan dalam penulisan laporan Tugas Akhir ini, untuk itu penulis dengan senang hati menerima segala kritik dan saran yang membangun untuk kesempurnaan laporan ini.

*Wassalamu'alaikum Wr, Wb*



Yogyakarta, Juni 2011

Penulis

## ABSTRAK

*Prediksi merupakan alat penting dalam menentukan segala sesuatu agar efektif dan efisien. Selama ini banyak peramalan dilakukan secara intuitif dengan menggunakan metode-metode statistik. Metode statistik tersebut, antara lain metode smoothing, Box-Jenkins, regresi dan sebagainya. Pemilihan metode tersebut tergantung pada berbagai aspek yang mempengaruhi yaitu aspek waktu, pola data, tipe model sistem yang diamati, tingkat keakuratan forecast atau prediksi yang diinginkan dan sebagainya. Oleh sebab itu, akan muncul suatu masalah apabila pengamatan atau pengujian dilakukan pada suatu sistem dinamis yang memiliki sistem pola data dengan formulasi yang selalu berubah-ubah seperti halnya sistem prediksi penjualan mobil. Metode peramalan yang digunakan yaitu menggunakan metode runtun waktu adalah peramalan yang menggunakan serangkaian pengamatan terhadap suatu peristiwa, kejadian, gejala, atau variabel yang diambil dari waktu ke waktu. Atau dengan kata lain, peramalan yang menggunakan serangkaian data masa lampau. Model peramalan untuk proses stasioner yaitu Autoregressive dan Moving Average, dan metode peramalan yang lain menggunakan teknologi jaringan syaraf tiruan (JST). Dengan JST identifikasi pola data dari sistem prediksi penjualan mobil dapat dilakukan dengan metode pendekatan pembelajaran atau pelatihan. Berdasarkan kemampuan belajar yang dimilikinya, maka JST dapat dilatih untuk mempelajari dan menganalisa pola data masa lalu dan berusaha mencari suatu formula atau fungsi yang akan menghubungkan pola data masa lalu dengan keluaran yang diinginkan. Penelitian ini bertujuan untuk mengetahui bagaimana melakukan prediksi penjualan mobil dengan menggunakan JST Backpropagation dan untuk mengetahui besarnya tingkat keakuratan forecast atau ramalan yang dihasilkan oleh JST Backpropagation, Autoregressive dan Moving Average. Prosedur yang digunakan, yaitu perumusan masalah, pengumpulan data dan analisis data yang terdiri dari perancangan model prediksi penjualan mobil menggunakan JST Backpropagation, Autoregressive dan Moving Average. Dari pengolahan data disimpulkan bahwa metode yang tepat untuk memprediksi penjualan mobil adalah JST Backpropagation dengan nilai MSE 21,55 dibandingkan nilai MSE Autoregressive 36,1 dan Moving Average 25,9. Maka nilai prediksi penjualan mobil berdasarkan metode JST Backpropagation yaitu  $t+1 = 6$  unit,  $t+2 = 6$  unit,  $t+3 = 8$  unit dan  $t+4 = 6$  unit mobil.*

*Kata kunci : forecasting, jaringan syaraf tiruan backpropagation, autoregressive, moving average*

## TAKARIR

Neural Network	= jaringan syaraf tiruan
Backpropagation	= propagasi balik
Backward	= menyebar mundur
Epoch	= siklus
Error	= tingkat kesalahan
Forecasting	= peramalan/prediksi
Feedforward	= perambatan maju
Hidden layer	= lapisan tersembunyi
Input	= masukan
Lead time	= tenggang waktu
Learning rate	= kecepatan pembelajaran
Momentum	= daya gerak
Multi layer perceptron	= perseptron yang memiliki lapisan banyak
Neuron	= sel syaraf
Output	= hasil
Over production	= produksi berlebih
Performance	= kemampuan
Range	= jarak antara
Target	= sesuatu yang ingin dicapai
Testing	= pengujian
Threshold	= ambang batas
Time series	= runtun waktu
Training	= pelatihan
Under production	= produksi kurang

# DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGAKUAN .....	ii
HALAMAN PENGESAHAN PEMBIMBING .....	iii
HALAMAN PENGESAHAN PENGUJI .....	iv
HALAMAN PERSEMBAHAN.....	v
HALAMAN MOTTO .....	vi
KATA PENGANTAR .....	vii
ABSTRAKSI.....	ix
TAKARIR .....	x
DAFTAR ISI .....	xi
DAFTAR TABEL.....	xv
DAFTAR GAMBAR .....	xvi
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian .....	4
1.5 Manfaat Penelitian .....	4
1.6 Sistematika Penulisan .....	5



## BAB II KAJIAN PUSTAKA

2.1 Peramalan ( <i>forecasting</i> ) .....	6
2.1.1 Definisi dan Tujuan Peramalan .....	7
2.2 Metode Peramalan Penjualan .....	7
2.2.1 Metode <i>Autoregressive</i> .....	9
2.2.2 Metode <i>Moving Average</i> .....	11
2.3 Jaringan Syaraf Tiruan .....	13
2.3.1 Sejarah Jaringan Syaraf Tiruan.....	13
2.3.2 Konsep Dasar Jaringan Syaraf Tiruan .....	15
2.3.3 Lapisan Jaringan Syaraf Tiruan .....	17
2.3.4 Arsitektur Jaringan Syaraf Tiruan .....	18
2.3.5 Proses Pembelajaran Jaringan Syaraf Tiruan.....	18
2.3.6 Fungsi Aktivasi .....	21
2.4 Jaringan Syaraf Tiruan <i>Backpropagation</i> .....	23
2.4.1 Konsep Jaringan <i>Backpropagation</i> .....	25
2.4.2 Arsitektur Jaringan <i>Backpropagation</i> .....	25
2.4.3 Pelatihan Jaringan <i>Backpropagation</i> .....	26
2.4.4 Pengujian Data .....	31
2.4.5 Momentum.....	32
2.4.6 Aplikasi <i>Backpropagation</i> dalam Peramalan.....	34

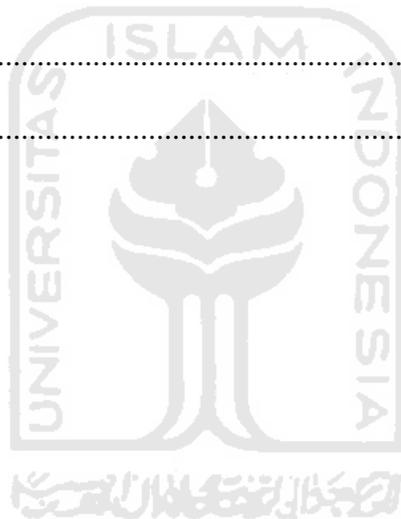
### **BAB III METODOLOGI PENELITIAN**

3.1 Objek Penelitian .....	37
3.2 Pengumpulan Data .....	37
3.2.1 Metode Pengumpulan Data.....	37
3.2.2 Data Yang Diperlukan .....	38
3.3. Diagram Penelitian.....	39
3.4 Analisis Model .....	40
3.4.1 Identifikasi Dan Perumusan Masalah .....	40
3.4.2 Penentuan Tujuan .....	40
3.4.3 Studi Pustaka .....	40
3.4.4 Pengumpulan Data.....	41
3.4.5 Analisis Kebutuhan Sistem.....	47
3.4.5.1 Fungsionalitas Sistem Jaringan Syaraf Tiruan <i>Backpropagation</i> .....	42
3.4.5.2 Fungsionalitas <i>Autoregressive</i> .....	47
3.4.7 Hasil Analisis.....	49
3.4.8 Kesimpulan dan Saran .....	49

### **BAB IV PENGUMPULAN DAN PENGOLAHAN DATA**

4.1 Data Umum Tempat Objek Penelitian .....	50
4.1.1 Sejarah Berdirinya Gaikindo .....	50
4.1.2 Visi dan Misi.....	52
4.1.3 Strategi Gaikindo .....	52
4.2 Pengumpulan Data .....	53

4.3 Perancangan JST <i>Backpropagation</i> .....	54
4.3.1 Implementasi Metode JST <i>Backpropagation</i> dengan Matlab.....	56
4.4 Perancangan <i>Autoregressive</i> .....	64
4.4.1 Implementasi Metode <i>Autoregressive</i> dengan Matlab.....	64
4.5 Pengolahan Data dengan Metode <i>Moving Average</i> .....	72
 <b>BAB V PEMBAHASAN</b>	
5.1 Pembahasan .....	73
 <b>BAB VI KESIMPULAN DAN SARAN</b>	
6.1 Kesimpulan.....	76
6.2. Saran .....	76
 <b>DAFTAR PUSTAKA</b>	
<b>LAMPIRAN</b>	



## DAFTAR TABEL

Tabel 4.1 Tabel Data Penjualan Mobil Baleno Tahun 2004 .....	53
Tabel 4.2 Tabel Hasil Peramalan Metode JST.....	62
Tabel 4.3 Tabel Hasil Peramalan Metode <i>Autoregressive</i> .....	69
Tabel 4.4 Tabel Hasil Peramalan Metode <i>Moving Average</i> .....	70
Tabel 5.1 Tabel Perbandingan 3 Metode Peramalan .....	73



## DAFTAR GAMBAR

Gambar 2.1 Gambar Struktur <i>Neuron</i> Jaringan Syaraf.....	15
Gambar 2.2 Gambar Fungsi Aktivasi pada Jaringan Syaraf Sederhana .....	16
Gambar 2.3 Gambar Jaringan Syaraf dengan Lapisan Tunggal .....	18
Gambar 2.4 Gambar Jaringan Syaraf dengan Banyak Lapisan .....	19
Gambar 2.5 Gambar Jaringan Syaraf dengan Lapisan Kompetitif .....	20
Gambar 2.6 Gambar Fungsi <i>Sigmoid Biner</i> .....	22
Gambar 2.7 Gambar Fungsi <i>Sigmoid Bipolar</i> .....	22
Gambar 2.8 Gambar Fungsi Aktivasi <i>Linear</i> (Identitas) .....	23
Gambar 2.9 Gambar Arsitektur <i>Backpropagation</i> .....	25
Gambar 3.1 Gambar Diagram Alir Penelitian .....	39
Gambar 3.2 Gambar Diagram Sistem Penghitungan <i>Backpropagation</i> Prediksi Penjualan Mobil.....	42
Gambar 3.1 Gambar <i>Flowchart</i> Perhitungan <i>Autoregressive</i> .....	47
Gambar 4.1 Gambar Logo Lama dan Baru Gaikindo.....	51
Gambar 4.2 Gambar Grafik Penjualan Mobil Baleno 2006.....	54
Gambar 4.3 Gambar Tampilan <i>Worksheet</i> Matlab .....	56
Gambar 4.4 Gambar Tampilan Jendela Editor Pelatihan.....	56
Gambar 4.5 Gambar Tampilan <i>Neural Network Training</i> .....	57
Gambar 4.6 Gambar Grafik Pelatihan Jaringan di Normalisasi (kiri) dan Grafik Pelatihan Denormalisasi (kanan) .....	57
Gambar 4.7 Gambar Tampilan Jendela Hasil <i>Error</i> Pelatihan.....	58

Gambar 4.8 Gambar Tampilan Jendela Hasil MSE Pelatihan .....	58
Gambar 4.9 Gambar Tampilan Jendela Editor Pengujian.....	59
Gambar 4.10 Gambar Grafik Pelatihan Jaringan di Normalisasi (kiri) dan Grafik Pelatihan Denormalisasi (kanan) .....	59
Gambar 4.11 Gambar Tampilan Jendela Hasil <i>Error</i> Pengujian.....	60
Gambar 4.12 Gambar Tampilan Jendela Hasil MSE Pengujian.....	61
Gambar 4.13 Gambar Tampilan Jendela <i>Command Window</i> .....	64
Gambar 4.14 Gambar Tampilan Jendela <i>System Identification Tool</i> .....	65
Gambar 4.15 Gambar Tampilan Jendela Linear Parameter Model.....	65
Gambar 4.16 Gambar Tampilan Jendela <i>System Identification Tool</i> dengan 13 Percobaan .....	66
Gambar 4.17 Gambar Tampilan Grafik dari Pelatihan.....	66
Gambar 4.18 Gambar Tampilan Jendela <i>Command Window</i> .....	67
Gambar 4.19 Gambar Tampilan Jendela <i>Command Window</i> untuk MSE Arx [240 180 1].....	68
Gambar 4.20 Gambar Grafik Hasil Pengujian dengan Arx [240 180 1] (kiri) dan Hasil Pengujian dengan Arx [300 100 1] (kanan) .....	69

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Perkembangan di sektor ekonomi yang mengalami peningkatan akhir-akhir ini juga membawa pengaruh terhadap kegiatan transaksi yang ada, baik perdagangan dan investasi usaha. Sebuah perusahaan dalam menjalankan bisnisnya tidak terlepas dari strategi pemasaran yang digunakan agar dapat bersaing di pasar global. Jika perusahaan tersebut berada dalam pasar komoditi, maka perusahaan harus berusaha menjadi produsen yang efisien, sehingga perusahaan dapat menawarkan produk dengan harga bersaing di pasar dan mampu melakukan pelayanan konsumen yang memuaskan, diantaranya menyediakan dan mensuplai produk yang cukup di pasar.

Untuk dapat melakukan penjualan secara efisien, diperlukan adanya suatu peramalan yang tepat, sehingga tidak terjadi kelebihan atau kekurangan produksi dan proses pendistribusian produk dapat berjalan lancar. Peramalan penjualan produk yang tepat akan terwujud jika dibuat suatu perencanaan dan analisis untuk menentukan seberapa besar volume penjualan yang harus dicapai. Saat ini banyak teknik analisis data yang canggih dapat digunakan untuk peramalan, guna mengantisipasi penyimpangan atau ketidakpastian dalam peramalan yang akan berdampak buruk bagi perusahaan.

Peramalan (*forecasting*) memiliki arti yang berbeda-beda dalam dunia bisnis dan memiliki arti yang lebih khusus dari pada menebak. Umumnya pola dan siklus penjualan cenderung tetap. Berdasarkan siklus runtun waktu (*time series*), biasanya penjualan produk cenderung membentuk pola penjualan yang tepat. Dengan demikian,

ramalan dapat dikatakan sebagai perhitungan yang memiliki dasar kuat dan lebih pasti, sehingga hasilnya diharapkan lebih obyektif dibandingkan dengan hanya sekedar melakukan prediksi (menebak). Dengan mencatat penjualan di waktu yang lampau, maka dapat lebih tepat dalam menentukan penjualan di waktu yang akan datang. (Supranto, 2001)

Pada kenyataannya dalam melakukan peramalan penjualan, penjualan yang sebenarnya tidak selalu sama dengan ramalan penjualannya. Perbedaan antara ramalan penjualan dengan penjualan aktual yang terjadi disebut penyimpangan (kesalahan peramalan). Banyak faktor yang dapat mempengaruhi proses peramalan penjualan diantaranya metode peramalan, promosi, *trend*, *new* produk *launching*, diversifikasi produk, penetrasi pasar, selera konsumen yang berubah dan lain-lain. Penyimpangan ramalan penjualan ini yang sering terjadi pada suatu perusahaan, dimana jumlah penjualan aktual lebih tinggi atau rendah dari pada ramalan yang telah ditetapkan oleh bagian produksi.

Salah satu teknologi yang digunakan untuk memprediksi penjualan di masa datang adalah metode jaringan syaraf tiruan *Backpropagation*. Metode ini telah diimplementasikan dalam berbagai aplikasi terutama dalam hal prediksi. Metode jaringan syaraf tiruan (JST) memiliki karakteristik yang menyerupai jaringan syaraf biologi dalam memproses informasi (Marimin, 2005). Melalui proses pelatihan, JST dapat menyimpan pengetahuan dari pola kejadian di masa lampau yang kemudian dapat digunakan untuk memprediksi kejadian yang akan terjadi dimasa akan datang. JST juga diartikan sebuah komputasi yang meniru cara kerja otak manusia, memiliki kemampuan belajar dan beradaptasi terhadap kondisi yang tidak beraturan dan mengolah data yang sedikit untuk melakukan generalisasi. Dengan adanya JST

kendala tersebut dapat diatasi, JST juga merupakan suatu sistem kecerdasan buatan yang tidak memerlukan model matematis. (Puspita. A, dan Eunike, 2007).

Dengan adanya berbagai macam arsitektur jaringan dan pelatihan yang ada pada jaringan syaraf tiruan maka dapat dipilih arsitektur jaringan dan pelatihan yang digunakan agar jaringan syaraf tiruan dapat mempelajari dan menganalisa pola data masa lalu lebih tepat sehingga diperoleh keluaran yang lebih akurat (dengan kesalahan atau *error* yang minimum).

## 1.2 Rumusan Masalah

Berdasarkan latar belakang masalah, maka dapat dirumuskan berbagai permasalahan yang akan dihadapi, antara lain :

1. Berapa besar tingkat keakuratan *forecast* / ramalan yang dihasilkan oleh metode jaringan syaraf tiruan *Backpropagation*, *Auto Regressive* dan *Moving Average*?
2. Berapa nilai prediksi / ramalan untuk penjualan mobil suzuki Baleno 4 hari mendatang?

## 1.3 Batasan Masalah

Batasan masalah bertujuan untuk menyederhanakan penelitian agar sesuai dengan tujuan yang diharapkan serta untuk menghindari kerancuan pembahasan sehingga ruang lingkup pembahasan menjadi jelas dan tidak meluas ke hal-hal lain. Penelitian ini dibatasi oleh hal-hal sebagai berikut :

1. Penelitian dilakukan di Gaikindo.
2. Data yang digunakan adalah data penjualan mobil harian pada tahun 2004 sampai dengan tahun 2008.
3. Menggunakan konsep Jaringan Syaraf Tiruan *Backpropagation*.

#### 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk :

1. Mengetahui keakuratan ramalan yang dihasilkan oleh jaringan syaraf tiruan *Backpropagation*, *Auto Regressive* dan *Moving Average*.
2. Mengetahui nilai prediksi penjualan mobil Suzuki Baleno untuk 4 hari mendatang.

#### 1.5 Manfaat Penelitian

Dengan adanya penelitian ini diharapkan dapat memberikan beberapa manfaat sebagai berikut:

1. Dapat memberikan suatu metode alternatif untuk melakukan peramalan yang mempunyai nilai akurasi yang baik.
2. Dapat bermanfaat bagi para pembaca untuk lebih mengetahui dan memahami tentang jaringan syaraf tiruan yang diaplikasikan untuk peramalan.

#### 1.6 Sistematika Penulisan

Dalam penyusunan tugas akhir ini, akan disusun sistematika penulisan sebagai berikut :

##### **BAB I : PENDAHULUAN**

Bab ini memuat latar belakang masalah, perumusan masalah, batasan masalah, tujuan dan manfaat penelitian serta sistematika penelitian.

##### **BAB II : KAJIAN PUSTAKA**

Bab ini membahas beberapa teori yang berkaitan erat dengan identifikasi lingkup permasalahan, proses analisis dan literature dari peramalan dengan menggunakan jaringan syaraf tiruan *Backpropagation*..

##### **BAB III : METODOLOGI PENELITIAN**

Mengandung uraian tentang kerangka dan bagan alir penelitian, teknik yang dilakukan, model yang dipakai, pembangunan dan pengembangan model, bahan atau materi, alat, tata cara penelitian dan data yang dikaji secara analisis yang akan dipakai.

#### **BAB IV : PENGOLAHAN DATA DAN HASIL PENELITIAN**

Bab ini memuat data-data hasil penelitian, baik data primer maupun data sekunder. Pengolahan data meliputi pengolahan terhadap data-data hasil penelitian.

#### **BAB V : PEMBAHASAN**

Melakukan pembahasan hasil yang diperoleh dalam penelitian, dan kesesuaian hasil dengan tujuan penelitian sehingga dapat menghasilkan sebuah rekomendasi.

#### **BAB VI : KESIMPULAN DAN SARAN**

Berisi tentang kesimpulan terhadap analisis yang dibuat dan saran atas hasil yang dicapai dan permasalahan yang ditemukan selama penelitian, sehingga perlu dilakukan rekomendasi untuk dikaji pada penelitian lanjutan.

## BAB II

### KAJIAN PUSTAKA

#### 2.1 Peramalan (*forecasting*)

Peramalan penjualan (*forecasting*) memegang peranan yang sangat penting dalam perencanaan dan pengambilan keputusan khususnya di bidang produksi dan operasi. Aktivitas manajemen produksi dan operasi menggunakan peramalan permintaan dalam perencanaan yang menyangkut perencanaan produksi, perencanaan pemenuhan kebutuhan bahan, perencanaan kebutuhan tenaga kerja, perencanaan kapasitas produksi, perencanaan desain dan *lay out* fasilitas, penentuan lokasi pabrik, penentuan metode proses produksi, penentuan jumlah mesin dan sebagainya.

Peranan peramalan penjualan (*forecasting*) ini disebabkan adanya tenggang waktu (*lead time*) antara suatu peristiwa atau kebutuhan dengan kebutuhan mendatang. Jadi, *forecasting* merupakan dasar dari perencanaan perusahaan dalam jangka panjang. *Forecasting* dibutuhkan untuk memperkirakan kebutuhan bahan baku, produk, tenaga kerja maupun kebutuhan lain sebagai respon terhadap perubahan permintaan (pasar). (Priyambodo, 2009)

##### 2.1.1 Definisi dan Tujuan Peramalan.

*Forecasting* adalah peramalan atau perkiraan mengenai sesuatu yang belum terjadi (Subagyo, 1986).

Dalam ilmu pengetahuan sosial segala sesuatu itu serba tidak pasti dan sukar untuk diperkirakan secara tepat. Untuk itu perlu adanya *forecast*. Ramalan yang dilakukan pada umumnya akan berdasarkan data yang terdapat di masa lampau yang

dianalisis dengan menggunakan metode-metode tertentu. Di dalam *forecasting* diupayakan agar *forecast* yang dibuat dapat meminimumkan pengaruh ketidakpastian tersebut.

Dengan kata lain *forecasting* bertujuan mendapatkan *forecast* yang bisa meminimumkan kesalahan meramal (*forecast error*) yang biasanya diukur dengan *mean squared error*, *mean absolute error*, *root mean squared error* dan sebagainya (Subagyo, 1986).

## 2.2 Metode Peramalan Penjualan

Secara umum, metode peramalan penjualan dapat dibagi dalam dua kategori utama, yaitu metode Kuantitatif dan metode Kualitatif. Pada metode peramalan penjualan metode Kualitatif merupakan metode subyektif, artinya besarnya angka penjualan ditetapkan berdasarkan asumsi dan estimasi. Metode biasanya digunakan untuk produk baru yang akan diluncurkan kepasar. Sedangkan metode peramalan penjualan metode Kuantitatif didasarkan atas data-data penjualan masa lalu yang kemudian diolah dengan berbagai metode statistik. Metode Kuantitatif dapat dibagi dalam ke dalam deret berkala atau runtun waktu (*time series*) dan metode kasual (*casual*).

Metode kuantitatif sangat beragam dan setiap teknik memiliki sifat, ketepatan dan biaya tertentu yang harus dipertimbangkan. Metode kuantitatif formal didasarkan atas prinsip-prinsip statistik yang memiliki ketepatan tinggi atau dapat meminimalkan kesalahan, lebih sistematis, dan lebih populer dalam penggunaannya. Untuk dapat menggunakan metode kuantitatif terdapat 3 kondisi yang harus dipenuhi, yaitu:

1. Tersedianya informasi tentang masa lalu.
2. Informasi tersebut dapat dikuantitatifkan dalam angka numerik.

3. Adanya asumsi bahwa beberapa pola masa lalu akan terus berlanjut.

Kondisi yang terakhir ini dikenal sebagai asumsi kesinambungan (*assumption of continuity*). Asumsi ini merupakan premis yang mendasari semua metode peramalan kuantitatif.

Metode kuantitatif dapat dibagi menjadi dua bagian yaitu sebagai berikut:

1. Metode Deret Waktu (*Time Series*)

Metode peramalan (*forecasting*) secara *time series* atau sering disebut metode “Deret Waktu” atau “Deret Berkala” didasarkan asumsi bahwa besarnya permintaan yang akan datang dapat diprediksi dari besarnya permintaan pada masa lalu. Langkah penting dalam menggunakan metode peramalan deret waktu adalah dengan mempertimbangkan jenis pola data. Pola data dapat dibedakan menjadi 4 jenis siklus dan *trend*, yaitu:

1. Pola horizontal, terjadi bilamana data berfluktuasi disekitar nilai rata-rata yang konstan.
2. Pola musiman, terjadi bilamana deret permintaan dipengaruhi oleh faktor musiman.
3. Pola siklus, terjadi bilamana dipengaruhi oleh fluktuasi ekonomi jangka panjang (siklus bisnis).
4. Pola *trend*, terjadi bilamana kenaikan/penurunan permintaan didasarkan pada *trend* ekonomi pasar yg berlangsung.

2. Metode *Casual*.

Metode peramalan secara *casual*, didasarkan adanya asumsi bahwa penjualan dipengaruhi oleh berbagai “peristiwa” yang sengaja dibuat yang dapat mempengaruhi penjualan, misalnya promosi, iklan, kegiatan kompetitor, dan lain-lain

Terdapat berbagai macam metode yang dapat digunakan untuk melakukan prediksi nilai sebuah data runtun waktu seperti penjualan atau harga saham. Beberapa diantaranya *Auto Regressive (AR)*, *Moving Average (MA)*, *Autoregressive Integrated Average (ARIMA)*, Jaringan Syaraf Tiruan, Algoritma Genetik dan logika *Fuzzy*.

### 2.2.1 Metode *Auto Regressive*

Metode *Autoregressive* merupakan model yang menggambarkan bahwa variabel dependen dipengaruhi oleh variabel dependen itu sendiri pada periode-periode yang sebelumnya atau autokorelasi dapat diartikan juga sebagai korelasi linier deret berkala dengan deret berkala itu sendiri dengan selisih waktu (lag) 0, 1, 2 periode atau lebih. Bentuk umum model *Autoregressive* dengan ordo p atau dituliskan dengan AR(p) mempunyai persamaan sebagai berikut :

$$Y_t = \mu + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + e_t \dots\dots\dots(2.1)$$

dimana :

$Y_t$  = deret waktu stasioner

$\phi_i$  = parameter auto korelasi ke-i dengan  $i = 1, 2, \dots, p$

$Y_{t-1}, \dots, Y_{t-p}$  = nilai masa lalu yang berhubungan

$e_t$  = nilai kesalahan pada saat t

$\mu$  = nilai konstan

Persamaan umum model AR(p) dapat juga ditulis sebagai berikut :

$$(1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p) Y_t = \mu + e_t \dots\dots\dots(2.2)$$

Dalam hal ini nilai B adalah operator mundur (*Backward shift Operator*), bentuk umum operator bergerak mundur ini ditulis sebagai berikut :

$B^d Y_t = y_{t-d}$ . Artinya jika operator  $B^d$  bekerja pada  $Y_t$  maka menggeser data tersebut sebanyak d periode kebelakang.

Model *Autoregressive* yang sering dijumpai dalam praktek adalah model AR(1) dan AR(2).

Persamaan AR(1) ditulis dengan :  $(1 - \phi_1 B)Y_t = \mu + e_t \dots\dots\dots(2.3)$

Persamaan AR(2) ditulis dengan :  $(1 - \phi_1 B - \phi_2 B^2)Y_t = \mu + e_t \dots\dots\dots(2.4)$

Model diatas disebut sebagai model *Autoregressive* (regresi diri sendiri) karena model tersebut mirip dengan persamaan regresi pada umumnya, hanya saja yang menjadi variabel independen bukan variabel yang berbeda dengan variabel dependen melainkan nilai sebelumnya (lag) dari variabel dependen ( $Y_t$ ) itu sendiri.

Banyaknya nilai lampau yang digunakan oleh model yaitu sebanyak  $p$ , menentukan tingkat model ini. Apabila hanya digunakan satu lag dependen, maka model ini dinamakan model *Autoregressive* tingkat 1 (*first-order autoregressive*) atau AR(1). Apabila nilai yang digunakan sebanyak  $p$  lag dependen, maka model ini dinamakan model *Autoregressive* tingkat  $p$  (*p-th order autoregressive*) atau AR( $p$ ).

### 2.2.2 Metode Moving Average

*Moving Average* merupakan indikator yang paling sering digunakan dan paling standar. *Moving average* sendiri memiliki aplikasi yang sangat luas meskipun sederhana. Dikatakan sederhana karena pada dasarnya metode ini hanya pengembangan dari metode rata-rata.

Rata-rata bergerak tunggal (*Moving average*) untuk periode  $t$  adalah nilai rata-rata untuk  $n$  jumlah data terbaru. Dengan munculnya data baru, maka nilai rata-rata yang baru dapat dihitung dengan menghilangkan data yang terlama dan menambahkan data yang terbaru. *Moving average* ini digunakan untuk memprediksi nilai pada periode berikutnya. Model ini sangat cocok digunakan pada data yang stasioner atau

data yang konstant terhadap variansi, tetapi tidak dapat bekerja dengan data yang mengandung unsur trend atau musiman.

Rata-rata bergerak pada orde 1 akan menggunakan data terakhir ( $F_t$ ), dan menggunakannya untuk memprediksi data pada periode selanjutnya. Metode ini sering digunakan pada data kuartalan atau bulanan untuk membantu mengamati komponen-komponen suatu runtun waktu. Semakin besar orde rata-rata bergerak, semakin besar pula pengaruh pemulusan (*smoothing*).

Dibanding dengan rata-rata sederhana (dari satu data masa lalu) rata-rata bergerak berorde T mempunyai karakteristik sebagai berikut.

1. Hanya menyangkut T periode terakhir dari data yang diketahui.
2. Jumlah titik data dalam setiap rata-rata tidak berubah dengan berjalannya waktu.

Kelemahan dari metode ini adalah :

1. Metode ini memerlukan penyimpanan yang lebih banyak karena semua T pengamatan terakhir harus disimpan, tidak hanya nilai rata-rata.
2. Metode ini tidak dapat menanggulangi dengan baik adanya trend atau musiman, walaupun metode ini lebih baik dibanding rata-rata total.

Diberikan N titik data dan diputuskan untuk menggunakan T pengamatan pada setiap rata-rata (yang disebut dengan rata-rata bergerak orde (T) atau MA(T), sehingga keadaannya adalah sebagai berikut :

$$F_{T+1} = \frac{X_1 + X_2 + \dots + X_T}{T} \dots\dots\dots(2.5)$$

dimana :

$$F_{T+1} = \text{Ramalan untuk periode ke } t+1$$

$X_t$  = Nilai aktual periode ke t

T = Jangka waktu rata-rata bergerak

Menghitung kesalahan ramalan pada *moving average* :

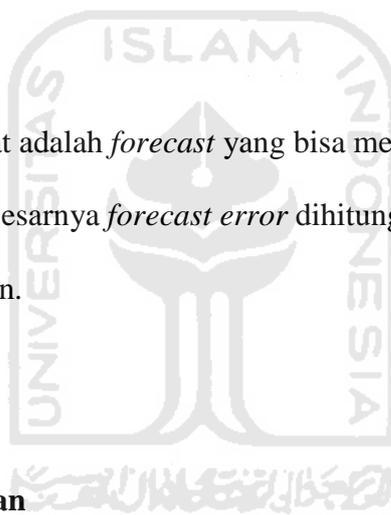
$$\text{Error (E)} = X_t - F_t \dots\dots\dots(2.6)$$

dimana :

$X_t$  = data riil periode ke - t

$F_t$  = ramalan periode ke - t

Hasil peramalan yang akurat adalah *forecast* yang bisa meminimumkan kesalahan meramal (*forecast error*). Besarnya *forecast error* dihitung dengan mengurangi data riil dengan besarnya ramalan.



### 2.3 Jaringan Syaraf Tiruan

Jaringan syaraf tiruan merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran otak manusia tersebut. Istilah buatan digunakan karena jaringan syaraf ini diimplementasikan dengan menggunakan sistem komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran. (Kusumadewi, 2004)

Jaringan syaraf tiruan (JST) berkembang secara pesat pada beberapa tahun terakhir. JST telah dikembangkan sebelum adanya suatu komputer konvensional yang

canggih dan terus berkembang walaupun pernah mengalami masa vakum selama beberapa tahun.

### 2.3.1 Sejarah Jaringan Syaraf Tiruan

Jaringan syaraf tiruan telah dikembangkan sejak tahun 1940-an dimana para ilmuwan menemukan psikologi otak manusia sama dengan pemrosesan yang dilakukan oleh komputer. Pada tahun 1943 McCulloch dan W.H.Pitts merancang model formal yang pertama kali sebagai perhitungan dasar *neuron*. Tahun 1949, Hebb mencoba mengkaji proses belajar yang dilakukan oleh *neuron*. Teori ini dikenal sebagai Hebbian Law. Di tahun 1954, Farley dan Clark mengembangkan model-model untuk relasi adaptif stimulus-respon dalam jaringan random.

Tahun 1958, Rosenblatt memperkenalkan konsep *perceptron* suatu jaringan yang terdiri dari beberapa lapisan yang saling berhubungan melalui umpan maju (*feed forward*). Konsep ini dimaksudkan untuk memberikan ilustrasi tentang dasar-dasar intelegensia secara umum. Hasil kerja Rosenblatt yang sangat penting adalah *perceptron convergence theorem* (1962) yang membuktikan bahwa bila setiap *perceptron* dapat memilah-milah dua buah pola yang berbeda maka siklus pelatihannya dapat dilakukan dalam jumlah yang terbatas.

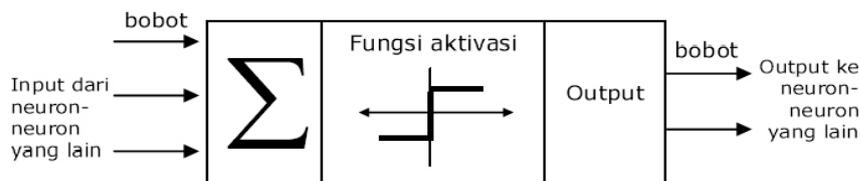
Pada tahun 1960 Widrow dan Hoff menemukan ADALINE (*Adaptive Linear Neuron*). Alat ini dapat beradaptasi dan beroperasi secara *linier*. Penemuan ini telah memperlebar aplikasi jaringan syaraf tiruan tidak hanya untuk pemilihan pola, tetapi juga untuk pengiriman sinyal khususnya dalam bidang *adaptive filtering*. Tahun 1969, Minsky dan Papert melontarkan suatu kritikan tentang kelemahan *perceptronnya* Rosenblatt di dalam memilah-milah pola yang tidak *linier*. Sejak saat itu penelitian di bidang jaringan syaraf tiruan telah mengalami masa vakum untuk kurang lebih satu dasawarsa.

Tahun 1982, Hopfield telah memperluas aplikasi JST untuk memecahkan masalah-masalah optimasi. Hopfield telah berhasil memperhitungkan fungsi energi ke dalam jaringan syaraf yaitu agar jaringan memiliki kemampuan untuk mengingat atau memperhitungkan suatu obyek dengan obyek yang pernah dikenal atau diingat sebelumnya (*associative memory*). Konfigurasi jaringan yang demikian dikenal sebagai *recurrent network*. Salah satu aplikasinya adalah *Travelling Salesman Problem* (TSP).

Pada tahun 1986 Rumelhart, Hinton dan William menciptakan suatu algoritma belajar yang dikenal sebagai propagasi balik (*Backpropagation*). Bila algoritma ini diterapkan pada *perceptron* yang memiliki lapisan banyak (*multi layer perceptron*), maka dapat dibuktikan bahwa pemilahan pola-pola yang tidak *linier* dapat diselesaikan sehingga dapat mengatasi kritikan yang dilontarkan oleh Minsky dan Papert. (Hermawan, 2006)

### 2.3.2 Konsep Dasar Jaringan Syaraf Tiruan

Pada dasarnya JST otak manusia, belajar dari suatu contoh karena mempunyai karakteristik yang adaptif, yaitu dapat belajar dari data-data sebelumnya dan mengenal pola data yang selalu berubah. Selain itu, JST merupakan sistem yang terprogram, artinya semua keluaran atau kesimpulan yang ditarik oleh jaringan didasarkan pada pengalamannya selama mengikuti proses pembelajaran atau pelatihan.



### Gambar 2.1 Struktur *Neuron* Jaringan Syaraf

Hal yang ingin dicapai dengan melatih JST adalah untuk mencapai keseimbangan antara kemampuan memorisasi dan generalisasi. Yang dimaksud kemampuan memorisasi adalah kemampuan JST untuk mengambil kembali secara sempurna sebuah pola yang telah dipelajari. Kemampuan generalisasi adalah kemampuan JST untuk menghasilkan respon yang bisa diterima terhadap pola-pola input yang serupa (namun tidak identik) dengan pola-pola yang sebelumnya telah dipelajari. Hal ini sangat bermanfaat bila pada suatu saat ke dalam JST itu diinputkan informasi baru yang belum pernah dipelajari, maka JST itu masih akan tetap dapat memberikan tanggapan yang baik, memberikan keluaran yang paling mendekati (Puspapitaningrum, 2006)

JST menyerupai otak manusia dalam dua hal, yaitu:

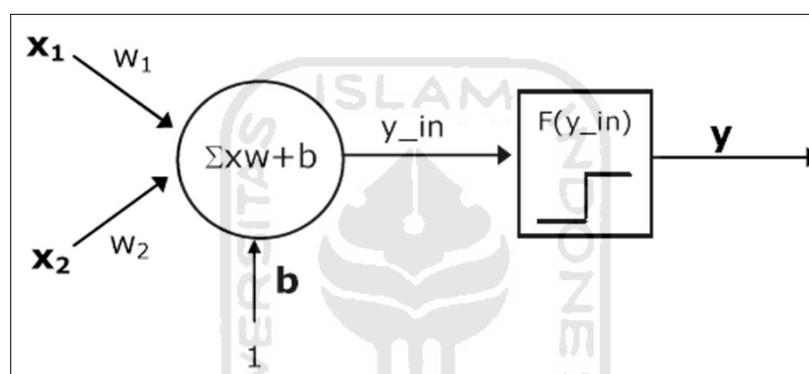
1. Pengetahuan diperoleh jaringan melalui proses belajar.
2. Kekuatan hubungan antar sel syaraf (*neuron*) yang dikenal sebagai bobot-bobot *sinaptik* digunakan untuk menyimpan pengetahuan.

JST dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi, dengan memakai asumsi :

1. Pemrosesan informasi terjadi pada elemen sederhana (*neuron*).
2. Penghubung antar *neuron* memiliki bobot yang akan memperkuat atau memperlemah sinyal.
3. *Neuron* memiliki fungsi aktivasi yang dikenakan pada penjumlahan *input* yang diterima. Besarnya *output* ini kemudian dibandingkan dengan suatu batas ambang (*threshold*).

*Neuron* adalah unit pemroses informasi yang menjadi dasar dalam pengoperasian jaringan syaraf tiruan (Siang, 2005). *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 bobotnya.
3. Fungsi aktivasi yang akan menentukan apakah sinyal dari *input neuron* akan diteruskan ke *neuron* lain atau tidak.



Gambar 2.2 Fungsi Aktivasi pada Jaringan Syaraf Sederhana.

Pada gambar 2.2 tersebut sebuah *neuron* akan mengolah  $N$  *input* ( $x_1, x_2, \dots, x_N$ ) yang masing-masing memiliki bobot  $w_1, w_2, \dots, w_N$  dan bobot bias  $b$ , dengan rumus :

$a$

$$= \sum_{i=1}^N x_i w_i \dots \dots \dots (2.7)$$

Kemudian fungsi aktivasi  $F$  akan mengaktivasi  $a$  menjadi *output* jaringan  $y$ .

JST ditentukan oleh 3 hal :

1. Pola hubungan antar *neuron* (disebut arsitektur jaringan).
2. Metode untuk menentukan bobot penghubung (disebut metode *pelatihan/training*).

3. Fungsi aktivasi, yaitu fungsi yang digunakan untuk menentukan keluaran *neuron*.
4. Suatu unit penjumlah yang akan menjumlah *input-input* sinyal yang sudah dikalikan dengan bobotnya.
5. Fungsi aktivasi yang akan menentukan apakah sinyal dari *input neuron* akan diteruskan ke *neuron* lain atau tidak.

### 2.3.3 Lapisan Jaringan Syaraf Tiruan

Setiap pola-pola informasi *input* dan *output* yang diberikan kedalam JST diproses dalam *neuron*. *Neuron-neuron* tersebut terkumpul di dalam lapisan-lapisan yang disebut *neuron layers*. Lapisan-lapisan penyusun JST tersebut dapat dibagi menjadi 3, yaitu:

#### 1. Lapisan *input*

Unit-unit di dalam lapisan *input* disebut unit-unit *input*. Unit-unit *input* tersebut menerima pola *inputan* data dari luar yang menggambarkan suatu permasalahan.

#### 2. Lapisan tersembunyi.

Unit-unit di dalam lapisan tersembunyi disebut unit-unit *output*. *Output* dari lapisan ini merupakan solusi JST terhadap suatu permasalahan.

#### 3. Lapisan *output*

Unit-unit di dalam lapisan *output* disebut unit-unit *output*. *Output* dari lapisan ini merupakan solusi JST terhadap suatu permasalahan.

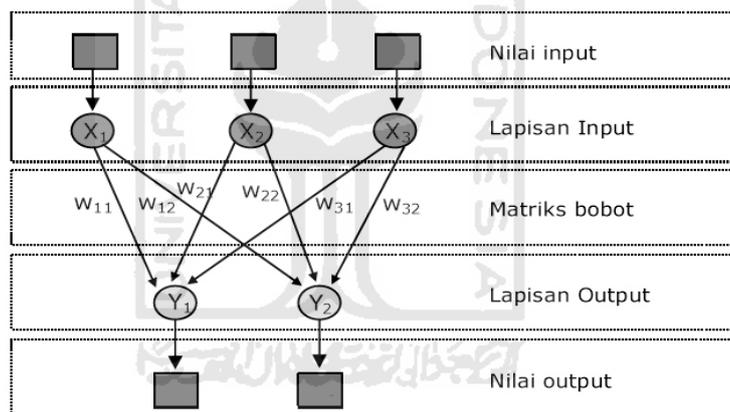
### 2.3.4 Arsitektur Jaringan Syaraf Tiruan

Umumnya, *neuron-neuron* yang terletak pada lapisan yang sama akan memiliki keadaan yang sama. Faktor terpenting dalam menentukan kelakuan suatu *neuron* adalah fungsi dan pola bobotnya. (Kusumadewi, 2003)

Ada beberapa arsitektur jaringan syaraf, antara lain :

a. Jaringan dengan lapisan tunggal (*singel layer net*)

Jaringan dengan lapisan tunggal hanya memiliki satu lapisan dengan bobot-bobot terhubung. Jaringan ini hanya menerima *input* kemudian secara langsung akan mengolahnya menjadi *output* tanpa harus melalui lapisan tersembunyi (Gambar 2.3). Pada gambar 2.3 tersebut, lapisan *input* memiliki 3 *neuron*, yaitu X1, X2 dan X3. Sedangkan pada lapisan *output* memiliki 2 *neuron* yaitu Y1 dan Y2. *Neuron-neuron* pada kedua lapisan saling berhubungan. Seberapa besar hubungan antara 2 *neuron* ditentukan oleh bobot yang bersesuaian. Semua unit *input* akan dihubungkan dengan setiap unit *output*.

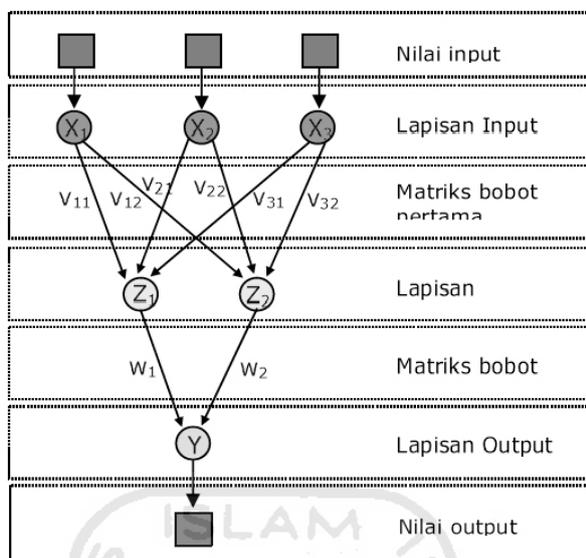


Gambar 2.3 Jaringan Syaraf dengan Lapisan Tunggal

b. Jaringan dengan banyak lapisan (*multilayer net*)

Jaringan dengan banyak lapisan 1 atau lebih lapisan yang terletak diantara lapisan *input* dan lapisan *output* (memiliki 1 atau lebih lapisan tersembunyi), seperti terlihat pada Gambar 2.4 umumnya, ada lapisan bobot-bobot yang terletak antara 2 lapis yang bersebelahan. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih sulit dari pada lapisan dengan lapisan tunggal, tentu saja dengan pembelajaran yang lebih rumit. Namun demikian, pada banyak kasus,

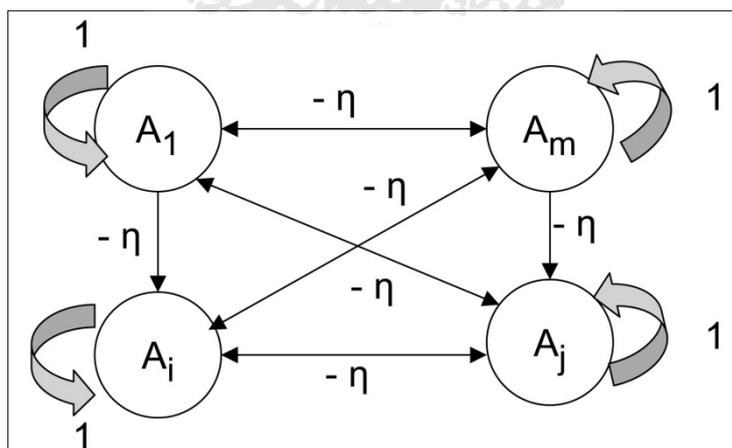
pembelajaran pada jaringan dengan banyak lapisan ini lebih sukses dalam menyelesaikan masalah.



Gambar 2.4 Jaringan Syaraf Dengan Banyak Lapisan

c. Jaringan dengan lapisan kompetitif (*competitiv layer net*)

Umumnya, hubungan antar *neuron* pada lapisan kompetitif ini tidak diperlihatkan pada diagram arsitektur. Gambar 2.5 menunjukkan salah satu contoh arsitektur jaringan dengan lapisan kompetitif yang memiliki bobot  $-\eta$ .



Gambar 2.5 Jaringan Syaraf Dengan Lapisan Kompetitif.

### 2.3.5 Proses Pembelajaran Jaringan Syaraf Tiruan

Keistimewaan dari JST adalah kemampuannya dalam melakukan pembelajaran dari *input* yang diberikan. Agar menghasilkan keluaran yang diinginkan, maka jaringan syaraf harus diberi pembelajaran terhadap bobot-bobot yang telah ditentukan pada tiap *layer* (lapisan). (Kusumadewi, 2004)

Ada dua cara pembelajaran dalam jaringan syaraf tiruan, yaitu :

1. *Supervised Learning* (pembelajaran terawasi)

Pada model ini, seolah-olah jaringan diberikan seorang “guru” yang membimbing proses pembelajarannya. Pembelajaran dilakukan terhadap pola pasangan *input* – *output* yang telah ditentukan oleh pemakai (ahli), dan jaringan diharapkan dapat mempelajari dari pola tersebut. Sehingga jika diberikan *input* lain, maka akan dihasilkan *output* yang sesuai dengan harapan. Beberapa model jaringan yang menggunakan metode pembelajaran terawasi, antara lain : *Delta Rule*, *Backpropagation* dan *Counterpropagation*.

2. *Unsepervised Learning* (pembelajaran tak terawasi)

Model ini kebalikan dari model terawasi di atas, disini jaringan hanya diberikan data *input* tanpa memberikan pasangan *output*nya. Jaringan hanya memodifikasi bobot-bobot data *input* sedemikian rupa, sehingga jika diberikan *input* yang hampir sama akan dihasilkan *output* yang sama pula. Hasil dari model ini adalah berupa *cluster* yaitu pengelompokkan dari data *input*.

### 2.3.6 Fungsi Aktivasi

Fungsi aktivasi merupakan fungsi yang dibangkitkan untuk mengaktifkan *neuron* bila telah melewati suatu nilai ambang tertentu (*threshold*), dengan cara menjumlahkan nilai bobot yang masuk.

Ada beberapa fungsi aktivasi yang digunakan dalam membangun jaringan syaraf tiruan, yaitu :

- a. Fungsi *linear* atau identitas
- b. Fungsi *step biner*
- c. Fungsi *bipolar*
- d. Fungsi *saturating linear*
- e. Fungsi *symmetric saturating linear*
- f. Fungsi *sigmoid biner*
- g. Fungsi *sigmoid bipolar*

Fungsi aktivasi yang sering digunakan adalah fungsi *sigmoid*, baik itu *sigmoid linear* ataupun *bipolar*.

Berikut adalah penyesuaian fungsi *sigmoid* dan *linear*:

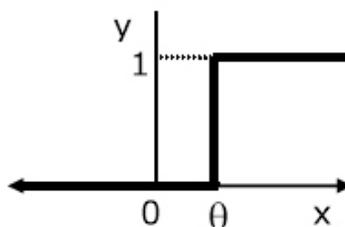
### 1. Sigmoid Biner

Fungsi yang berbentuk kurva S ini biasanya digunakan untuk melatih jaringan syaraf tiruan dengan metode *Backpropagation*. Dimana fungsi ini memiliki *range* nilai antara 0 sampai 1 dengan persamaan, sebagai berikut :

$$y = f(x) = \frac{1}{1 + e^{-\sigma x}} \dots \dots \dots (2.8)$$

dengan :  $f'(x) = \sigma f(x) [1 - f(x)]$

$$y = \begin{cases} 0, & \text{jika } x < \theta \\ 1, & \text{jika } x \geq \theta \end{cases}$$



Gambar 2.6 Fungsi *Sigmoid Biner*

### 2. Sigmoid Bipolar

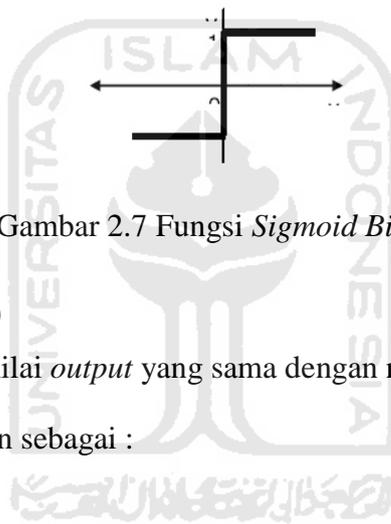
Hampir sama dengan fungsi *sigmoid biner*, perbedaannya pada *range* nilai fungsi ini antara -1 sampai 1, dengan persamaan, sebagai berikut :

$$y = f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \dots\dots\dots(2.9)$$

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

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

$$y = \begin{cases} 1, & \text{jika } x > 0 \\ 0, & \text{jika } x = 0 \\ -1, & \text{jika } x < 0 \end{cases}$$



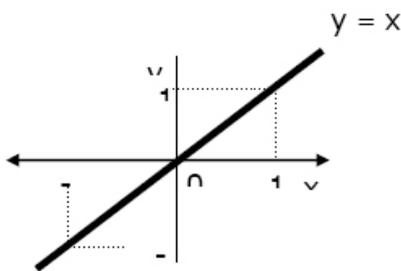
Gambar 2.7 Fungsi *Sigmoid Bipolar*

3. Fungsi *Linear* (Identitas)

Fungsi *linear* memiliki nilai *output* yang sama dengan nilai *inputnya* (gambar 2.8).

Fungsi *linear* dirumuskan sebagai :

$$y = x \dots\dots\dots(2.10)$$



Gambar 2.8 Fungsi Aktivasi *Linear* (Identitas)

**2.4 Jaringan Syaraf Tiruan *Backpropagation***

Algoritma pelatihan *Backpropagation* (propagasi balik), pertama kali dirumuskan oleh Werbos dan dipopulerkan oleh Rumelhart dan McClelland untuk dipakai pada JST. Algoritma ini termasuk metoda pelatihan *supervised* dan didesain untuk operasi pada jaringan *feed forward multi layer*. Algoritma ini merupakan metode optimasi sederhana yang fungsinya untuk meminimalisasi *error* atau kesalahan dari keluaran yang dihasilkan oleh jaringan syaraf tiruan dengan target dari pola yang diinginkan. Pelatihan pada propagasi balik terbagi dalam tiga tahap, yaitu: tahap *feedforward* (propagasi maju) pada *input training pattern* (pola pelatihan masukan), *backpropagation of the associated error* (propagasi balik), dan *adjustment of the weights* (penyesuaian bobot). Setelah proses pembelajaran selesai, maka jaringan siap dipakai untuk mengenali pola, dimana pada proses pengenalan pola hanya tahap propagasi maju yang digunakan.

Pada saat pembelajaran harga laju pembelajaran atau  $\alpha$  digunakan untuk mengontrol besarnya perubahan bobot. Dengan menggunakan harga  $\alpha$  yang kecil, maka dapat mencegah respon yang besar terhadap perhitungan kesalahan pada saat jaringan mempelajari setiap pola.

Pada tahap pembelajaran di dalam algoritma propagasi balik, diperlukan suatu kondisi untuk menghentikan proses pembelajaran. Berbagai kondisi seperti *squared error*, *total squared error*, dan *mean squared error* dapat digunakan untuk menghentikan proses pembelajaran. Definisi dari masing-masing kondisi tersebut diatas adalah sebagai berikut :

1. *Squared error*, adalah jumlah dari masing-masing kuadrat dari perbedaan antara target yang telah ditentukan dengan keluaran yang dihasilkan oleh jaringan setiap *neuron* pada lapisan keluaran.
2. *Total squared error*, adalah jumlah *squared error* untuk setiap pasangan pelatihan.

3. *Mean squared error*, bisa berarti *squared error* dibagi dengan jumlah *neuron* pada lapisan keluaran atau *total squared error* dibagi dengan jumlah pasangan pelatihan.

Pada umumnya digunakan *total squared error* di dalam algoritma propagasi balik, tetapi pada dasarnya ketiga kondisi tersebut mempunyai kemiripan yaitu sebagai informasi *error* yang telah didefinisikan untuk menghentikan proses pembelajaran. Harga dari *error* yang telah didefinisikan, sebelumnya sudah ditentukan dan jika harga *error* yang dihitung pada tahap pembelajaran lebih kecil dari harga *error* yang telah didefinisikan maka proses pembelajaran dihentikan. Harga dari *error* yang didefinisikan untuk menghentikan proses pembelajaran akan menentukan ketepatan jaringan pada saat pengenalan pola dan lamanya proses pembelajaran. Semakin kecil harga *error* yang didefinisikan akan menyebabkan semakin tinggi tingkat ketepatan jaringan pada saat pengenalan pola nanti, tetapi proses pembelajaran menjadi semakin lama dan sebaliknya.

#### **2.4.1 Konsep Jaringan *Backpropagation***

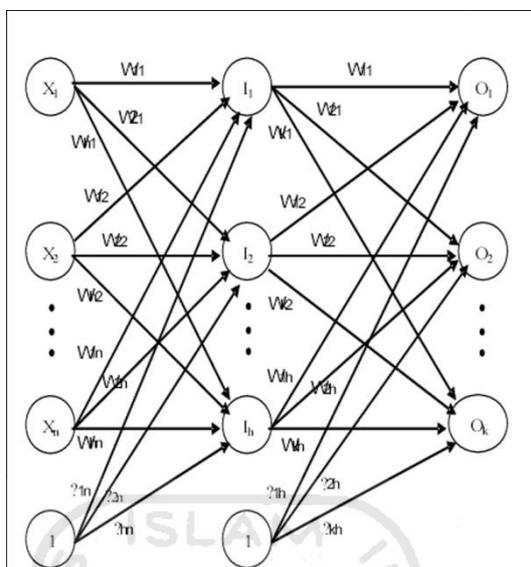
Pada intinya, proses pelatihan dengan metode *Backpropagation* terdiri dari tiga langkah, yaitu :

1. Data-data dimasukkan kedalam lapisan *input*.
2. Perhitungan dengan propagasi balik dari *error* yang bersangkutan.
3. Pembaharuan bobot dan bias.

#### **2.4.2 Arsitektur Jaringan *Backpropagation***

Model jaringan *Backpropagation* adalah *multilayer* dengan minimal memiliki satu lapisan tersembunyi. Arsitektur dari model *Backpropagation* merupakan kombinasi dari sekumpulan sel atau elemen yang tersusun dalam bentuk lapisan serta

membentuk suatu jaringan. Gambar 2.9 merupakan contoh dari arsitektur jaringan *Backpropagation* dengan satu lapisan tersembunyi.



Gambar 2.9 Arsitektur *Backpropagation*

Gambar diatas adalah model *Backpropagation* dengan satu lapisan *input* dengan tiga unit ( $X_1, X_2, X_3$ ), satu lapisan tersembunyi dengan tiga unit ( $I_1, I_2, I_3$ ), satu lapisan *output*nya dengan tiga unit ( $O_1, O_2, O_3$ ) serta dua buah indeks bias masing-masing satu lapisan pada lapisan tersembunyi (1).  $W$  adalah bobot dari unit *input* ke unit tersembunyi, dan bobot dari unit tersembunyi ke unit *output*.

### 2.4.3 Pelatihan Jaringan *Backpropagation*

Aturan pelatihan jaringan propagasi balik terdiri dari dua tahapan, *feedforward* dan *Backpropagation*. Pada jaringan diberikan sekumpulan contoh pelatihan yang disebut set pelatihan. Set pelatihan ini digambarkan dengan sebuah *vector feature* yang disebut dengan vektor *input* yang diasosiasikan dengan sebuah *output* yang menjadi target pelatihannya. Dengan kata lain set pelatihan terdiri dari vektor *input* dan juga vektor *output* target. Keluaran dari jaringan berupa sebuah vektor *output* aktual. Selanjutnya dilakukan perbandingan antara *output* aktual yang dihasilkan dengan *output* target dengan cara melakukan pengurangan diantara kedua *output*

tersebut. Hasil dari pengurangan merupakan *error*. *Error* dijadikan sebagai dasar dalam melakukan perubahan dari setiap bobot yang ada dengan mempropagasikannya kembali. (Puspitaningrum, 2006)

Setiap perubahan bobot yang terjadi dapat mengurangi *error*. Siklus perubahan bobot (*epoch*) dilakukan pada setiap set pelatihan hingga kondisi berhenti dicapai, yaitu bila mencapai jumlah *epoch* yang diinginkan atau hingga sebuah nilai ambang yang ditetapkan terlampaui. (Puspitaningrum, 2006).

Algoritma pelatihan jaringan propagasi balik terdiri dari tiga tahapan yaitu:

1. Tahap umpan maju (*feedforward*)

Pola masukan dihitung maju mulai dari layar masukan hingga layar keluaran menggunakan fungsi aktivasi yang ditentukan.

2. Tahap umpan mundur (*Backpropagation*)

Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan yang terjadi itu dipropagasi mundur. Dimulai dari garis yang berhubungan langsung dengan unit-unit di layar keluaran

3. Tahap pengupdatean bobot dan bias.

Modifikasi bobot untuk menurunkan kesalahan yang terjadi.

Secara rinci algoritma pelatihan jaringan *Backpropagation* dapat diuraikan sebagai berikut:

Langkah 0 : Inisialisasi bobot-bobot, konstanta laju pemahaman ( $\alpha$ ), toleran *error* atau nilai bobot (bila menggunakan nilai bobot sebagai kondisi berhenti) atau set maksimal *epoch* (jika menggunakan banyaknya *epoch* sebagai kondisi berhenti).

Langkah 1 : Selama kondisi berhenti belum dicapai, maka lakukan langkah ke-2 hingga langkah ke-9.

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

Tahap I : Umpan Maju (*feedforward*)

Langkah 3: Setiap unit *input*  $x_i$  (dari unit ke-1 hingga unit ke- $n$  pada lapisan *input*) mengirimkan sinyal *input* ke setiap *input* yang berada pada lapisan tersembunyi.

Langkah 4 : Masing-masing unit di lapisan tersembunyi (dari unit ke-1 hingga unit ke- $p$ ) dikalikan dengan bobotnya dan dijumlahkan serta ditambahkan dengan biasnya :

$$Z_{in_j} = V_{0j} + \sum_{i=1}^n X_i V_{ij} \dots \dots \dots (2.11)$$

gunakan fungsi aktivasi untuk menghitung sinyal *output*nya :

$$z_j = f(z_{in_j}) \dots \dots \dots (2.12)$$

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

Langkah 5 : Masing-masing unit *output* ( $y_k$ ,  $k= 1,2,3,\dots,m$ ) dikalikan dengan bobot dan dijumlahkan serta ditambahkan dengan biasnya

$$Y_{in_k} = W_{ok} + \sum_{j=1}^p Z_j W_{jk} \dots \dots \dots (2.13)$$

gunakan fungsi aktivasi untuk menghitung sinyal *output*nya:

$$y_k = f(y_{in_k}) \dots \dots \dots (2.14)$$

dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit *outputnya*).

Langkah 5 ini dilakukan sebanyak jumlah lapisan tersembunyi.

Tahap II : Umpan Mundur (*backward propagation*)

Langkah 6 : Masing-masing unit *output* ( $y_k$ ,  $k=1,2,3,\dots,m$ ) menerima pola target  $t_k$  sesuai dengan pola masukan (*input*) saat pelatihan dan kemudian informasi kesalahan (*error*) lapisan *output* ( $\delta_k$ ) dihitung.  $\delta_k$  dikirim ke lapisan dibawahnya dan digunakan untuk menghitung besarnya koreksi bobot dan bias ( $\Delta w_{jk}$  dan  $\Delta b_{2k}$ ) antara lapisan tersembunyi dengan lapisan *output* :

$$\delta_{2k} = (t_k - y_k) f'(y_{ink}) \dots\dots\dots(2.15)$$

$$\phi_{2jk} = \delta_k z_j \dots\dots\dots(2.16)$$

$$\beta_{2k} = \delta_k \dots\dots\dots(2.17)$$

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai  $w_{jk}$ ) :

$$\Delta w_{jk} = \alpha \phi_{2jk} \dots\dots\dots(2.18)$$

Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai  $b_{2k}$ ) :

$$\Delta b_{2k} = \alpha \beta_{2k} \dots\dots\dots(2.19)$$

Langkah 6 ini juga dilakukan sebanyak jumlah lapisan tersembunyi, yaitu menghitung informasi *error* dari suatu lapisan tersembunyi ke lapisan tersembunyi sebelumnya.

Langkah 7 : Pada setiap unit di lapisan tersembunyi (dari unit ke-1 hingga ke-p;  $i=1\dots n$ ;  $k=1\dots m$ ) dilakukan perhitungan informasi kesalahan lapisan tersembunyi ( $\delta_j$ ).  $\Delta_j$  kemudian digunakan untuk menghitung besar koreksi bobot dan bias ( $\Delta V_{ij}$  dan  $\Delta V_{jo}$ ) antara la pisan *input* dan lapisan tersembunyi.

$$\delta_{in_j} = \sum_{k=1}^m \delta_k W_{jk} \dots \dots \dots (2.20)$$

Kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi *error*.

$$\delta 1_j = \delta_{in_j} f'(z_{in_j}) \dots \dots \dots (2.21)$$

$$\phi 1_{ij} = \delta 1_j x_j \dots \dots \dots (2.22)$$

$$\beta 1_j = \delta 1_j \dots \dots \dots (2.23)$$

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai  $v_{ij}$ ) :

$$\Delta v_{ij} = \alpha \phi 1_{ij} \dots \dots \dots (2.24)$$

hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai  $b_{1j}$ ) :

$$\Delta b_{1j} = \alpha \beta 1_j \dots \dots \dots (2.25)$$

Tahap II : Pengupdatean Bobot dan Bias

Langkah 8 : Masing-masing unit *output* ( $y_k, k=1,2,3,\dots,m$ ) dilakukan *pengupdatean* bias dan bobotnya ( $j = 0,1,2,\dots,p$ ) sehingga menghasilkan bobot dan bias baru :

$$w_{jk}(\text{baru}) = \Delta w_{jk} + w_{jk}(\text{lama}) + \dots \dots \dots (2.26) \quad b_{2k}(\text{baru}) = b_{2k}(\text{lama}) + \Delta b_{2k} \dots \dots \dots (2.27)$$

Tiap-tiap unit tersembunyi ( $Z_j, j=1,2,3,\dots,p$ ) memperbaiki bias dan bobotnya ( $i=0,1,2,\dots,n$ ) :

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \dots \dots \dots (2.28)$$

$$b_{1j}(\text{baru}) = b_{1j}(\text{lama}) + \Delta b_{1j} \dots \dots \dots (2.29)$$

Langkah 9 : Uji kondisi berhenti (akhir iterasi).

Jika tes kondisi berhenti telah terpenuhi, pelatihan dapat dihentikan. Untuk menentukan kondisi berhenti terdapat 2 cara yang biasa di pakai yaitu sebagai berikut:

1. Dengan membatasi iterasi yang diinginkan dilakukan. Misal, jaringan akan dilatih sampai dengan iterasi ke-500, yang dimaksud satu iterasi adalah perulangan langkah ke-3 sampai langkah ke-8 untuk semua *training* data yang ada.
2. Dengan membatasi *error*. Pada metode *backpropagation*, dipakai metode *mean square error* untuk menghitung rata-rata *error* antara *output* yang dikehendaki pada *training* data *output* yang dihasilkan oleh jaringan. Misalnya, *error* telah mencapai 0,01 (1%), pelatihan dihentikan. Besar persen *error* tersebut tergantung kepresisian yang dibutuhkan oleh sistem yang bersangkutan.

#### 2.4.4 Pengujian Data

Setelah proses pelatihan dilakukan berdasarkan langkah-langkah algoritma *Backpropagation*, maka diperoleh nilai bobot akhir dan bias akhir yaitu dari iterasi terakhir dengan nilai *error* atau galat 0,01. Nilai bobot dan bias inilah yang akan digunakan dalam melakukan pengujian terhadap data dengan umpan maju (*feedforward*). Adapun langkah-langkah pengujian data adalah sebagai berikut :

Tahap I : Umpan Maju (*feedforward*)

1. Langkah 3 : Setiap unit *input*  $x_i$  (dari unit ke-1 hingga unit ke- $n$  pada lapisan *input*) mengirimkan sinyal *input* ke setiap *input* yang berada pada lapisan tersembunyi.
2. Langkah 4 : Masing-masing unit di lapisan tersembunyi (dari unit ke-1 hingga unit ke- $p$ ) dikalikan dengan bobotnya dan dijumlahkan serta ditambahkan dengan biasnya :

$$Z_{in_j} = V_{0j} + \sum_{i=1}^n X_i V_{ij} \dots \dots \dots (2.30)$$

gunakan fungsi aktivasi untuk menghitung sinyal *output*nya:

$$z_j = f(z_{in_j}) \dots \dots \dots (2.31)$$

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

3. Langkah 5 : Masing-masing unit *output* ( $y_k$ ,  $k= 1,2,3,\dots,m$ ) dikalikan dengan bobot dan dijumlahkan serta ditambahkan dengan biasnya

$$Y_{in_k} = W_{ok} + \sum_{j=1}^p Z_j W_{jk} \dots \dots \dots (2.32)$$

gunakan fungsi aktivasi untuk menghitung sinyal *outputnya* :

$$y_k = f(y_{\text{ink}}) \dots\dots\dots(2.33)$$

dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit *outputnya*).

Setelah pelatihan selesai dilakukan, jaringan dapat dipakai untuk pengenalan pola. Dalam hal ini, hanya umpan maju (langkah 4 dan 5) saja yang dipakai untuk menentukan keluaran jaringan.

Dalam beberapa kasus pelatihan yang dilakukan memerlukan iterasi yang banyak sehingga membuat proses pelatihan menjadi lama. Untuk mempercepat iterasi dapat dilakukan dengan parameter  $\alpha$  atau laju pemahaman. Nilai  $\alpha$  terletak antara 0 dan 1 ( $0 < \alpha \leq 1$ ). Jika harga  $\alpha$  semakin besar, maka iterasi yang dipakai semakin sedikit. Akan tetapi jika harga  $\alpha$  terlalu besar, maka akan merusak pola yang sudah benar sehingga pemahaman menjadi lambat.

Proses pelatihan yang baik dipengaruhi pada pemilihan bobot awal, karena bobot awal sangat mempengaruhi apakah jaringan mencapai titik lokal atau global, dan seberapa cepat konvergensinya. Oleh karena itu dalam standar *Backpropagation*, bobot dan bias diisi dengan bilangan acak kecil dan biasanya bobot awal diinisialisasi secara random dengan nilai antara -0,5 sampai 0,5 atau -1 sampai 1 (interval yang lainnya).

#### **2.4.5 Momentum**

Dalam *Backpropagation*, standar perubahan bobot didasarkan atas gradien yang terjadi untuk pola yang dimasukkan saat itu. Modifikasi dilakukan dengan merubah bobot yang didasarkan atasarah gradien pola terakhir dan pola sebelumnya (disebut momentum) yang dimasukkan. Jadi tidak hanya pola masukan terakhir saja yang diperhitungkan. Penambahan momentum dimaksudkan untuk menghindari

perubahan bobot yang mencolok akibat adanya data yang sangat berbeda dengan yang lain (*outlier*). Apabila beberapa data terakhir yang diberikan ke jaringan memiliki pola serupa (berarti arah gradien sudah benar), maka perubahan bobot dilakukan secara cepat. Namun apabila data terakhir yang dimasukkan memiliki pola yang berbeda dengan pola sebelumnya, maka perubahan bobot dilakukan secara lambat. Dengan penambahan momentum, bobot baru pada waktu ke  $(t+1)$  didasarkan atas bobot pada waktu  $t$  dan  $(t-1)$ . Disini harus ditambahkan dua variabel yang mencatat besarnya momentum untuk dua iterasi terakhir.

Jika  $\mu$  adalah konstanta ( $0 \leq \mu \leq 1$ ) yang menyatakan parameter momentum maka bobot baru dihitung berdasarkan persamaan sebagai berikut :

$$W_{kj}(t + 1) = W_{kj} + \alpha \delta_k Z_j + \mu(W_{kj}(t) - W_{kj}(t - 1)) \dots\dots\dots(2.34)$$

dengan,

$W_{kj}(t)$  = bobot awal pola kedua (hasil dari iterasi pola pertama).

$W_{kj}(t - 1)$  = bobot awal pada iterasi pola pertama.

dan,

$$V_{ji}(t + 1) = V_{ji}(t) + \alpha \delta_k X_j + \mu(V_{ji}(t) - V_{ji}(t - 1)) \dots\dots\dots(2.35)$$

Dengan,

$V_{ji}(t)$  = bobot awal pola kedua (hasil iterasi pola pertama).

$V_{ji}(t - 1)$  = bobot awal pada iterasi pertama.

Daftar notasi yang digunakan dalam algoritma *Backpropagation* yaitu sebagai berikut :

x	data <i>training</i> untuk <i>input</i>
	$x = (x_1, \dots, x_2, \dots, x_n)$
t	data <i>training</i> untuk <i>output</i> (target)

	$t = (t_1, \dots, t_2, \dots, t_m)$
$\alpha$	<i>learning rate</i> yaitu parameter yang mengontrol perubahan bobot selama pelatihan. Jika <i>learning rate</i> besar, jaringan semakin cepat belajar, tetapi hasilnya kurang akurat. <i>Learning rate</i> biasanya dipilih antara 0 dan 1
$X_i$	unit <i>input</i> ke – i. Untuk unit <i>input</i> , sinyal yang masuk dan keluar pada suatu unit dilambangkan dengan variabel yang sama, yaitu $X_i$
$Z_j$	<i>Hidden</i> unit ke – j. Sinyal <i>input</i> pada $Z_j$ dilambangkan dengan $Z_{inj}$ . Sinyal <i>output</i> (aktivasi) untuk $Z_j$ dilambangkan $Z_j$
$v_{oj}$	Bias untuk hidden unit ke – j
$v_{ij}$	Bobot antara unit <i>input</i> ke – i dan <i>hidden</i> unit ke – j
$Y_k$	unit <i>output</i> ke – k. Sinyal <i>input</i> ke $Y_k$ dilambangkan dengan $y_{ink}$ . Sinyal <i>output</i> (aktivasi) untuk $Y_k$ dilambangkan dengan $y_k$
$w_{ok}$	Bias untuk unit <i>output</i> ke – k.
$w_{jk}$	Bobot antara <i>hidden</i> unit ke – j dan unit <i>output</i> ke – k
$\delta_k$	Faktor koreksi <i>error</i> untuk bobot $w_{jk}$
$\delta_j$	Faktor koreksi <i>error</i> untuk bobot $v_{ij}$ .

#### 2.4.6 Aplikasi *Backpropagation* Dalam Peramalan

Peramalan adalah salah satu bidang yang paling bagus dalam mengaplikasikan metode *Backpropagation*. Secara umum, masalah peramalan dapat dinyatakan dengan sejumlah data runtun waktu (*time series*)  $X_1, X_2, \dots, X_n$ . Masalahnya adalah memperkirakan berapa nilai  $X_{n+1}$  berdasarkan  $X_1, X_2, \dots, X_n$ . Jumlah data dalam satu periode (misalnya satu tahun) pada suatu kasus dipakai sebagai jumlah masukan

dalam *Backpropagation*. Sebagai targetnya diambil data bulanan pertama setelah periode berakhir. Langkah-langkah membangun struktur jaringan untuk peramalan sebagai berikut :

#### 1. Normalisasi dan Denormalisasi

Langkah awal sebelum melakukan proses pelatihan pada jaringan yang akan digunakan untuk peramalan adalah transformasi data. Sebab-sebab utama data ditransformasi adalah agar kestabilan taburan data dicapai. Selain itu berguna untuk menyesuaikan nilai data dengan *range* fungsi aktivasi yang digunakan dalam jaringan (Siang, 2005). Dan untuk mengembalikan nilai yang sudah di normalisasi dilakukan pula denormalisasi.

#### 2. Pembagian Data

Pembagian data dilakukan dengan membagi data penelitian menjadi data pelatihan dan pengujian. Komposisi data pelatihan dan pengujian bisa dilakukan dengan *trial and error*, namun komposisi data yang sering digunakan adalah sebagai berikut :

- a. 80% untuk data pelatihan dan 20% untuk data pengujian.
- b. 70% untuk data pelatihan dan 30% untuk data pengujian.
- c. 60% untuk data pelatihan dan 40% untuk data pengujian.
- d. Dan seterusnya.

Proses pembagian data ini sangat penting, agar jaringan mendapat data pelatihan yang secukupnya. Jika data yang dibagi kurang dalam proses pelatihan maka akan menyebabkan jaringan mungkin tidak dapat mempelajari taburan data dengan baik. Sebaliknya, jika data yang dibagi terlalu banyak untuk proses pelatihan maka akan melambatkan proses pemusatan (konvergensi). Masalah *overtraining* (data pelatihan yang berlebihan) akan menyebabkan jaringan cenderung untuk menghafal data yang dimasukkan daripada menggeneralisasi.

### 3. Perancangan Arsitektur Jaringan Yang Optimum

Menentukan jumlah *neuron* masukan, *neuron* lapisan tersembunyi, jumlah lapisan tersembunyi berikutnya dan *neuron* keluaran yang akan digunakan dalam jaringan.

Penentuan ini dilakukan dengan *trial and error*.

### 4. Pemilihan Koefisien Pemahaman dan Momentum

Dalam hal ini pemilihan koefisien pemahaman dan momentum mempunyai peranan yang penting untuk struktur jaringan yang akan dibangun. Dalam membangun jaringan yang akan digunakan dalam peramalan, hasil keputusan yang kurang memuaskan dapat diperbaiki dengan penggunaan koefisien pemahaman dan momentum secara *trial and error* untuk mendapatkan nilai bobot yang paling optimum agar MSE jaringan dapat diperbaiki.

### 5. Memilih dan Menggunakan Arsitektur Jaringan yang Optimum

Tingkat keakuratan ramalannya akan dinilai setelah jaringan dibangun. Jaringan yang optimum dinilai dengan melihat nilai MSE (*Mean Square Error*) terkecil.

$$MSE = \frac{1}{n} \sum_{t=1}^n e_t^2 = \frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2 \dots\dots\dots(2.36)$$

dengan,

$n$  = bilangan ramalan.

$y_t$  = nilai aktual pada waktu  $t$ .

$\hat{y}_t$  = nilai ramalan pada waktu  $t$ .

### 6. Pemilihan Jaringan Optimum dan Penggunaanya untuk Peramalan

Jaringan dengan nilai MSE terkecil dipilih sebagai jaringan yang optimum untuk digunakan dalam peramalan.

## BAB III

### METODELOGI PENELITIAN

Pada bab ini akan dibahas mengenai metode penelitian, yaitu tahapan-tahapan yang akan dilalui dalam melakukan penelitian sebagai acuan dalam melangkah agar tidak menyimpang dari tujuan yang ingin dicapai.

#### 3.1 Obyek Penelitian

Penelitian ini dilakukan di Gaikindo, Jakarta Pusat. Obyek penelitian difokuskan pada masalah prediksi penjualan mobil Suzuki Baleno pada tahun 2004 - 2008 dengan metode jaringan syaraf tiruan *Backpropagation*, metode *Autoregressive* dan metode *Moving Average* dengan bantuan *software* Matlab.

#### 3.2 Pengumpulan Data

##### 3.2.1 Metode Pengumpulan Data

Dalam melaksanakan penelitian ini, peneliti menggunakan beberapa tahapan dan metode dalam pengumpulan data sebagai berikut :

1. Studi Pustaka

Membaca berbagai literatur yang berhubungan dengan konsep - konsep dasar jaringan syaraf tiruan.

2. Metode Observasi

Metode observasi merupakan metode pengumpulan data dengan cara melakukan pengamatan secara langsung pada perusahaan.

3. Metode Diskusi

Pada tahap ini dilakukan pengumpulan data penelitian melalui diskusi dengan pihak perusahaan tentang berbagai hal tentang penjualan otomotif di Indonesia.

### 3.2.2 Data yang Diperlukan

#### 1. Data Primer

Data primer merupakan data yang diambil secara langsung dari obyek penelitian.

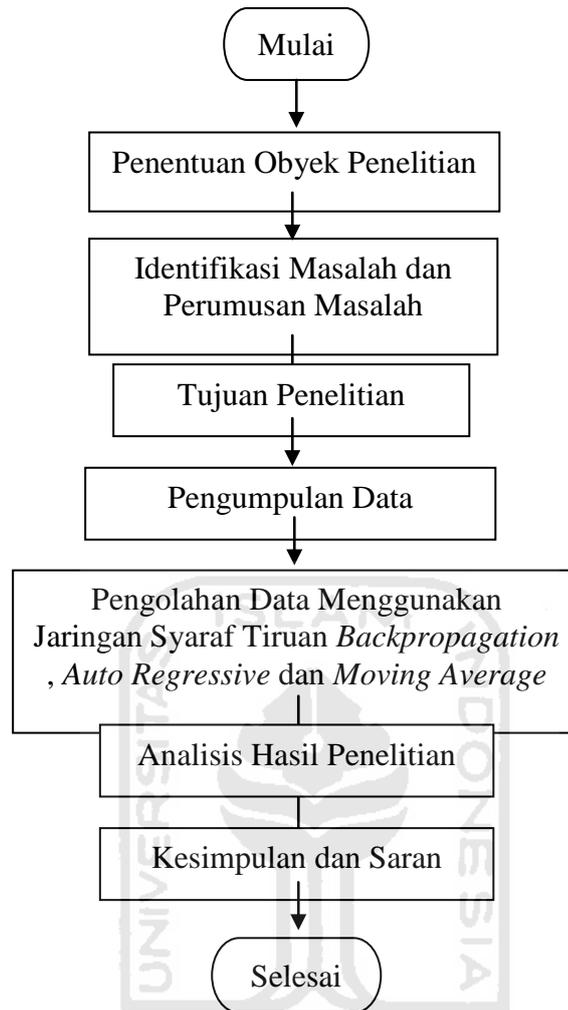
Data primer ini terdiri dari:

- a. Profil perusahaan.
- b. Struktur organisasi perusahaan.
- c. Data penjualan mobil Suzuki Baleno tahun 2004 – 2008.

#### 2. Data Sekunder

Data sekunder adalah data tambahan yang relevan dengan penelitian ini. Diantaranya diperoleh dari pustaka, *browsing* internet dan sumber lain yang dimaksudkan untuk mendapatkan landasan teori yang mengarah pada kelengkapan penjelasan topik penelitian sehingga kesimpulan yang diperoleh memiliki bobot ilmiah

### 3.3 Diagram Penelitian



Gambar 3.1 Diagram Alir Penelitian

### 3.4 Analisis Model

#### 3.4.1 Identifikasi dan Perumusan Masalah

Identifikasi masalah dilakukan untuk merumuskan masalah yang maknanya merumuskan butir-butir yang lebih atau sudah jelas dan sistematis terhadap permasalahan yang diungkapkan pada latar belakang masalah.

#### 3.4.2 Penentuan Tujuan

Langkah berikutnya yaitu menetapkan tujuan dari pembangunan konsep umum terhadap pemecahan masalah tersebut. Dari konsep ini didapatkan langkah-langkah yang harus diambil dalam memecahkan permasalahan tersebut.

#### 3.4.3 Studi Pustaka

Studi pustaka dilakukan untuk mempelajari acuan dalam literatur ilmiah yang sudah ada. Ada dua macam studi pustaka yaitu studi pustaka induktif dan deduktif.

- a. Kajian induktif adalah kajian pustaka yang bermakna untuk menjaga keaslian penelitian. Dan bermanfaat bagi peneliti untuk menjadi kekinian topik penelitian. Kajian ini diperoleh dari jurnal, *browsing*, majalah dan lain sebagainya.
- b. Kajian deduktif merupakan landasan teori yang dipakai sebagai acuan untuk memecahkan masalah penelitian. Dalam penelitian ini studi pustaka meliputi studi tentang : penjualan, *forecasting*, *autoregressive*, jaringan syaraf tiruan *backpropagation* dan *Moving Average*.

#### 3.4.4 Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data untuk lebih mengetahui mengenai sistem yang diteliti. Dari data dan informasi yang dikumpulkan akan dapat diketahui mengenai sistem saat ini.

Data-data dan informasi dapat diperoleh melalui :

1. Metode wawancara (*interview*).
2. Observasi/pengamatan secara langsung di Gaikindo.
3. Studi dokumentasi.

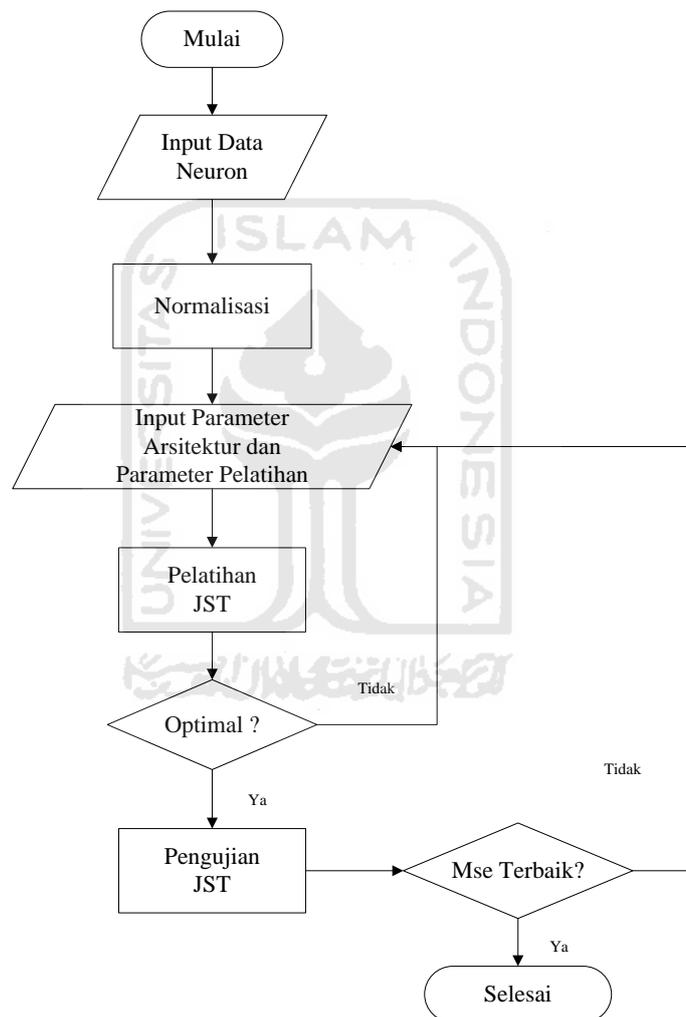
#### 3.4.5 Analisis Kebutuhan Sistem

Pada sistem yang akan dibangun kali ini adalah sistem yang mudah dalam pembuatan serta pelatihan data dalam sistem JST serta metode *Autoregressive* dan metode *Moving Average*. Dengan perancangan yang struktural dengan *flowchart* (diagram alir) untuk mengembangkan dan menggambarkan langkah-langkah algoritma dalam menentukan proses perhitungan pelatihan dan pengujian pada JST serta identifikasi model pada *Autoregressive*. Dengan menggunakan metode ini diharapkan hasil analisis menjadi lebih terstruktur sehingga dapat dijelaskan dengan baik serta memungkinkan proses-proses dapat terdefinisikan dengan baik dalam memprediksi penjualan mobil. Untuk melakukan pelatihan dan prediksi penjualan mobil digunakan simulasi jaringan syaraf tiruan metode *Backpropagation*, metode *Autoregressive* dan metode *Moving Average*

### 3.4.6 Analisis Fungsionalitas Sistem

#### 3.4.6.1 Fungsionalitas Sistem Jaringan Syaraf Tiruan Backpropagation

Pada penelitian ini dibangun sistem penghitungan variabel terbatas pada prediksi penjualan mobil dengan *input* data pelatihan dan data pengujian dimana *outputnya* berupa data hasil perkiraan. Diagram dari sistem yang dibuat tampak pada gambar 3.2.



Gambar 3.2. Diagram Sistem Penghitungan *Backpropagation* Prediksi Penjualan Mobil

Dari Gambar 3.2 dapat dijelaskan tahap penghitungan JST *Backpropagation* sebagai berikut :

### 1. Pemilihan Data Inputan

Data pelatihan yang digunakan pada proses pelatihan metode *Backpropagation* menggunakan data harian dari penjualan mobil Suzuki Baleno yaitu mulai dari tahun 2004 sampai tahun 2008. Data pelatihan dibagi ke dalam dua bagian, yaitu :

1. Data masukan (*input*), berupa data runtun waktu (*time series*)  $x_1, x_2, \dots, x_n$  yang terdiri dari jumlah penjualan mobil harian yang sama dalam kurun waktu sesuai dengan arsitektur jaringan.
2. Data target, yaitu data penjualan mobil pada satu hari setelah kurun waktu data masukan ( $x_{n+1}$ ).

Data pengujian yang dipakai memiliki karakteristik yang sama dengan data pada tahap pelatihan, dengan menginputkan yang akan diprediksi. Data berdasarkan prediksi tersebut merupakan data target yang akan disimulasikan pada variabel yang berisi pola jaringan hasil pelatihan untuk menghasilkan nilai *output* yang merupakan hasil prediksi penjualan mobil.

Pada penelitian ini, data penjualan mobil pada tanggal 1 Januari 2004 sampai 31 Desember 2008 dibagi menjadi data pelatihan (*training*) dan pengujian (*testing*). Persentase data pelatihan adalah 70 % dari data, sejumlah 739 data (1 Januari 2004 sampai 31 Desember 2006) dan 30% untuk data pengujian dan validasi, sejumlah 497 data (1 Januari 2007 sampai 30 November 2008).

### 2. Normalisasi Data

Sebelum data dianalisis dilakukan normalisasi data, bertujuan untuk meningkatkan keakurasian dari hasil *output* dan memfasilitasi proses pelatihan (*learning*) dari

jaringan syaraf tiruan. Ini merupakan operasi kritis sebab jaringan syaraf adalah pencocokan pola, sehingga ketika pola data telah ditunjukkan maka akan mengubah perilaku jaringan syaraf tiruan.

Normalisasi data berguna untuk menghilangkan *noise* pada data. Metode normalisasi yang digunakan *mapminmax* pada *range* [-1 1].

Syntax *mapminmax* :

$$Y = \frac{(y_{\max} - y_{\min}) * (X - X_{\min})}{(X_{\max} - X_{\min})} + Y_{\min} \dots\dots\dots(3.1)$$

Y adalah nilai diantara -1 dan 1.  $Y_{\max} = 1$ ,  $y_{\min} = -1$  dan  $x_{\max}$  serta  $x_{\min}$  bergantung pada nilai maksimum dan minimum data *inputnya*. Proses transformasi balik ke data awal dilakukan dengan menggunakan invers dari persamaan (3.1) dengan mencari variabel X. (Demuth, 2009). Dan pada Matlab mendapatkan nilai sebenarnya dari keluaran dilakukan proses *reverse* normalisasi data atau denormalisasi.

### 3. Merancang Model Jaringan Syaraf Tiruan

Penentuan model jaringan syaraf tiruan yang akan dipakai harus ditentukan dengan model manakah yang paling mudah beradaptasi dan yang paling sering dipakai untuk membandingkan setiap kemungkinan. Langkah-langkah membangun model jaringan syaraf tiruan untuk prediksi penjualan adalah :

- a. Jumlah *neuron input* : jumlah ini mudah ditentukan karena telah dilakukan pada tahap sebelumnya
- b. Jumlah *hidden layer* : jumlah dari *hidden layer* bergantung pada jumlah *input neuron* dan properti data.
- c. Jumlah *hidden neuron* : teknik yang umum digunakan dalam menentukan jumlah *hidden neuron* adalah percobaan. Hal penting yang harus diperhatikan

adalah selalu memilih jaringan dengan performa terbaik dan jumlah *hidden neuron* yang sedikit.

- d. Fungsi aktivasi : fungsi aktivasi digunakan untuk menentukan *output* dari *neuron* yang diproses. Teknik yang dipakai untuk menentukan fungsi aktivasi terbaik adalah percobaan. Kriteria penentu yang dipakai untuk menentukan fungsi aktivasi adalah kemampuannya untuk mempercepat fase pembelajaran dan meningkatkan keakuratan dari jaringan syaraf tiruan. Fungsi yang digunakan *sigmoid biner (logsig)*, *sigmoid bipolar (tansig)* dan fungsi identitas atau *purelin*. Penentuan jumlah *hidden layer* dan jumlah *neuron* ditentukan secara *trial and error* sampai dicapai *error* minimum.
- e. Penempatan perintah pada program pelatihan untuk menyimpan MSE pelatihan, MSE pengujian dan *epoch* apabila tercapai MSE pengujian yang lebih minimum dari sebelumnya. Modifikasi ini diharapkan untuk membantu mempercepat tercapainya minimum *error* dari *epoch* maksimum yang dijalankan. Metode pelatihan yang diterapkan pada JST *Backpropagation* adalah *gradien descent momentum*.

#### 4. Pelatihan (*training*)

Pelatihan pada jaringan syaraf tiruan meliputi proses iteratif dari data *input* prediksi sehingga jaringan tersebut sesuai dan dapat dipakai untuk prediksi. Tujuan dari pelatihan adalah meminimalkan *error* yang mengindikasikan bahwa model jaringan syaraf tiruan sudah sesuai dengan *input*. Fungsi *error* yang dipakai pada model jaringan syaraf tiruan adalah MSE. Fungsi *error* ini digunakan untuk penggantian bobot dari jaringan. Penggantian bobot dari jaringan dilakukan sehingga *output* dari jaringan dapat sesuai dengan target. Dari pelatihan dapat

dicari nilai bobot yang menghubungkan semua *neuron* sehingga nilai *error* yang dihasilkan pada jaringan syaraf menurun.

#### 5. Perhitungan *error*

Perhitungan *error* bertujuan untuk pengukuran keakurasian. Ada tiga macam perhitungan *error* yang sering dipakai, yaitu *Mean Square Error* (MSE), *Mean Absolute Error* (MAE) dan *Mean Absolute Percentage Error* (MAPE).

MSE merupakan salah satu dari beberapa macam *error* yang sering dipakai. MSE merupakan rata-rata kuadrat dari selisih antara *output* jaringan dengan target *output*. Tujuan utama adalah *output* jaringan dengan target *output*. Tujuan utama adalah memperoleh nilai *error* ini sekecil-kecilnya dengan secara iteratif mengganti nilai bobot yang terhubung pada semua *neuron* pada jaringan syaraf tiruan. Untuk mengetahui seberapa banyak bobot harus diganti, setiap iterasi memerlukan perhitungan *error* yang berasosiasi dengan setiap neuron pada output dan *hidden layer*. Rumus dari MSE yang akan digunakan pada metode jaringan syaraf adalah sebagai berikut :

$$MSE = \frac{e_1^2 + e_2^2 + e_3^2 + \dots + e_n^2}{n} \dots\dots\dots(3.2)$$

dimana :

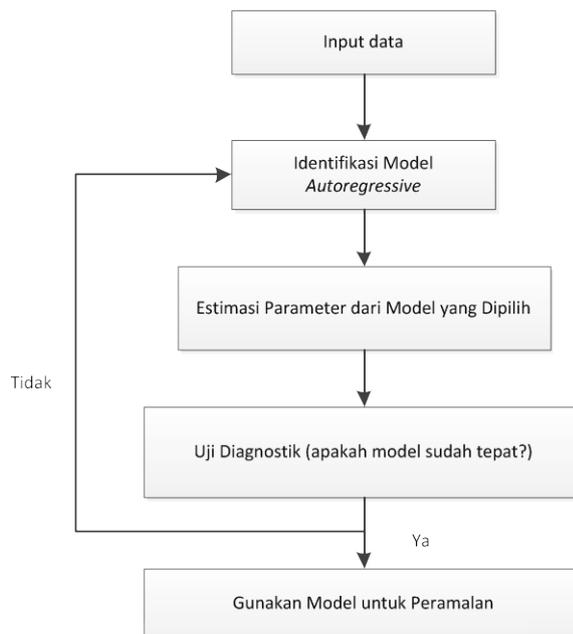
mse = *mean square error*

e = *error* dari *neuron* pada metode jaringan syaraf

n = jumlah *error* (target) pada metode jaringan syaraf

#### 3.4.6.2 Fungsionalitas Metode *Autoregressive*

Langkah-langkah membentuk model peramalan penjualan dengan metode *Autoregressive* :



Gambar 3.3 *Flowchart Perhitungan Autoregressive*

#### 1. Identifikasi model

Dalam tahap ini akan dicari model yang dianggap paling sesuai dengan data. Diawali dengan membuat plot data asli, membuat trend analisisnya,

#### 2. Estimasi Parameter

Tahap selanjutnya setelah model awal teridentifikasi sementara adalah mencari estimasi terbaik atau paling efisien untuk parameter dalam model itu. Seperti parameter-parameter *Autoregressive*, musiman dan tidak musiman harus ditetapkan dengan cara yang terbaik. Terdapat dua cara yang mendasar untuk mendapatkan parameter-parameter terbaik dalam mencocokkan deret berkala yang sedang dimodelkan (Makridakis,1995), yaitu sebagai berikut :

1. Dengan cara mencoba-coba menguji beberapa nilai yang berbeda dan memilih satu nilai tersebut (sekumpulan nilai, apabila terdapat lebih dari satu parameter yang akan ditaksir) yang meminimumkan jumlah kuadrat nilai sisa (*sum of squared residuals*).

2. Perbaiki secara iteratif memilih taksiran awal dan kemudian membiarkan program komputer memperhalus penaksiran tersebut secara iteratif.

3. Uji Diagnostik

Uji diagnostik yaitu memeriksa atau menguji apakah model telah dispesifikasi secara benar atau apakah telah dipilih ordo yang benar.

4. Peramalan

Apabila model memadai maka model tersebut dapat digunakan untuk melakukan peramalan. Sebaliknya, apabila model belum memadai maka harus ditetapkan model

yang lain.

### 3.4.7 Hasil Analisis

Setelah data selesai di olah, pada tahap ini hasil pengolahan data dan implementasi di analisa terlebih dahulu sebelum ditarik sebuah kesimpulan dan hasil penelitian.

### 3.4.8 Kesimpulan dan Saran

Kesimpulan merupakan pernyataan singkat, jelas dan tepat mengenai apa yang diperoleh, dibutuhkan dan dijabarkan dari tujuan penelitian. Sedangkan saran diberikan agar penelitian dapat berguna untuk penelitian selanjutnya.

## BAB IV

### PENGUMPULAN DAN PENGOLAHAN DATA

#### 4.1 Data Umum Tempat Objek Penelitian

##### 4.1.1 Sejarah Berdirinya Gaikindo

Mobil diperdagangkan dan digunakan di Indonesia jauh sebelum Perang Dunia II dan setelah kemerdekaan pada tahun 1945 penjualan mobil mulai berkembang. Pada tahun 1960-an, pemerintah diatur secara resmi industri berkembang dan pemain merasa perlu untuk membentuk organisasi untuk mewakili kekhawatiran dan kebutuhan mereka. Berikut tahap-tahap berkembangnya Gaikindo :

1. Tahun 1969 didirikan *Asosiasi Otomotif Indonesia Sole Agent* (GAKINDO) dan *Association of Car Assemblers* (GAM) didirikan.
2. Tahun 1975 kedua organisasi bergabung ke dalam *Association of Indonesian Sole Agents and Car Assemblers* (GAAKINDO).
3. Tahun 1985 GAAKINDO berevolusi menjadi sebuah organisasi baru yang disebut *Asosiasi Industri Kendaraan Bermotor Indonesia* (GAIKINDO). Pada saat itu GAIKINDO memiliki 37 anggota, yang terdiri dari agen tunggal, perakitan mobil, dan produsen komponen utama.
4. Tahun 1998 terjadi krisis keuangan dan cukup memperlambat perkembangan industri bermotor di Indonesia.
5. Tahun 2004 Gaikindo mengubah logo untuk menunjukkan bagaimana organisasi telah berkembang selama bertahun-tahun. Simbol sebelumnya, yang menampilkan

roda kemudi di dalam segi lima, cocok ketika industri baru saja membintangi untuk mengembangkan dan dilindungi oleh pemerintah. Ikon baru, menunjukkan sebuah mobil meluncur ke depan dan melaju ke masa depan, cocok lebih baik dengan kemerdekaan Gaikindo dan dinamika. Garis melengkung pada *fleksibilitas chow* kendaraan dan nada warna-warni melambangkan komitmen Gaikindo untuk mendukung lingkungan yang bersih dan sehat.



Gambar 4.1 Logo Lama dan Baru Gaikindo

6. Tahun 2005 GAIKINDO dikembangkan maskot untuk menunjukkan komitmennya terhadap lingkungan dan sebagai sarana untuk mendidik masyarakat. Maskot itu, menjadi simbol tokoh yang ramah dan keluar. Ini menggambarkan suatu makhluk dibebaskan yang tidak akan sujud kepada suatu kepentingan tertentu. GAIKINDO ingin sekali untuk berhubungan dengan masyarakat dengan cara yang menyenangkan dengan maskot baru. Indonesia memulai industri otomotif untuk menerapkan teknologi *Euro-2 Standard*, mencari dukungan dari peraturan oleh Menteri Lingkungan Hidup. Masih di tahun 2005, industri otomotif nasional membuat pemulihan setelah tujuh tahun sejak krisis pada tahun 1998 dengan 532.000 unit pada tahun 2005 penjualan, rekor baru pada waktu itu.
7. Tahun 2005 GAIKINDO menjadi anggota tetap Organisasi Internasional Produsen Kendaraan Bermotor, yang juga dikenal sebagai Organisasi *Internationale des Constructeurs d'Automobiles* (OICA) yang Berkantor pusat di Paris. Organisasi otomotif dunia mendukung Indonesia *International Motor Show Otomotif* diadakan

oleh GAIKINDO.

8. Tahun 2008 Penjualan memecahkan rekor baru 603.774 unit di tengah gejolak ekonomi global. Ekspor juga memecahkan rekor baru dengan 100.000 unit.

#### **4.1.2 Visi dan Misi**

Visi dan misi Gaikindo adalah sebagai berikut :

1. Visi adalah untuk menjadi sebuah organisasi otomotif kelas dunia, profesional dan independen, yang memainkan peran aktif sebagai mitra sejajar pemerintah dalam mengembangkan industri yang akan mahir bersaing di pasar internasional.
2. Misi adalah untuk mengubah industri otomotif Indonesia dan meningkatkan daya saing di pasar global.

#### **4.1.3 Strategi Gaikindo**

Gaikindo memiliki beberapa strategi dalam organisasinya untuk mendukung perkembangan industri otomotif di Indonesia yaitu :

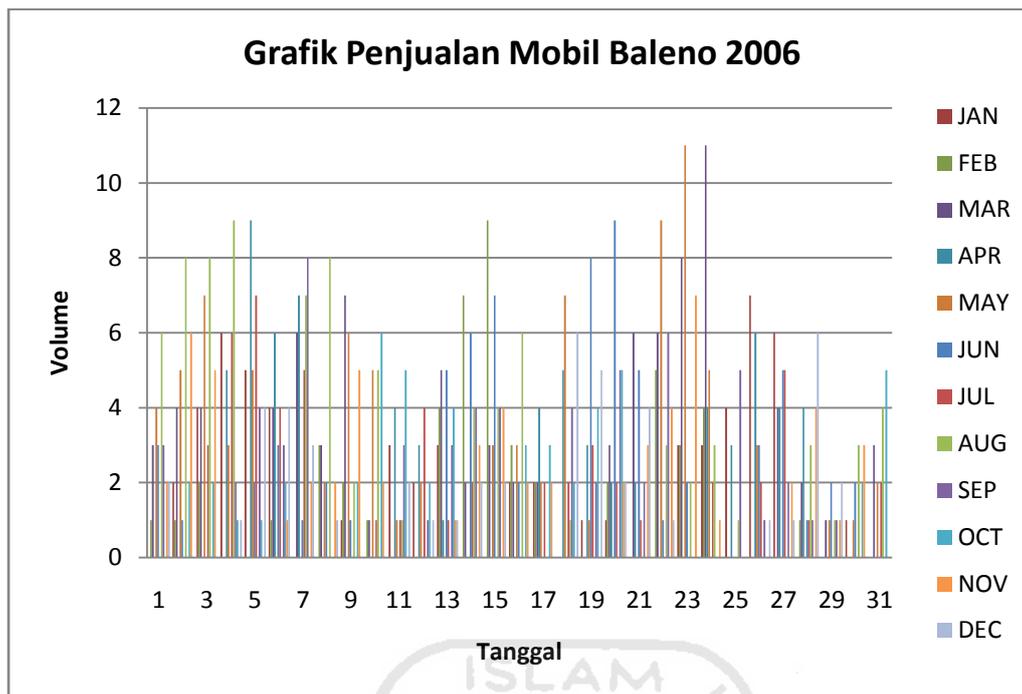
1. Untuk mengatur dan mendukung acara internasional untuk mencari perhatian global dan mempromosikan kesempatan perdagangan dan investasi di dalam negeri
2. Untuk mendorong, mempromosikan dan melindungi kepentingan industri otomotif di Indonesia.
3. Untuk bertindak sebagai sarana komunikasi antara para anggotanya untuk penguatan dan promosi kepentingan bersama mereka.
4. Untuk mengumpulkan, mengkompilasi dan menyusun informasi mengenai masalah yang mempengaruhi kepentingan industri otomotif pada umumnya dan anggota khususnya dan kemudian mendistribusikan informasi tersebut dalam bentuk laporan atau statistik untuk anggota

## 4.2 Pengumpulan Data

Prediksi penjualan mobil adalah suatu usaha untuk mengetahui penjualan di waktu mendatang berdasarkan data di masa lalu. Pengumpulan data yang dilakukan adalah data harian penjualan mobil Suzuki Baleno dari tahun 2004 sampai tahun 2008. Berikut merupakan salah satu tabel data penjualan harian dalam satu tahun.

Tabel 4.1 Data Penjualan Harian Mobil Baleno Tahun 2004

TGL.	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
1	-	-	6	2	-	2	4	-	6	2	6	8
2	3	-	2	2	-	4	6	7	12	-	2	10
3	-	5	4	-	5	-	-	6	5	-	8	13
4	-	6	7	-	3	0	-	8	-	8	4	-
5	4	5	12	2	7	-	3	4	-	4	5	-
6	5	4	-	1	9	-	4	3	8	11	-	7
7	4	-	-	9	8	4	9	-	19	9	-	6
8	2	-	4	10	-	3	5	-	8	8	5	9
9	2	7	13	-	-	5	8	5	6	6	-	11
10	-	6	7	-	6	4	-	2	7	-	9	4
11	-	3	7	-	7	11	-	9	-	-	4	-
12	2	9	9	3	5	-	4	3	-	14	6	-
13	2	8	-	4	9	-	8	2	-	9	-	4
14	6	-	-	10	6	7	6	-	4	12	-	8
15	1	-	3	3	-	2	6	-	7	6	-	9
16	4	8	2	7	-	6	5	8	16	-	-	2
17	-	3	11	-	2	-	-	10	8	-	2	6
18	-	4	4	-	1	2	-	5	-	9	4	-
19	5	5	6	9	2	-	12	4	-	7	2	-
20	7	2	-	3	-	-	8	7	5	3	-	10
21	1	-	-	6	7	5	7	-	7	6	-	6
22	-	-	-	2	6	2	5	-	14	2	4	5
23	2	-	3	8	5	4	3	4	10	-	6	7
24	-	2	4	-	4	5	-	8	8	-	8	5
25	-	6	2	-	10	8	-	6	-	2	10	-
26	4	4	8	11	7	-	2	6	-	1	7	-
27	5	8	-	4	4	-	2	9	12	5	-	12
28	3	-	-	7	9	3	6	-	11	4	-	5
29	7	-	6	0	5	6	3	-	9	2	6	5
30	3	-	3	3	3	5	4	8	8	-	5	6
31	-	-	5	-	8	-	-	9	-	-	-	4
	<b>72</b>	<b>95</b>	<b>128</b>	<b>106</b>	<b>138</b>	<b>88</b>	<b>120</b>	<b>133</b>	<b>190</b>	<b>130</b>	<b>110</b>	<b>162</b>



Gambar 4.1 Grafik Penjualan Mobil Baleno 2006

#### 4.3 Perancangan JST *Backpropagation*

Perencanaan sistem prediksi penjualan mobil Suzuki Baleno menggunakan aplikasi jaringan syaraf tiruan dengan metode *Backpropagation* terdiri dari langkah-langkah berikut :

##### 4. Tahap umpan maju (*feedforward*)

Pola masukan dihitung maju mulai dari layar masukan hingga layar keluaran menggunakan fungsi aktivasi yang ditentukan.

##### 5. Tahap umpan mundur (*Backpropagation*)

Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan yang terjadi itu dipropagasi mundur. Dimulai dari garis yang berhubungan langsung dengan unit-unit di layar keluaran

6. Tahap pengupdatean bobot dan bias.

Modifikasi bobot untuk menurunkan kesalahan yang terjadi.

Sebelum melakukan pelatihan diperlukan tahapan-tahapan pengaturan JST

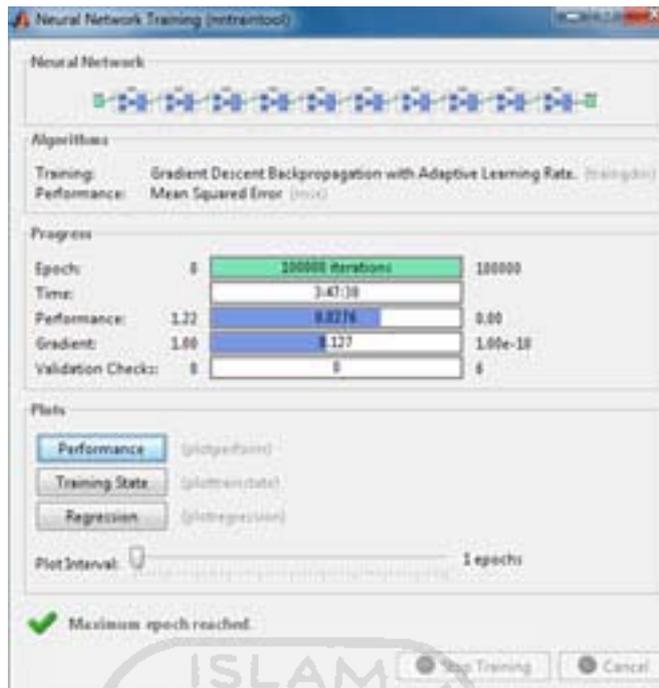
*Backpropagation* yaitu :

1. Arsitektur jaringan syaraf : *jaringan syaraf Backpropagation*
2. Jumlah sel *input* : *3 neuron*
3. Jumlah lapisan tersembunyi : *90-80-60-70-60-50-40-30-20-10-5 neuron*
4. Jumlah *neuron output* : *1 neuron*
5. Maksimum *epoch* : *10000 iterasi*
6. Fungsi aktivasi : *Tansig, tansig, tansig, tansig, logsig, tansig, logsig, tansig, logsig, logsig, purelin.*

Pada pengolahan data JST *Backpropagation*, nilai *learning rate* ( $\alpha$ ) dan momentum ( $\mu$ ) diberi nilai yang berbeda-beda untuk mendapatkan akurasi peramalan yang terbaik.

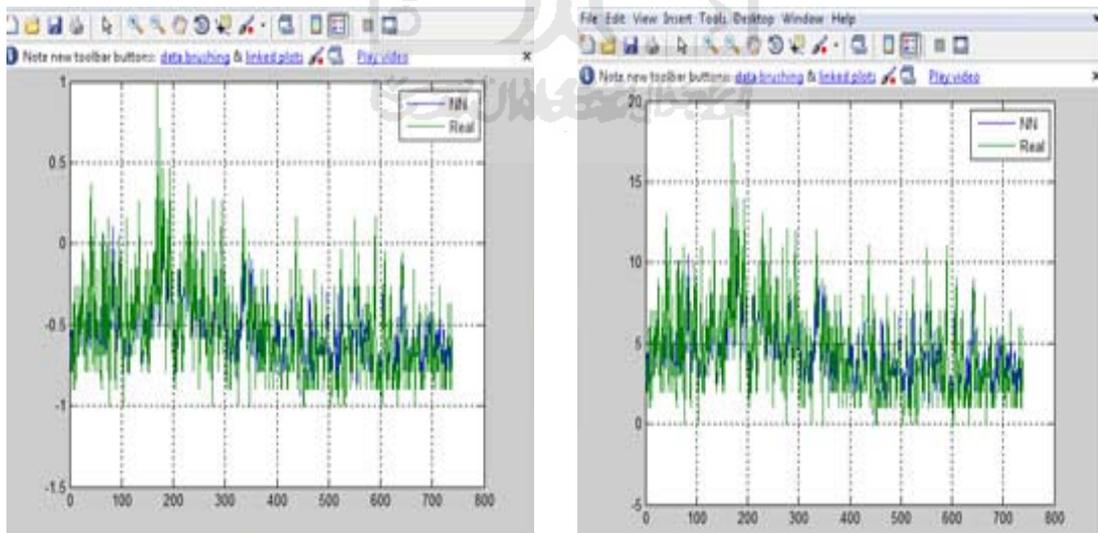
1. *Learning rate* = 0.2 dan momentum = 0.48
2. *Learning rate* = 0.21 dan momentum = 0.49
3. *Learning rate* = 0.18 dan momentum = 0.48





Gambar 4.5 Tampilan *Neural Network Training*

Dari gambar di atas dapat dilihat maksimum *epoch* 100000 lebih duluan tercapai. Setelah jaringan pelatihan selesai buat plot gambar untuk mengetahui apakah jaringan pelatihan dapat mengenali pola data yang ada.



Gambar 4.6 Grafik Pelatihan Jaringan di Normalisasi (kiri) dan Grafik Pelatihan Denormalisasi (kanan)

Dari gambar di atas dapat dilihat bahwa jaringan syaraf tiruan dapat belajar mengenali pola, dimana grafik jaringan mengikuti gerak laju data yang ada.

3. Kembali ke jendela utama Matlab, masukkan perintah untuk mengetahui hasil *error* pelatihan dan MSE pelatihan



```

MATLAB 7.8.0 (R2009a)
File Edit Debug Parallel Desktop Window Help
Current Directory: C:\Users\USER\Documents\MATLAB
Shortcuts How to Add What's New

Command Window
New to MATLAB? Watch the Video, see Demos, or read Getting Started

>> error=basinorm-y

error =

Column 1 through 15
-0.2114 0.1048 0.0004 -0.1583 -0.0018 0.0004 +0.0537 0.0003 -0.0020 -0.0004 0.2097 0.1057 -0.2118 -0.0018 -0.0015

Column 16 through 30
-0.0022 -0.0016 0.1570 -0.0021 -0.0015 0.3148 -0.0014 -0.0019 -0.0014 -0.0010 -0.0003 0.0001 -0.0006 -0.0010 -0.0014

Column 31 through 45
0.2408 0.5252 -0.0016 -0.0014 -0.0012 0.0000 0.2455 -0.0007 -0.0538 0.0546 -0.0014 0.1056 -0.0018 -0.0017 0.3145

Column 46 through 60
-0.0013 -0.0004 0.1589 -0.0018 -0.0017 -0.1756 -0.0015 -0.0020 -0.0008 -0.1045 -0.0019 -0.0004 -0.0010 -0.0020 -0.0010

Column 61 through 75
0.2082 -0.0010 -0.0009 -0.0016 -0.0015 -0.0010 -0.0005 -0.0007 -0.0022 -0.0004 0.0433 -0.0021 -0.0016 -0.0011 0.1734

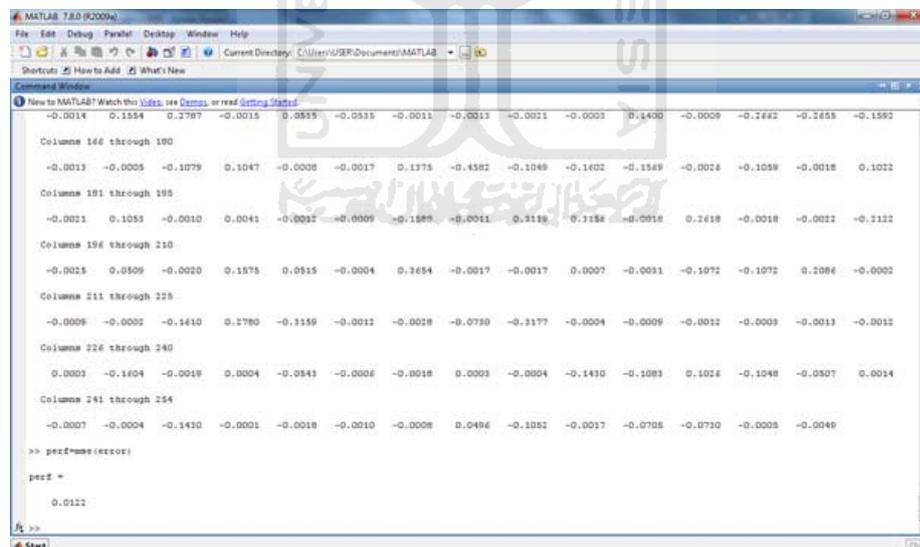
Column 76 through 90
-0.0017 -0.0013 -0.0014 -0.0017 -0.0007 -0.0007 -0.0016 -0.0014 -0.0011 -0.0017 -0.0020 -0.0016 -0.0011 -0.0026 -0.2143

Column 91 through 105
-0.0005 -0.0014 -0.0025 -0.0006 -0.0009 -0.0005 -0.0009 -0.0009 -0.0013 -0.0010 -0.0007 -0.0539 -0.2117 0.0003 -0.0009

Column 106 through 119

```

Gambar 4.7 Tampilan Jendela Hasil *Error* Pelatihan



```

MATLAB 7.8.0 (R2009a)
File Edit Debug Parallel Desktop Window Help
Current Directory: C:\Users\USER\Documents\MATLAB
Shortcuts How to Add What's New

Command Window
New to MATLAB? Watch the Video, see Demos, or read Getting Started

-0.0014 0.1554 0.2787 -0.0015 0.0515 -0.0535 -0.0011 -0.0013 -0.0021 -0.0003 0.1400 -0.0009 -0.2462 -0.2655 -0.1592

Column 166 through 180
-0.0013 -0.0005 -0.1079 0.1047 -0.0008 -0.0017 0.1375 -0.6582 -0.1049 -0.1602 -0.1549 -0.0026 -0.1039 -0.0018 0.1022

Column 181 through 195
-0.0021 0.1053 -0.0010 0.0041 -0.0032 -0.0009 -0.1589 -0.0011 0.3138 0.3156 -0.0018 0.2418 -0.0018 -0.0022 -0.2122

Column 196 through 210
-0.0025 0.0509 -0.0020 0.1575 0.0515 -0.0004 0.3654 -0.0017 -0.0017 0.0007 -0.0031 -0.1072 -0.1072 0.2086 -0.0002

Column 211 through 225
-0.0005 -0.0005 -0.1610 0.2780 -0.3359 -0.0012 -0.0028 -0.0750 -0.3177 -0.0004 -0.0009 -0.0012 -0.0003 -0.0013 -0.0012

Column 226 through 240
0.0002 -0.1604 -0.0019 0.0004 -0.0543 -0.0006 -0.0018 0.0003 -0.0004 -0.1430 -0.1083 0.1024 -0.1048 -0.0507 0.0014

Column 241 through 254
-0.0007 -0.0004 -0.1430 -0.0001 -0.0018 -0.0010 -0.0008 0.0496 -0.1052 -0.0017 -0.0705 -0.0730 -0.0005 -0.0049

>> perf=ms(error)

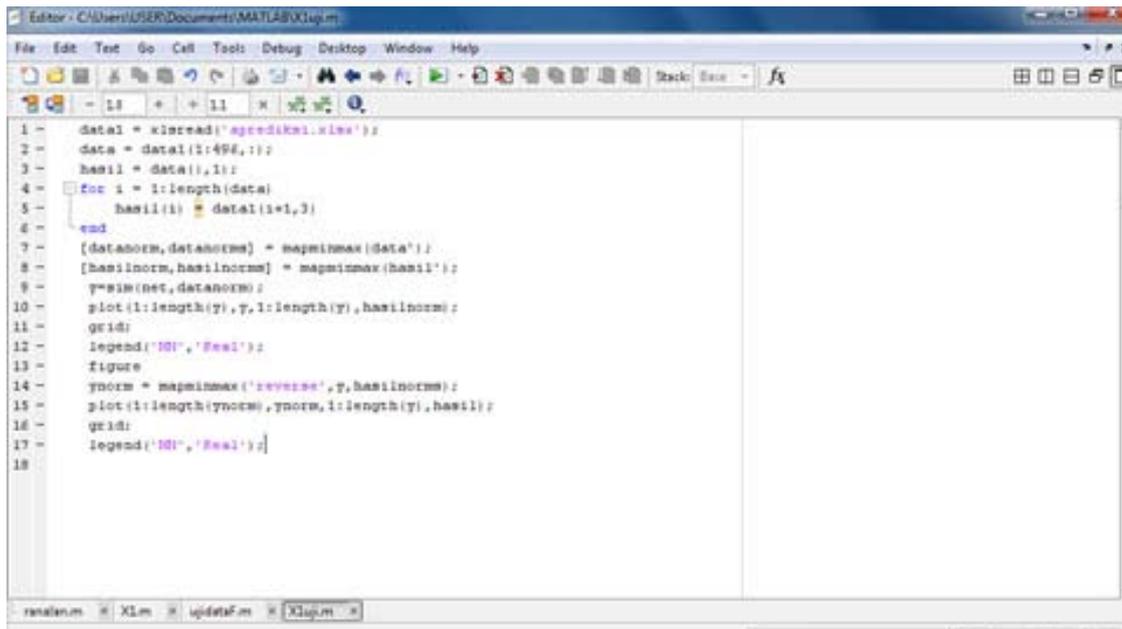
perf =

0.0122

```

Gambar 4.8 Jendela Hasil MSE Pelatihan

Dari pelatihan didapat nilai MSE 0.0122, maka jaringan ini dapat dilanjutkan ke tahap pengujian. Masukkan perintah pada M-File Editor untuk menguji struktur jaringan yang sudah dilatih.

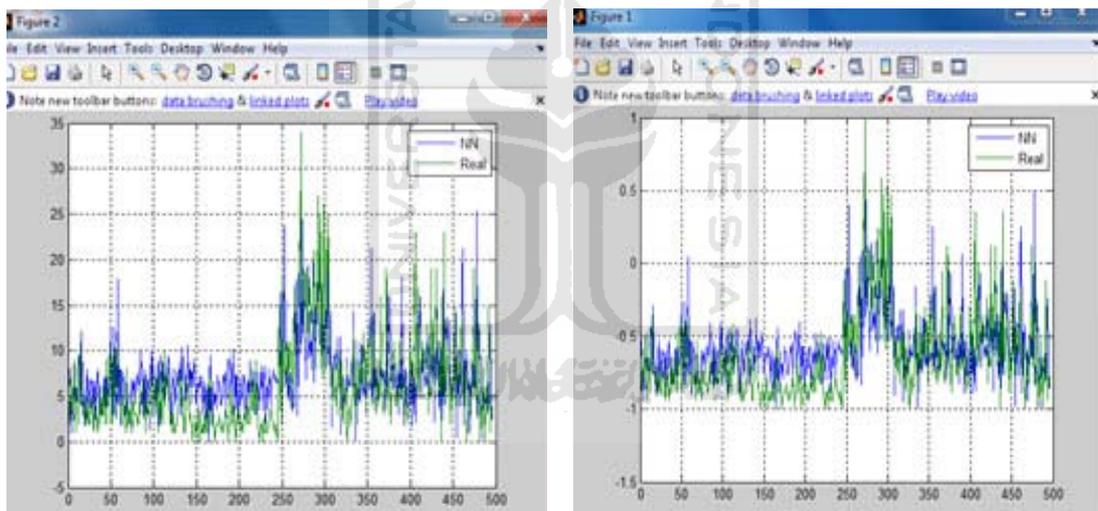


```

1 - data1 = xlsread('agredakml.xlsx');
2 - data = data1(1:494,:);
3 - hasil = data(:,1);
4 - for i = 1:length(data)
5 -     hasil(i) = data1(i=1,3)
6 - end
7 - [datanorm, datanorme] = mapminmax(data');
8 - [hasilnorm, hasilnorme] = mapminmax(hasil');
9 - y = fitnet(datanorme);
10 - plot(1:length(y), y, 1:length(y), hasilnorm);
11 - grid;
12 - legend('NN', 'Real');
13 - figure
14 - ynorm = mapminmax('reverse', y, hasilnorme);
15 - plot(1:length(ynorm), ynorm, 1:length(y), hasil);
16 - grid;
17 - legend('NN', 'Real');
18

```

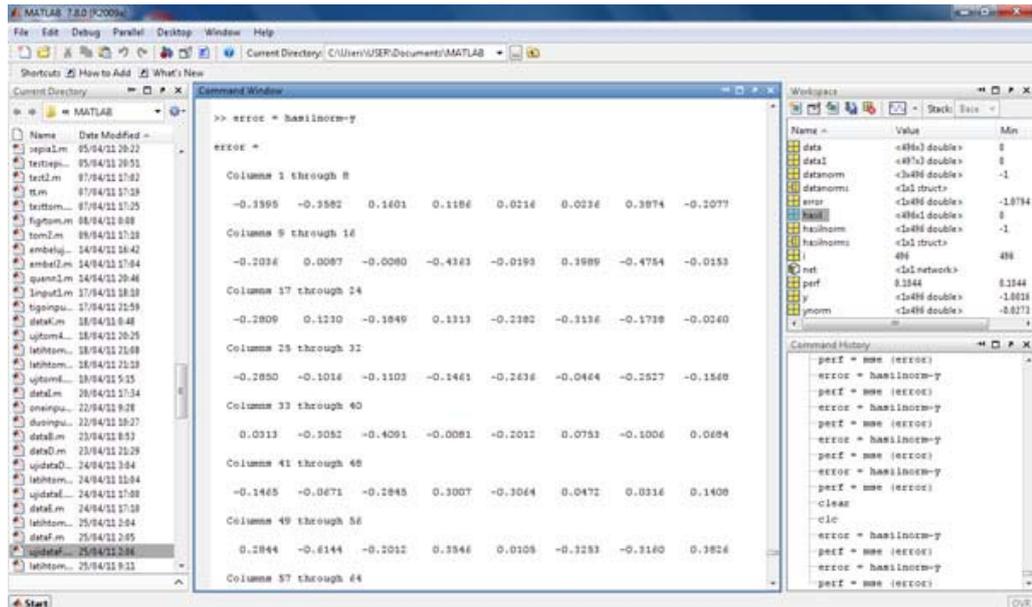
Gambar 4.9 Tampilan Jendela Editor Pengujian



Gambar 4.10 Grafik Hasil Pengujian Normalisasi (kiri) dan Grafik Hasil Pengujian Denormalisasi (kanan)

Analisa dari grafik bahwa jaringan syaraf tiruan masih dapat mengikuti pola data yang ada. Dan dapat dilakukan peramalan.

4. Masukkan perintah kembali pada jendela *Command Window* untuk menghitung *error* dan nilai MSE yang dicapai.



Gambar 4.11 Tampilan jendela Hasil *Error* Pengujian

Nilai *error* dapat diperoleh dengan memasukan perintah pada jendela *Command Window* yaitu :

$$\text{error} = \text{hasilnorm} - y \dots\dots\dots(4.1)$$

dimana *hasilnorm* adalah nilai target yang sudah di normalisasi dan *y* adalah hasil prediksi validasi (simulasi).

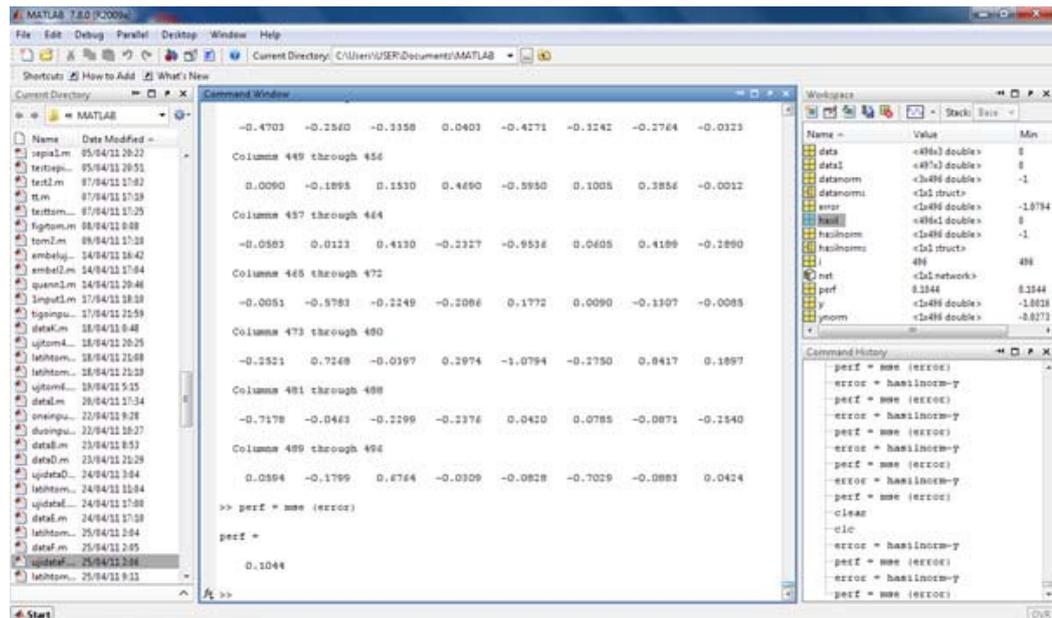
Contoh pada *software* Matlab :

```
p = [-10 -5 0 5 10];
t = [0 0 1 1 1];
y = sim(net,p)
e = t-y
perf = mse(e) (Matlab.2007)
```

Contoh perhitungan nilai *error* pada  $\alpha$  (0.18) dan  $\mu$  (0.48) pada tahap pelatihan data pertama :

$$\begin{aligned} \text{Error (data1)} &= \text{hasilnorm} - y \\ &= (-0.88235294) - (-0.52284) \\ &= -0.35952 \end{aligned}$$

Dan seterusnya sampai data terakhir. Hasil perhitungan *error* terlampir.



Gambar 4.12 Jendela Hasil MSE Pengujian

Untuk mengetahui nilai MSE pengolahan data dengan *software* Matlab dapat diketik perintah pada *Command Window* yaitu :

`perf = mse (error)`

atau dapat juga dihitung manual dengan rumur MSE yaitu :

$$MSE = \frac{e_1^2 + e_2^2 + e_3^2 + \dots + e_n^2}{n} \quad (4.2)$$

dimana :

mse = *mean square error*

e = *error* dari *neuron* pada metode jaringan syaraf

n = jumlah *error* (target) pada metode jaringan syaraf

Contoh perhitungan MSE untuk  $\alpha$  (0.2) dan  $\mu$  (0.48) pengujian sebagai berikut :

$$\begin{aligned}
 MSE &= ((-0.4606)^2 + (-0.07203)^2 + (-0.133106)^2 + \dots + (-0.13702)^2) / 496 \\
 &= 0.1250
 \end{aligned}$$

Nilai MSE pada setiap pelatihan dan pengujian terlampir.

Setelah pengolahan data selesai diperoleh hasil prediksi berdasarkan nilai *learning rate* ( $\alpha$ ) dan momentum ( $\mu$ ) yang telah di tentukan. Berikut hasilnya seperti di bawah ini :

Tabel 4.2 Hasil Prediksi Mobil Suzuki Baleno

Data	$\alpha(0.2), \mu(0.48)$	$\alpha(0.21), \mu(0.49)$	$\alpha(0.18), \mu(0.48)$
7	4	7	25
3	7	8	8
17	7	6	3
9	2	5	6
0	8	5	12
6	2	2	7
2	3	4	6
1	6	6	5
1	5	7	0
8	8	6	7
4	6	5	6
0	3	0	4
6	4	10	5
3	6	0	6
15	2	7	4
9	2	10	10
7	9	6	8
1	9	6	13
4	4	7	6
3	5	6	2

Dari data peramalan yang di dapat, hitung nilai MSE untuk mengetahui nilai yang terbaik dari percobaan yang telah dilakukan :

$$MSE = \frac{1}{n} \sum_{t=1}^n e_i^2 = \frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2 \dots\dots\dots(4.3)$$

dengan,

n = bilangan ramalan.

$y_t$  = nilai aktual pada waktu t.

$\hat{y}_t$  = nilai ramalan pada waktu t.

Menghitung nilai *msenya* yaitu :

1. Peramalan dengan *learning rate* 0.2 dan momentum 0.48

$$MSE = ((7-4)^2 + (3-7)^2 + (17-7)^2 + \dots + (4-4)^2 + (3-5)^2) / 20 = 30.6$$

2. Peramalan dengan *leraning rate* 0.21 dan momentum 0.49

$$MSE = ((7-7)^2 + (3-8)^2 + (17-6)^2 + \dots + (4-7)^2 + (3-6)^2) / 20 = 21.55$$

3. Peramalan dengan *learning rate* 0.18 dan momentum 0.48

$$MSE = ((7-25)^2 + (3-8)^2 + (17-3)^2 + \dots + (4-6)^2 + (3-2)^2) / 20 = 51.75$$

Dari tiga percobaan yang dilakukan pada metode jaringan syaraf tiruan dengan nilai *learing rate* dan momentum yang berbeda setelah di uji dan di prediksi memberikan nilai MSE dan hasil prediksi yang berbeda pula. Ini bearti nilai *learing rate* dan momentum sangat mempengaruhi keberhasilan jaringan syaraf tiruan. Dan nilai *learning rate* dan momentum yang memberikan hasil prediksi terbaik adalah nilai *learning rate* 0.21 dan momentum 0.48.

#### 4.4 Perancangan *Autoregressive*

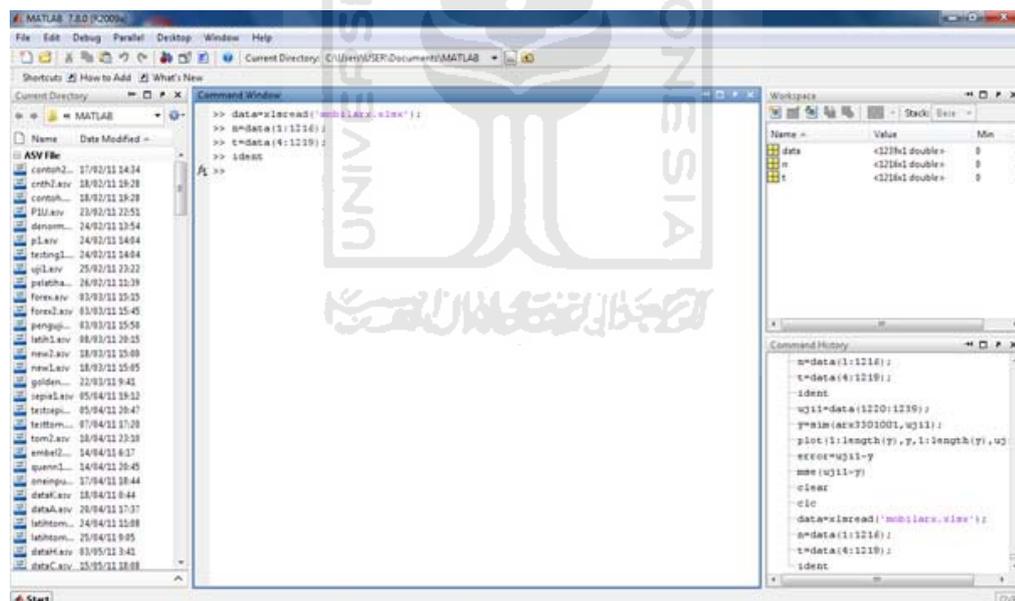
Pada pengolahan data dengan metode *Autoregressive* dilakukan empat kegiatan pokok yaitu :

1. Identifikasi Model
2. Estimasi Parameter dalam Model
3. Uji Diagnostik
4. Peramalan

##### 4.4.1 Pengolahan Data *Autoregressive* dengan Matlab

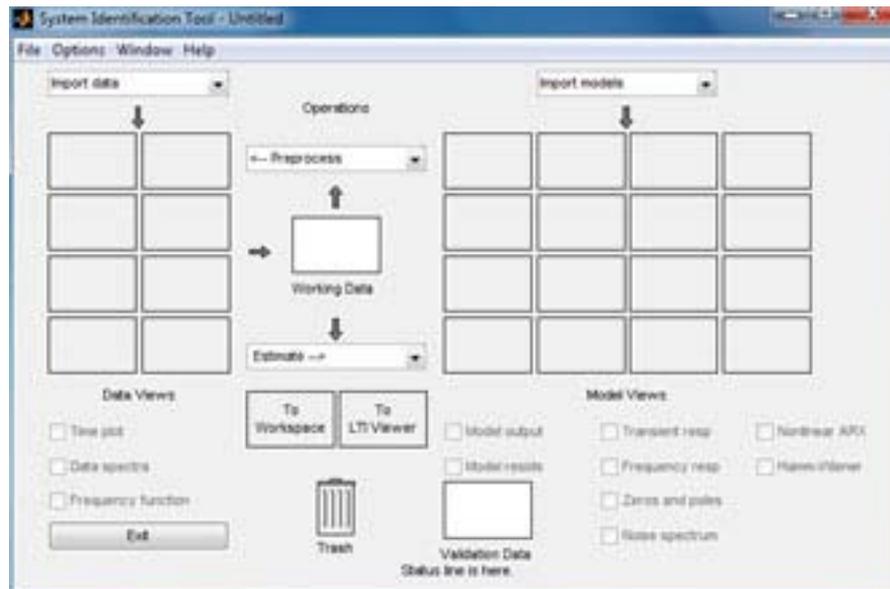
Adapun langkah-langkah dalam melakukan peramalan *Autoregressive* dengan *software* Matlab dapat dijelaskan sebagai berikut

1. Masukan perintah pada jendela *Command Window*



Gambar 4.13 Tampilan Jendela *Command Window*

2. Setelah mengetik perintah 'ident' untuk membuka jendela identifikasi yang maka akan muncul jendela seperti berikut ini :



Gambar 4.14 Tampilan Jendela *System Identification Tool*



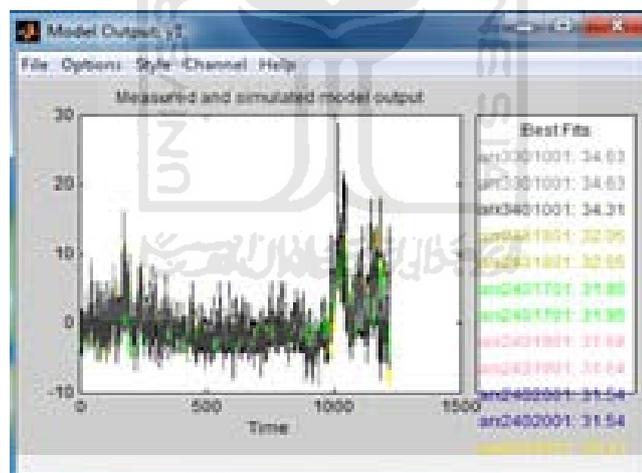
Gambar 4.15 Tampilan Jendela Linear Parameter Model

3. Lakukan identifikasi, parameter yang diperlukan dan verifikasi hasilnya, jika belum sesuai dengan yang diharapkan tentukan kembali tentukan parameternya dan verifikasi kembali.



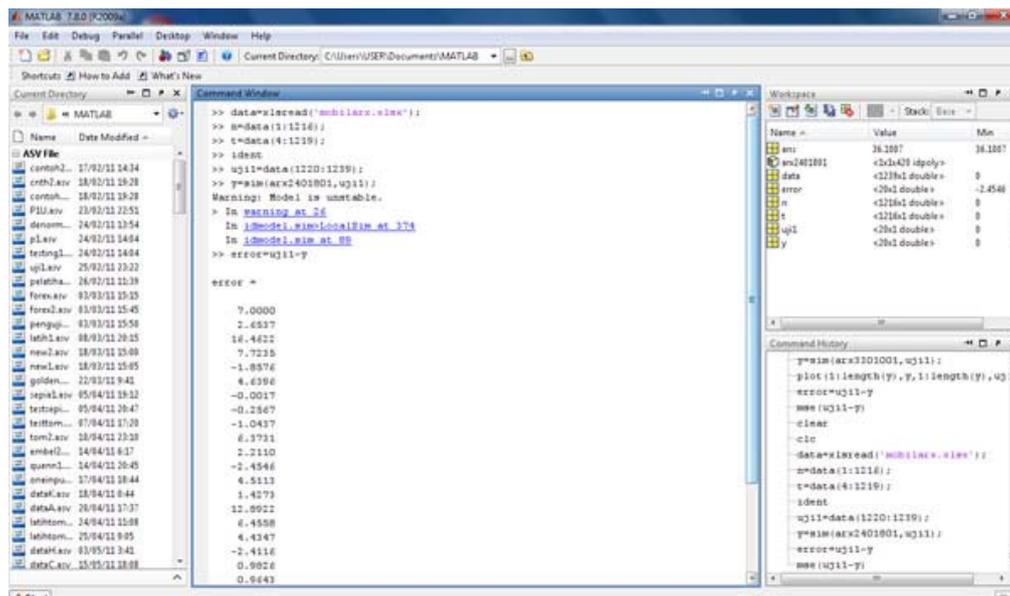
Gambar 4.16 Tampilan Jendela *System Identification Tool* dengan 13 Percobaan

4. Dari beberapa kali percobaan *Autoregressive* atau Arx pada Matlab yang memiliki *best fits* terbaik dapat dilihat dari gambar sebagai berikut :



Gambar 4.17 Tampilan Grafik dari Pelatihan

Kemudian pilihlah grafik yang memberikan *best fits* terbaik yaitu Arx [240 180 1] dan Arx [300 100 1] untuk di lakukan pengujian dan peramalan, maka kembali ke *Command Window* untuk memasukkan perintah selanjutnya :

Gambar 4.18 Tampilan Jendela *Command Window*

Berikut perhitungan *Autoregressive* dengan Matlab

Discrete-time IDPOLY model:  $A(q)y(t) = B(q)u(t) + e(t)$

$$\begin{aligned}
 A(q) = & 1 - 0.3355 q^{-1} - 0.08108 q^{-2} - 0.05987 q^{-3} - 0.04941 q^{-4} \\
 & - 0.02609 q^{-5} - 0.04029 q^{-6} + 0.007216 q^{-7} - 0.06598 q^{-8} \\
 & + 0.01775 q^{-9} - 0.05473 q^{-10} - 0.02172 q^{-11} + 0.01335 q^{-12} \\
 & - 0.00705 q^{-13} - 0.03279 q^{-14} + 0.01527 q^{-15} - 0.01852 q^{-16} \\
 & - 0.04293 q^{-17} + 0.02483 q^{-18} - 0.01078 q^{-19} - 0.01436 q^{-20} \\
 & - 0.02777 q^{-21} - 0.01991 q^{-22} + 0.01488 q^{-23} + 0.004318 q^{-24} \\
 & + 0.002271 q^{-25} - 0.01363 q^{-26} + 0.0009141 q^{-27} + 0.0006083 q^{-28} \\
 & + 0.006744 q^{-29} - 0.0008727 q^{-30} + 0.01642 q^{-31} - 0.03987 q^{-32}
 \end{aligned}$$

$$\begin{aligned}
 & \vdots \\
 & \vdots \\
 & + 0.004086 q^{-220} + 0.03114 q^{-221} - 0.007474 q^{-222} \\
 & + 0.004028 q^{-223} + 0.08079 q^{-224} - 0.003677 q^{-225} \\
 & + 0.07166 q^{-226} - 0.03754 q^{-227} - 0.04105 q^{-228} - 0.0408 q^{-229} \\
 & + 0.03119 q^{-230} - 0.07276 q^{-231} + 0.01147 q^{-232} - 0.0661 q^{-233} \\
 & - 0.03256 q^{-234} + 0.05548 q^{-235} + 0.1102 q^{-236} - 0.04489 q^{-237} \\
 & - 0.03361 q^{-238} - 0.08867 q^{-239} + 0.04067 q^{-240}
 \end{aligned}$$

$$\begin{aligned}
 B(q) = & 0.04948 q^{-1} + 0.03902 q^{-2} + 0.0157 q^{-3} + 0.0299 q^{-4} \\
 & - 0.01761 q^{-5} + 0.05558 q^{-6} - 0.02814 q^{-7} + 0.04434 q^{-8} \\
 & + 0.01132 q^{-9} - 0.02374 q^{-10} - 0.003344 q^{-11} + 0.0224 q^{-12} \\
 & - 0.02567 q^{-13} + 0.008125 q^{-14} + 0.03254 q^{-15} - 0.03522 q^{-16} \\
 & + 0.0003885 q^{-17} + 0.003961 q^{-18} + 0.01738 q^{-19} + 0.009516 q^{-20}
 \end{aligned}$$

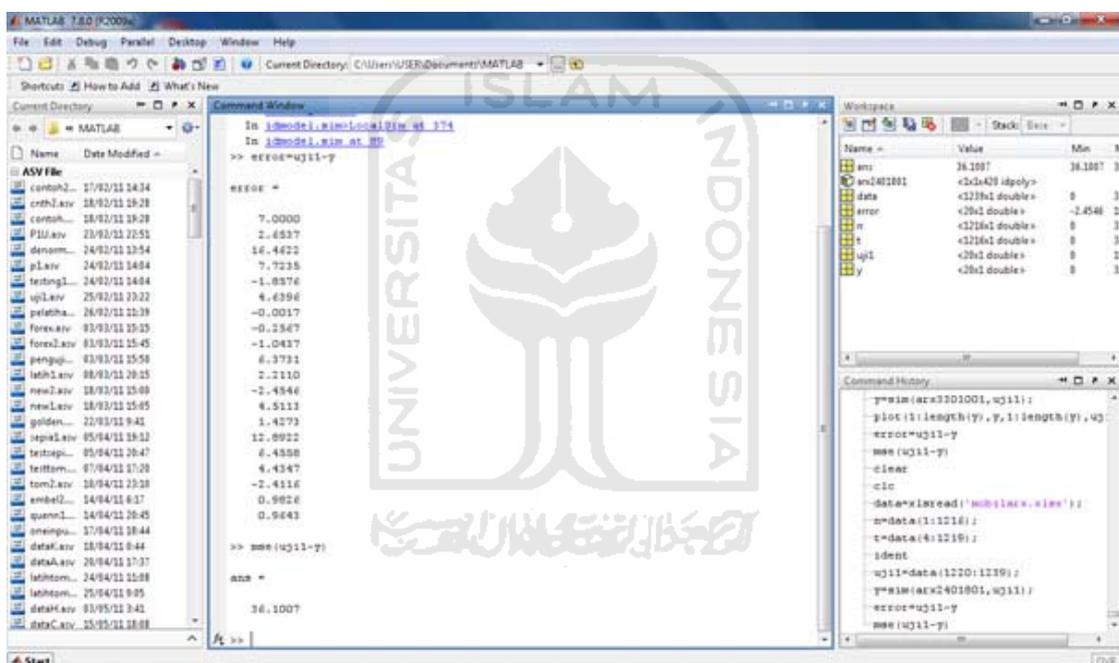
$\vdots$

$$\begin{aligned}
& + 0.01285 q^{-161} - 0.02054 q^{-162} - 0.03502 q^{-163} + 0.02683 q^{-164} \\
& - 0.04479 q^{-165} - 0.006176 q^{-166} - 0.008718 q^{-167} \\
& - 0.002329 q^{-168} + 0.005547 q^{-169} - 0.03404 q^{-170} \\
& - 0.009689 q^{-171} - 0.006369 q^{-172} + 0.03271 q^{-173} \\
& - 0.01531 q^{-174} - 0.01891 q^{-175} - 0.04555 q^{-176} - 0.00792 q^{-177} \\
& + 0.03236 q^{-178} - 0.01991 q^{-179} + 0.01673 q^{-180}
\end{aligned}$$

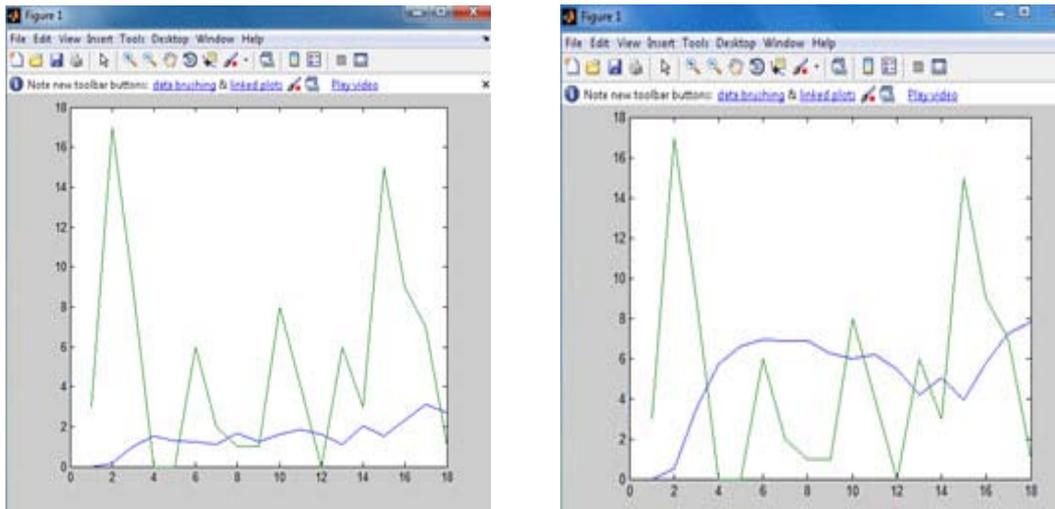
Estimated using ARX from data set eDat

Loss function 3.56911 and FPE 6.17612

5. Dari model *autoregressive* yang di dapat maka dapat dilakukan ke tahap selanjutnya yaitu mencari nilai MSE



Gambar 4.19 Tampilan Jendela *Command Window* untuk MSE Arx [240 180 1]



Gambar 4.20 Grafik Hasil Pengujian dengan Arx [240 180 1] (kiri) dan Hasil Pengujian dengan Arx [300 100 1] (kanan)

6. Dan hasil prediksi dengan metode *Autoregressive* sebagai berikut :

Tabel 4.3 Hasil Prediksi *Autoregressive*

Data	ARX [240 180 1]	ARX [330 100 1]
7	0	0
3	0	1
17	1	2
9	1	5
0	2	8
6	1	8
2	2	9
1	1	8
1	2	8
8	2	7
4	2	7
0	3	7
6	2	5
3	2	5
15	1	5
9	3	6
7	3	7
1	3	8
4	3	8
3	2	7

Dari data peramalan yang di dapat, hitung nilai MSE untuk mengetahui nilai yang terbaik dari percobaan yang telah dilakukan :

$$MSE = \frac{1}{n} \sum_{t=1}^n e_i^2 = \frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2 \dots\dots\dots(4.3)$$

dengan,

n = bilangan ramalan.

$y_t$  = nilai aktual pada waktu t.

$\hat{y}_t$  = nilai ramalan pada waktu t.

Menghitung nilai MSEnya yaitu :

1. Peramalan dengan model ARX [240 180 1]

$$MSE = ((7-0)^2 + (3-0)^2 + (17-1)^2 + \dots + (4-3)^2 + (3-2)^2) / 20 = 36.1$$

2. Peramalan dengan model ARX [ 330 100 1]

$$MSE = ((7-0)^2 + (3-1)^2 + (17-2)^2 + \dots + (4-8)^2 + (3-7)^2) / 20 = 38.51$$

Dari tabel dapat dilihat nilai prediksi dan nilai MSE dari ke-2 percobaan yang telah dilakukan dengan model Ar [ 240 180 1] memberikan hasil yang lebih mendekati nilai aktual di dibandingkan dengan model Ar [330 100 1].

#### 4.5 Pengolahan Data dengan Metode *Moving Average*

Pengolahan data dengan metode *Moving Average* dihitung secara manual dengan menggunakan rumus seperti di bawah ini.

$$F_{T+1} = \frac{X_1 + X_2 + \dots + X_T}{T} \dots\dots\dots(4.4)$$

dimana :

$F_{T+1}$  = Ramalan untuk periode ke t+1

$X_T$  = Nilai aktual periode ke t

T = Jangka waktu rata-rata bergerak

Untuk mengetahui nilai peramalan dan *error* terbaik, dilakukan 2 percobaan dengan 3 periode dan 5 periode dengan menggunakan data bulan Desember 2008. Perhitungan dengan metode *Moving Average* dapat dilihat pada tabel berikut :

Tabel 4.4 Perhitungan Peramalan dengan 3 Periode dan 5 Periode

Penjualan (Des-08)	<i>Moving Average</i> 3 Periode	<i>Peramalan</i>	<i>Moving Average</i> 5 Periode	<i>Peramalan</i>
7	$(19+5+11)/3 = 11.6$	12	$(7+6+19+5+11)/5 = 9.6$	10
3	$(5+11+7)/3 = 7.6$	8	$(6+19+5+11+7)/5 = 9.6$	10
17	$(11+7+3)/3 = 7$	7	$(19+5+11+7+3)/5 = 9$	9
9	$(7+3+17)/3 = 9$	9	$(5+11+7+3+17)/5 = 8.6$	9
0	$(3+17+9)/3 = 9,6$	10	$(11+7+3+17+9)/5 = 9.4$	9
6	$(17+9+0)/3 = 8.6$	9	$(7+3+17+9+0)/5 = 7.2$	7
2	$(9+0+6)/3 = 5$	5	$(3+17+9+0+6)/5 = 7$	7
1	$(0+6+2)/3 = 2.6$	3	$(17+9+0+6+2)/5 = 6.8$	7
1	$(6+2+1)/3 = 3$	3	$(9+0+6+2+1)/5 = 3.6$	4
8	$(2+1+1)/3 = 1.3$	1	$(0+6+2+1+1)/5 = 2$	2
4	$(1+1+8)/3 = 3.3$	3	$(6+2+1+1+8)/5 = 3.6$	4
0	$(1+8+4)/3 = 4.3$	4	$(2+1+1+8+4)/5 = 3.2$	3
6	$(8+4+0)/3 = 4$	4	$(1+1+8+4+0)/5 = 2.8$	3
3	$(4+0+6)/3 = 3.3$	3	$(1+8+4+0+6)/5 = 3.8$	4
15	$(0+6+3)/3 = 3$	3	$(8+4+0+6+3)/5 = 4.2$	4
9	$(6+3+15)/3 = 8$	8	$(4+0+6+3+15)/5 = 5.6$	6
7	$(3+15+9)/3 = 9$	9	$(0+6+3+15+9)/5 = 6.6$	7
1	$(15+9+7)/3 = 10.3$	10	$(6+3+15+9+7)/5 = 8$	8
4	$(9+7+1)/3 = 5.6$	6	$(3+15+9+7+1)/5 = 7$	7
3	$(7+1+4)/3 = 4$	4	$(15+9+7+1+4)/5 = 7.2$	7

Menghitung nilai MSE pada metode *Moving Average* yaitu :

$$MSE = \frac{\sum |X_t - F_t|^2}{n} \dots\dots\dots(4.5)$$

1. MSE 3 periode =  $((-4.6)^2 + (-4.6)^2 + (10)^2 + \dots + (-1)^2) / 20 = 29$

2. MSE 5 periode =  $((-2.6)^2 + (-6.6)^2 + (8)^2 + \dots + (-4.2)^2) / 20 = 25.9$

Dari perhitungan MSE dapat dilihat *Moving Average* dengan 5 periode memiliki MSE yang lebih kecil dari pada perhitungan *Moving Average* dengan 3 periode. Maka *Moving Average* dengan 5 periode memberikan hasil peramalan yang lebih baik dari 3 periode.



## BAB V

### PEMBAHASAN

#### 5.1 Pembahasan

Dari hasil penelitian yang telah dilakukan dengan 3 metode dibandingkan dengan fakta data di lapangan yakni data penjualan mobil Suzuki Baleno bulanan Desember 2008 dapat dilihat pada tabel berikut ini :

Tabel 5.1 Perbandingan Data Aktual JST, AR dan MA

Data Aktual	JST	<i>Autoregressive</i>	<i>Moving Average</i>
7	7	0	10
3	8	0	10
17	6	1	9
9	5	1	9
0	5	2	9
6	2	1	7
2	4	2	7
1	6	1	7
1	7	2	4
8	6	2	2
4	5	2	4
0	0	3	3
6	10	2	3
3	0	2	4
15	7	1	4

9	10	3	6
7	6	3	7
1	6	3	8
4	7	3	7
3	6	2	7
MSE	21.55	36.1	25.9

Berdasarkan tabel diatas, dapat dilihat metode jaringan syaraf tiruan memberikan nilai MSE yang terbaik dari pada metode *Autoregressive* dan metode *Moving Average*. Maka dari itu nilai prediksi mobil Baleno 4 hari mendatang menggunakan metode jaringan syaraf tiruan yaitu,  $t + 1 = 6$  unit,  $t+2 = 6$  unit,  $t + 3 = 8$  unit dan  $t + 4 = 6$  unit mobil.

Pada proses perhitungan jaringan syaraf tiruan hanya menggunakan nilai *learning rate* ( $\alpha$ ) 0.2 – 0.21 dan 0.18 sedangkan *learning rate* ( $\alpha$ ) terletak antara 0 dan 1, sehingga ada kemungkinan jika  $\alpha$  berbeda maka hasilnya juga berbeda.

Hasil *forecasting* atau nilai ramalan volume penjualan ini mengalami peningkatan jika dibandingkan dengan bulan sebelumnya. Peningkatan nilai ramalan volume penjualan ini juga terjadi jika dibandingkan dengan bulan Desember 2006 dan bulan Desember 2007. Hal ini terjadi karena adanya faktor musiman, seperti minat konsumen yang tidak menentu, dan munculnya produk baru baik yang dikeluarkan oleh perusahaan sendiri maupun yang dikeluarkan oleh perusahaan lain.

Hasil ramalan volume penjualan ini tidak sepenuhnya harus digunakan dalam pengambilan keputusan (perusahaan) mengenai banyaknya mobil Suzuki Baleno yang harus diproduksi, melainkan harus dipertimbangkan lagi dengan memperhatikan aspek-aspek lainnya yang mempengaruhi tingkat volume penjualan mobil Suzuki

Baleno. Hal ini perlu dilakukan karena melihat nilai *forecasting* volume penjualan mobil Suzuki Baleno pada bulan-bulan yang sudah diketahui volume penjualannya dengan nilai volume penjualan mobil Suzuki Baleno pada bulan tersebut berbeda cukup jauh. Oleh karena itu perusahaan harus dipertimbangkan lagi jika akan menggunakan nilai ramalan ini sebagai dasar dalam penyediaan (produksi) mobil Suzuki Baleno untuk periode berikutnya.



## BAB VI

### KESIMPULAN DAN SARAN

#### 6.1 Kesimpulan

Dari hasil penelitian dan pembahasan maka penulis dapat menarik kesimpulan sebagai berikut :

1. Dari tiga metode yang digunakan untuk memprediksi penjualan mobil Suzuki Baleno diperoleh nilai MSE yaitu metode jaringan syaraf tiruan sebesar 21,55, metode *Autoregressive* sebesar 36,1 dan metode *Moving Average* sebesar 25,9. Dari nilai MSE yang diperoleh metode jaringan syaraf tiruan memberikan nilai MSE yang terkecil.
2. Nilai prediksi penjualan mobil Suzuki Baleno berdasarkan metode jaringan syaraf tiruan yaitu  $t + 1 = 6$  unit,  $t + 2 = 6$  unit,  $t + 3 = 8$  unit dan  $t + 4 = 6$  unit mobil.

#### 6.2 Saran

Berdasarkan hasil penelitian maka saran yang dapat disampaikan adalah sebagai berikut :

1. Diperlukan penelitian lebih lanjut tentang metode-metode *forecasting* yang lebih praktis, lebih efisien serta menghasilkan *forecast error* yang lebih kecil dibandingkan dengan metode Jaringan Syaraf Tiruan *Backpropagation*, metode *Autoregressive* dan metode *Moving Average*.
2. Untuk perusahaan perlu untuk meningkatkan peramalan volume penjualan mobil periode mendatang untuk meningkatkan produksi sehingga keuntungan yang diperoleh akan maksimal.

## DAFTAR PUSTAKA

- Hermawan, A. (2006). *Jaringan Saraf Tiruan Teori dan Aplikasi*. Yogyakarta : Andi
- Haykin, S. (1999). *Neural Network a Comprehensive Foundation*. New Jersey :  
Prentice Hall
- Kusumadewi, S. (2004). *Membangun Jaringan Syaraf Tiruan (Menggunakan Matlab dan Excel Link)*. Yogyakarta : Graha Ilmu
- Marimin. (2005). *Teori dan Aplikasi Sistem Pakar dalam Teknologi Manajerial*.  
Bogor : Ed.2. IPB Press
- Siang, J. J. (2005). *Jaringan Syaraf Tiruan dan Pemogramannya Menggunakan Matlab*. Yogyakarta : Andi
- Subagyo, P. (1986). *Forecasting Konsep Dan Aplikasi*. Yogyakarta : BPFE  
Yogyakarta.
- Supranto, J. (2001). *Teknik Riset Pemasaran Dan Ramalan Penjualan*. Jakarta :  
PT. Rineka Cipta.
- Priyambodo. (2009). *Peramalan Penjualan (Forecasting)*.  
<http://priyambodo71.wordpress.com/manajemen-operasi/forecasting...>  
(diakses 11 Des 2009)
- Puspitaningrum, D. (2006). *Pengantar Jaringan Syaraf Tiruan*. Yogyakarta : Andi
- Puspita, A. dan Eunike. (2007). *Penggunaan Jaringan Saraf Tiruan Metode Backpropagation Untuk Memprediksi Bibir Sumbing*. Yogyakarta.
- Yani, E. (2005). *Pengantar Jaringan Syaraf Tiruan*.  
[http://trirezqiarantoro.files.worpress.com/2007/05/jaringan\\_syaraf\\_tiru](http://trirezqiarantoro.files.worpress.com/2007/05/jaringan_syaraf_tiru)  
[an.pdf](#). (diakses 10 Jan.2010)



## Program Utama Autoregressive

```
data = xlsread('mobilarx.xlsx');
```

```
n = data (1:1216);
```

```
t = data (4:1219);
```

```
ident
```

```
Opening System Identification Tool ..... done.
```

```
Uji1 = data(1220:1239);
```

```
y = sim(arx2401801,uji1);
```

```
plot(1:length(y),y,1:length(y),uji1);
```

```
Uji2 = data(1220:1239);
```

```
y = sim(arx2401801,uji2);
```

```
plot(1:length(y),y,1:length(y),uji2);
```

```
error = uji2-y
```

```
mse=(uji2-y)
```

Catatan :

Semua perintah dilakukan pada jendela *Command Window* dan untuk identifikasi parameter AR dilakukan pada jendela *Identification System Tool*

