# PENERAPAN *NEURAL NETWORK* DALAM PENDUGAAN HARGA SAHAM PADA PERUSAHAAN PERKEBUNAN : STUDI KASUS PADA PT. ASTRA AGRO LESTARI Tbk

**TESIS**

**LINDA SARI DEWI**

**14000370**

**PROGRAM PASCASARJANA MAGISTER ILMU KOMPUTER**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**

**NUSA MANDIRI**

**JAKARTA**

**2012**

# PENERAPAN *NEURAL NETWORK* DALAM PENDUGAAN HARGA SAHAM PADA PERUSAHAAN PERKEBUNAN : STUDI KASUS PADA PT. ASTRA AGRO LESTARI Tbk

## TESIS

**Diajukan sebagai salah satu syarat untuk memperoleh gelar**

**Magister Ilmu Komputer (M.Kom)**

**LINDA SARI DEWI**

**14000370**

**PROGRAM PASCASARJANA MAGISTER ILMU KOMPUTER**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**

**NUSA MANDIRI**

**JAKARTA**

**2012**

# BAB 1

# PENDAHULUAN

## 1.1 Latar Belakang

Pada perkembangan bisnis dunia, investasi saham menjadi salah satu alternatif untuk masyarakat dunia menanamkan modalnya dengan tujuan mendapat keuntungan yang akan datang. Prediksi bursa saham adalah salah satu tugas yang menantang bagi investor keuangan di dunia. Tantangan ini disebabkan oleh ketidakpastian harga saham di pasar saham.

Pasar modal adalah kegiatan yang bersangkutan dengan penawaran umum dan perdagangan efek, perusahaan publik yang berkaitan dengan efek yang diterbitkannya, lembaga dan profesi yang berkaitan dengan efek ( UU No.8 Tahun 1995 ).

Saham adalah tanda penyertaan atau kepemilikan seseorang atau badan dalam suatu perusahaan atau perusahaan terbatas serta wujud saham berupa selembar kertas yang menerangkan siapa pemiliknya ( Setiawan, 2008 ).

Penentuan harga saham telah menarik banyak peneliti di bursa saham. Studi terbaru menunjukkan bahwa penggunaan *Neural Network* dalam lingkungan bisnis telah meningkat selama beberapa tahun terakhir. Neural Network telah mendapatkan popularitas lebih karena kemampuannya untuk mendekati fungsi non linier ke tingkat akurasi yang tinggi (Sureshkumar, 2012). Dan *Model Multilayer Perceptron (MLP)* dan *Radial Basis Function (RBF)* algoritma paling populer di dalam *neural network* karena terbukti mampu untuk menyelesaikan berbagai macam masalah (Rabunal, 2006).

Penting bagi investor dalam memprediksi harga saham untuk mengetahui bagaimana prospek investasi suatu saham perusahaan di masa yang akan datang

dengan cepat dan akurat. Tinggi rendahnya harga saham dipengaruhi oleh banyak faktor seperti kondisi dan kinerja perusahaan, resiko, deviden, tingkat suku bunga, kondisi perekonomian, kebijaksanaan pemerintah, laju inflasi, penawaran dan permintaan serta masih banyak lagi (Setiawan, 2008).

Untuk memprediksi harga saham yang akan datang diperlukan suatu model yang mampu memprediksi harga saham tersebut menggunakan data harga saham yang disimpan dalam database atau data warehouse beberapa tahun yang lalu.Maka dengan itu, dilakukan komparasi dengan menggunakan Pendekatan *Neural Network* yaitu model *Mulitlayer Perceptron (MLP)* dan model *Radial Basis Function (RBF).* Untuk mengetahui model mana yang tepat digunakan untuk memprediksi harga saham yang akan datang dengan cepat dan akurasi yang tinggi.

PT. Astra Argo Lestari adalah sebuah perusahaan yang bergerak dibidang perkebunan khususnya sebagai industri kelapa sawit mentah ke pasar dunia. Pada tanggal 9 Desember 1997 perusahaan ini mencatatkan sahamnya di bursa efek sampai saat ini perusahaan menawarkan sahamnya. Dari tahun 1997 sampai saat ini harga saham tercatat dalam database pasar saham. Maka dengan menggunakan data-data historis tersebut, diduga investor dapat memprediksi harga saham yang akan datang tersebut sehingga investor dapat melihat prospek kedepan.

Hal ini menarik perhatian peneliti untuk melakukan riset dan memanfaatkan data yang ada dengan menerapkan *data mining* berdasarkan algoritma neural network dalam pendugaan harga saham yang akan datang.

*Data mining* pada dasarnya berhubungan dengan analisa data dan penggunaan teknik-teknik perangkat lunak dalam mencari pola dan keteraturan dalam himpunan data yang sifatnya tersembunyi.

Pendekatan menggunakan *neural network* adalah pendekatan untuk meningkatkan keakurasian dalam meramalkan hasil yang akan datang menggunakan pola data yang beragam (Sureshkumar, 2012).

Dari data historis saham tersebut, maka peneliti berusaha memanfaatkan data yang ada untuk meramalkan harga saham pada tahun berikutnya yang dapat digunakan oleh perusahaan dan investor. Perangkat lunak yang akan digunakan adalah DTReg, sebagai mesin pembelajaran yang *open-source,* yang mendukung dalam pengolahan data menggunakan *Neural Network* untuk mendapatkan harga saham yang akan datang.

## 1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, dapat disimpulkan bahwa untuk mempredikasi harga saham yang akan datang, dapat dilakukan dengan menggunakan teknik analisis data historis yaitu dari data harga saham yang lalu.

Membahas semua masalah tersebut memang memungkinkan, namun dalam melakukan riset, peneliti hanya membatasi tentang pendugaan harga saham pada PT. Astra Agro Lestari.

Sehubungan dengan itu, maka masalah penelitian dapat dirumuskan sebagai berikut :

1. Bagaimana memanfaatkan data yang ada, agar diperoleh suatu manfaat bagi investor?
2. Apakah pendekatan neural network dapat digunakan untuk memprediksi harga saham PT. Astra Argo Lestari yang akan datang?
3. Apakah dengan menggunakan neural network dapat digunakan untuk merekomendasikan investor tentang saham yang akan datang?

## 1.3 Tujuan Penelitian

Berdasarkan dari rumusan masalah yang ada, maka tujuan yang dicapai pada penelitian ini adalah :

1. Memanfaatkan data yang ada untuk mendapatkan informasi yang berguna bagi perusahaan dan investor.
2. Memberikan pengetahuan tentang pendugaan harga saham kepada perusahaan dan investor untuk dijadikan bahan pertimbangan dalam pembelian saham dengan menerapkan *neural network* menggunakan *Multilayer Perceptron (MLP)* dan *Radial Basis Function (RBF).*
3. Untuk meningkatkan tingkat keakuratan prediksi harga saham yang akan datang.

## 1.4 Ruang Lingkup Penelitian

Batasan penelitian dari permasalahan ini adalah :

1. Metode yang digunakan pada pendugaan harga saham ini adalah dengan menerapkan *neural network* menggunakan *Multilayer Perceptron (MLP)* dan *Radial Basis Function (RBF)*.
2. Hanya akan menganalisis harga saham yang nantinya dapat digunakan untuk strategi pembelian di bursa saham.
3. Data yang digunakan bersumber dari PT. Astra Agro Lestari dengan tahun pengamatan 2006, pada tanggal 1 Januari 2006 sampai dengan 30 Desember 2011 yang berfokus pada harga saham.

## 1.5 Hipotesis

Pada penelitian ini hipotesis yang diajukan adalah diduga *neural network* yang berdasarkan model *Multilayer Perceptron (MLP)* dan *Radial Basis Function (RBF*) dapat bermanfaat dalam menentukan harga saham yang akan datang pada PT. Astra Agro Lestari yang dapat dimanfaatkan oleh para investor.

## 1.6 Sistematika Penelitian

Hasil penelitian ini ditulis dalam lima bab, yaitu :

BAB 1 PENDAHULUAN

Pada bab pertama akan menguraikan latar belakang masalah dari penelitian, identifikasi masalah, tujuan penelitian, ruang lingkup penelitian, hipotesis, dan sistematika penelitian.

BAB 2 LANDASAN / KERANGKA PEMIKIRAN

Pada bab dua ini menjelaskan teori-teori yang mendukung dalam penelitian yang terdiri dari tinjauan pustaka, tinjauan studi serta tinjauan organisasi atau obyek penelitian.

BAB 3 METODOLOGI PENELITIAN

Bab ini membahas tentang metode pengumpulan data dan metode penelitian serta eksperimen. Eksperimen dilakukan dengan metode *Neural Network ( NN )* dengan model *Multilayer Perceptron (MLP)* dan *Radial Basis Function (RBF)* untuk meningkatkan akurasi terutama dalam memprediksi harga saham yang lebih akurat.

BAB 4 HASIL PENELITIAN DAN PEMBAHASAN

Bab ini membahas tentang pengujian model yang dihasilkan dari bab sebelumnya. Pengujian dilakukan dengan mengukur kinerja menggunakan metode *Neural Network ( NN )* untuk melihat akurasi dari metode tersebut dan penggunaan *software* dalam mengolah data. Serta menjelaskan hasil penelitian dan analisa yang diperoleh.

BAB 5 PENUTUP

Bab ini membahas kesimpulan dari penelitian dan saran untuk penelitian selanjutnya.

# BAB 2

# LANDASAN TEORI / KERANGKA PEMIKIRAN

## 2.1 Tinjauan Pustaka

Dalam penelitian tesis ini peneliti melakukan tinjauan studi dengan menggunakan buku dan jurnal yang berhubungan dengan tema yang peneliti pilih.

### 2.1.1 *Data Mining*

Menurut Daryl Pregibons *Data mining* adalah perpaduan dari ilmu statistik, kecerdasan buatan, dan penelitian bidang *database* (Gorunescu, 2011). *Data mining* adalah bagian dari pengenalan pola, pengenalan pola adalah suatu disiplin ilmu yang mempelajari bagaimana kita mengelompokan obyek ke berbagai kelas dan bagaimana dari data bisa kita temukan kecenderungannya.

Menurut Gartner *Data Mining* adalah suatu proses menemukan hubungan yang berarti, pola, dan kecenderungan dengan memeriksa dalam sekumpulan besar data yang tersimpan dalam penyimpanan dengan menggunakan teknik pengenalan pola seperti teknik statistik dan matematika (Larose, 2005).

Adapun definisi yang lain yaitu (Larose, 2005):

a. *Data mining* merupakan analisis dari peninjauan kumpulan data untuk menemukan hubungan yang tidak diduga dan meringkas data dengan cara yang berbeda dengan sebelumnya, yang dapat dipahami dan bermanfaat bagi pemilik data.
b. *Data mining* merupakan bidang dari beberapa bidang keilmuan yang menyatukan teknik dari pembelajaran mesin, pengenalan pola, statistik, *database*, dan visualisasi untuk penanganan permasalahan pengambilan informasi dari database yang besar.

Kemajuan luar biasa yang terus berlanjut dalam bidang *data mining* didorong oleh beberapa faktor, antara lain (Larose, 2005):

1. Pertumbuhan yang cepat dalam kumpulan data.
2. Penyimpanan data dalam *data warehouse*, sehingga seluruh perusahaan memiliki akses ke dalam *database* yang andal.
3. Adanya peningkatan akses data melalui navigasi web dan intranet.
4. Tekanan kompetisi bisnis untuk meningkatkan penguasaan pasar dalam globalisasi ekonomi.
5. Perkembangan teknologi perangkat lunak untuk *data mining* (ketersediaan teknologi).
6. Perkembangan yang hebat dalam kemampuan komputasi dan pengembangan kapasitas media penyimpanan.

Dari definisi-definisi yang telah disampaikan, hal penting yang terkait dengan *data mining* adalah (Kusrini, 2009):

1. *Data mining* merupakan suatu proses otomatis terhadap data yang sudah ada.
2. Data yang akan diproses berupa data yang sangat besar.
3. Tujuan *data mining* adalah mendapatkan hubungan atau pola yang mungkin memberikan indikasi yang bermanfaat.

Istilah *data mining* dan *knowledge discovery in database* (KDD) seringkali digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yang besar. Sebenarnya kedua istilah tersebut memiliki konsep yang berbeda, tetapi berkaitan satu sama lain. Dan salah satu tahapan dalam keseluruhan proses KDD adalah *data mining*. Proses KDD secara garis besar dapat dijelaskan sebagai berikut (Larose, 2005):

1. *Data Selection*

   Pemilihan (seleksi) data dari sekumpulan data operasional perlu di lakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang akan digunakan untuk proses *data mining*, disimpan dalam suatu berkas, terpisah dari basis data operasional.

2. *Pre-processing/ Cleaning*

Sebelum proses *data mining* dapat di laksanakan, perlu dilakukan proses *cleaning* pada data yang menjadi fokus KDD. Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak (*tipografi*). Juga dilakukan proses *enrichment*, yaitu proses "memperkaya" data yang sudah ada dengan data atau informasi lain yang relevan dan diperlukan untuk KDD, seperti data atau informasi eksternal.

3. *Transformation*

   *Coding* adalah proses transformasi pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses *data mining*. Proses *coding* dalam KDD merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam basis data.

4. *Data mining*

   *Data mining* adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algoritma dalam *data mining* sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses KDD secara keseluruhan.

5. *Interpretation/ Evaluation*

   Pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini merupakan dari bagian dari proses KDD yang di sebut *interpretation*. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesis yang ada sebelumnya.

Data mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu (Larose, 2005):

1. Deskripsi

Terkadang peneliti dan analis secara sederhana ingin mencoba mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Sebagai contoh, petugas pengumpulan suara mungkin tidak dapat menemukan keterangan atau fakta bahwa siapa yang tidak cukup professional akan sedikit didukung dalam pemilihan presiden. Deskripsi dari pola dan kecenderungan sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.

2. Estimasi

Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih ke arah numerik dari pada ke arah kategori. Model dibangun menggunakan *record* lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi. Selanjutnya, pada peninjauan berikutnya estimasi nilai dari variabel target dibuat berdasarkan nilai variabel prediksi, Sebagai contoh, akan dilakukan estimasi tekanan darah sistolik pada pasien rumah sakit berdasarkan umur pasien, jenis kelamin, indeks berat badan, dan level sodium darah. Hubungan antara tekanan darah sistolik dan nilai variabel prediksi dalam proses pembelajaran akan menghasilkan model estimasi. Model estimasi yang dihasilkan dapat digunakan untuk kasus baru lainnya.

Contoh lain yaitu estimasi nilai indeks prestasi kumulatif mahasiswa program pascasarjana dengan melihat nilai indeks prestasi mahasiswa tersebut pada saat mengikuti program sarjana.

3. Prediksi

Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan di masa mendatang. Contoh prediksi dalam bisnis dan penelitian adalah:

a. Prediksi harga beras dalam tiga bulan yang akan datang.
b. Prediksi persentase kenaikan kecelakaan lalu lintas tahun depan jika batas bawah kecepatan dinaikan.

Beberapa metode dan teknik yang di gunakan dalam klasifikasi dan estimasi dapat pula digunakan (untuk keadaan yang tepat) untuk prediksi.

4. Klasifikasi

Dalam klasifikasi, terdapat target variabel kategori. Sebagai contoh, pengolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang, pendapatan rendah. Contoh lain klasifikasi dalam bisnis dan penelitian adalah :

a. Menentukan apakah suatu transaksi kartu kredit merupakan transaksi yang curang atau bukan.
b. Memperkirakan apakah suatu pengajuan hipotek oleh nasabah merupakan suatu kredit yang baik atau buruk.
c. Mendiagnosis penyakit seorang pasien untuk mendapatkan termasuk kategori penyakit apa.

5. Pengklusteran

Pengklusteran merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan. Kluster adalah kumpulan *record* yang memiliki kemiripan satu dengan yang lainnya dan memiliki ketidak miripan dengan *record-record* dalam kluster lain. Pengklusteran berbeda dengan klasifikasi yaitu tidak adanya variabel target dalam pengklusteran. Pengklusteran tidak mencoba untuk melakukan klasifikasi, mengestimasi, atau memprediksi nilai dari variabel target, Akan tetapi, algoritma pengklusteran mencoba untuk melakukan pembagian terhadap keseluruhan data menjadi kelompok-kelompok yang memiliki kemiripan (homogen), yang mana kemiripan *record* dalam satu kelompok akan bernilai maksimal, sedangkan kemiripan dengan *record* dalam kelompok lain akan bernilai minimal. Contoh pengklusteran dalam bisnis dan penelitian adalah:

a. Mendapatkan kelompok-kelompok konsumen untuk target pemasaran dari suatu produk bagi perusahaan yang tidak memiliki dana pemasaran yang besar.
b. Untuk tujuan audit akuntansi, yaitu melakukan pemisahan terhadap perilaku financial dalam baik dan mencurigakan.
c. Melakukan pengklusteran terhadap ekspresi dari gen, untuk mendapatkan kemiripan perilaku dari gen dalam jumlah besar.

6. Asosiasi

Tugas asosiasi dalam *data mining* adalah menemukan yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang belanja. Contoh asosiasi dalam bisnis dan penelitian adalah:

a. Meneliti jumlah pelanggan dari perusahaan telekomunikasi seluler yang diharapkan untuk memberikan respons positif terhadap penawaran *upgrade* layanan yang diberikan.
b. Menemukan barang dalam supermarket yang dibeli secara bersamaan dan barang yang tidak pernah dibeli secara bersamaan.

### *2.1.2 Neural Network*

*Neural Network (* Jaringan Saraf Tiruan ) merupakan suatu studi tentang peramalan runtut waktu (*time series*) sebagai *input* dan target dari *output* yang diinginkan pada proses pelatihan adalah data periode sebelum tahun yang akan diramal, data tersebut digunakan untuk menentukan bobot yang optimal. Setelah bobot optimal didapatkan dari proses pelatihan, bobot-bobot tersebut digunakan untuk menentukan nilai peramalan jika sistem diuji oleh data yang pernah masuk dalam sistem peramalan (Setiawan, 2008).

Langkah-langkah prediksi harga saham dengan *neural network* adalah sebagai berikut:

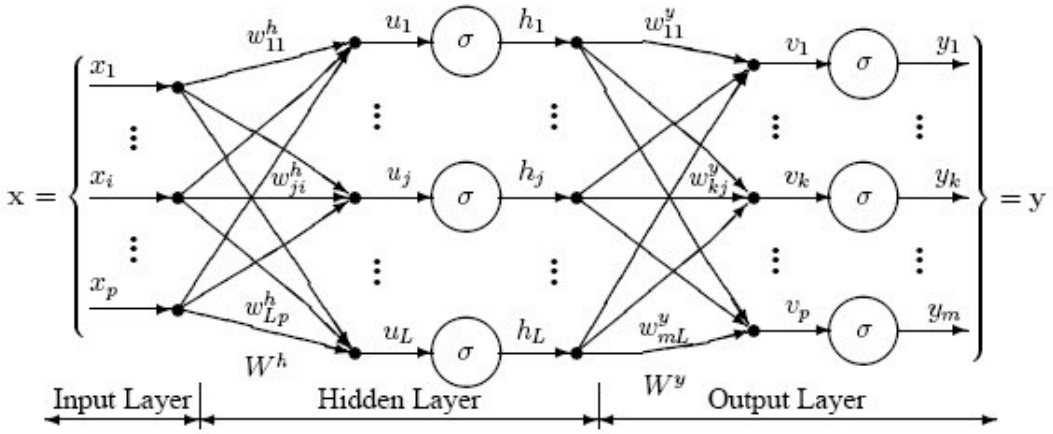
1. Input data saham periode sebelumnya
2. Data akan diproses melalui 2 tahap, yaitu tahap pelatihan dan tahap pengujian
3. Output akan didapatkan berupa harga saham yang akan datang

#### **2.1.2.1 *Multilayer Perceptron (MLP)***

##### ***2.1.2.1.1 Arsitektur Multilayer Perceptron (MLP)***

*Arsitektur Multilayer Perceptron (MLP)* merupakan model yang sangat sering digunakan dalam model peramalan time series. Gambar 2.1 menunjukan *MLP* yang terdiri dari tiga unit layer yang biasa digunakan untuk tujuan peramalan. Input layer adalah lag observasi awal, sedangkan output memberikan hasil peramalan untuk data berikutnya. *Hidden layer* biasanya merupakan fungsi transfer nonlinier yang digunakan untuk memproses informasi

yang di terima dari input layer.




**Gambar 2.1 Arsitektur dari Model *Multilayer Perceptron (MLP)***

Jaringan ini memiliki lapisan input (di sebelah kiri) dengan tiga neuron, satu lapisan tersembunyi (di tengah) dengan tiga neuron dan lapisan output (di sebelah kanan) dengan tiga neuron.

Jaringan ini memiliki lapisan input (di sisi kiri) dengan tiga sel saraf, satu lapisan tersembunyi (di bagian tengah) dengan tiga sel saraf dan sebuah lapisan output (di sisi kanan) dengan tiga sel saraf. Terdapat satu sel saraf pada lapisan input untuk setiap variabel penduga ($x_1$ ,..., $x_p$).

**Lapisan Input**

Sebuah vektor nilai variabel penduga ($x_1$ ,..., $x_p$) disajikan sebagai lapisan input. Lapisan input mendistribusikan nilai-nilai tersebut kepada setiap sel saraf dalam lapisan tersembunyi. Selain variabel penduga, terdapat input konstanta bernilai 1.0, yang disebut bias yang dimasukkan kedalam setiap lapisan tersembunyi; bias tersebut dikalikan dengan berat dan ditambahkan pada keseluruhan nilai yang menuju sel saraf.

**Lapisan Tersembunyi**

Setelah tiba pada sel saraf dalam lapisan tersembunyi, nilai dari setiap sel saraf input dikalikan dengan berat ($w_{ji}$), dan nilai yang dihasilkan setelah diberi beban (dikalikan dengan beban) ditambahkan bersama-sama menghasilkan nilai kombinasi $u_j$. Jumlah setelah dibebani ($u_j$) dimasukkan kedalam fungsi transfer, $\sigma$, yang menghasilkan nilai output $h_j$. Nilai output dari lapisan tersembunyi didisribusikan menuju lapisan output.

**Lapisan Output**

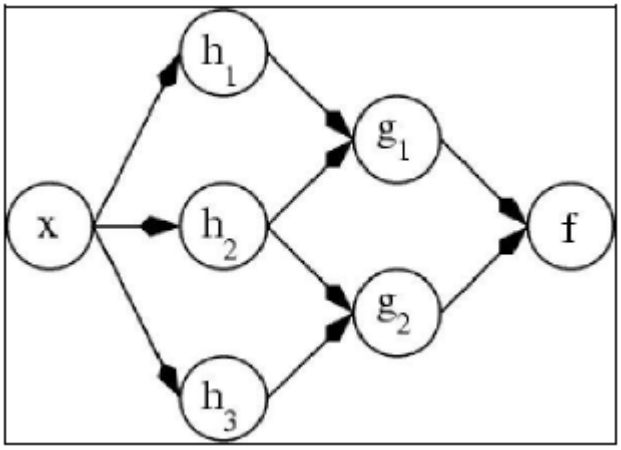
Setelah tiba di sel saraf pada lapisan output, nilai dari setiap lapisan tersembunyi dikalikan dengan suatu berat ($w_{kj}$), dan nilai yang dihasilkan setelah diberi beban atau dikalikan dengan berat ditambahkan bersama-sama menghasilkan suatu nilai kombinasi $v_j$. Jumlah nilai yang telah diberi beban ($v_j$) dimasukkan kedalam suatu fungsi transfer, yang menghasilkan suatu nilai $y_k$. Nilai y merupakan output dari jaringan.

Jika suatu analisis regresi ditunjukkan dengan suatu variabel target yang bersifat kontinu, maka terdapat sel saraf tunggal pada lapisan output, dan menghasilkan nilai y tunggal. Pada masalah klasifikasi dengan variabel target yang bersifat kategoris, terdapat sejumlah N sel saraf pada lapisan output menghasilkan nilai-nilai N, nilai dari setiap N kategori dari variabel target.

Diagram jaringan yang ditunjukkan di atas merupakan jaringan saraf perseptron yang terhubung sepenuhnya, dengan tiga lapisan, dan memberi nilai masukan ke arah depan. “Terhubung sepenuhnya” berarti bahwa output dari setiap input dan sel saraf tersembunyi terdistribusi ke semua sel saraf pada lapisan berikutnya. “Memberi nilai masukan ke arah depan” berarti bahwa nilai-nilai yang didistribusikan hanya bergerak dari input meuju lapisan tersembunyi, lalu menuju lapisan output; tidak ada nilai yang dimasukkan kembali ke lapisan sebelumnya (Jaringan Berulang memungkinkan nilai untuk dimasukkan dengan arah yang berlawanan)

Semua jaringan saraf memiliki sebuah lapisan input dan sebuah lapisan output, namun jumlah lapisan tersembunyi bisa bervariasi. Ini adalah diagram

jaringan perseptron dengan dua lapisan tersembunyi dan empat lapisan secara keseluruhan:




**Gambar 2.2 Perseptron Dengan 4 Lapisan**

Ketika terdapat lebih dari satu lapisan tersembunyi, output dari salah satu lapisan tersembunyi dimasukkan menuju lapisan tersembunyi berikutnya dan beban-beban dengan nilai yang berbeda dikenakan pada jumlah nilai keseluruhan yang menuju setiap lapisan.

### *2.1.2.1.2* Pelatihan Perseptron Lapis Ganda

Tujuan dari proses pelatihan ini adalah untuk menemukan rangkaian nilai berat/beban yang akan menyebabkan output dari jaringan saraf sesuai dengan nilai target yang sebenarnya sedekat mungkin. Ada beberapa hal yang terlibat dalam perancangan dan pelatihan jaringan perseptron lapis ganda (Sherrod, 2010) :

- pemilihan berapa banyak lapisan tersembunyi yang digunakan pada jaringan.
- Menentukan berapa banyak sel saraf yang digunakan pada setiap lapisan tersembunyi.
- Menemukan penyelesaian optimal secara global yang mencegah minimal lokal.

- Mengumpulkan penyelesaian tersebut sehingga menjadi satu penyelesaian optimal pada jangka waktu yang beralasan.
- Memvalidasi jaringan saraf pada pengujian overfitting.

#### 2.1.2.1.3 Menemukan Penyelesaian Optimal Secara Global

Jaringan saraf yang umum dapat memiliki sepasang beban yang bernilai ratusan yang nilainya harus ditemukan agar menghasilkan suatu penyelesaian optimal. Jika jaringan saraf merupakan model linear seperti regresi linear, akan sangat memberi peluang untuk menemukan serangkaian beban yang optimal. Tetapi, output dari suatu jaringan saraf sebagai suatu fungsi berat seringkali sangat bersifat tidak linear; hal ini membuat proses optimisasi menjadi hal yang rumit.

Metode optimisasi seperti penurunan tercuram (steepest descent) dan gradien konjugat (conjugate gradient) sangat cocok digunakan dalam menemukan minima lokal jika pencarian dimulai dalam suatu lembah yang dekat dengan minimum lokal. Metode tersebut tidak memiliki kemampuan untuk melihat gambar yang besar dan menemukan minimum global.

DTREG menggunakan algoritma (Sherrod, 2010) untuk memilih kisaran awal pada nilai beban mula-mula. Kemudian, algoritma tersebut menggunakan algoritma gradien konjugat untuk mengoptimalkan beratnya. Gradien konjugat biasanya menemukan berat optimal dengan cepat, tetapi tidak ada jaminan bahwa nilai berat yang telah ditemukan merupakan nilai optimal secara global. Jadi, hal tersebut berguna, yaitu memungkinkan DTREG untuk mencoba waktu optimisasi berganda dengan serangkaian nilai beban acak awal yang berbeda-beda.

#### 2.1.2.1.4 Gradien Konjugat

Dengan serangkaian nilai berat awal yang dipilih secara acak yang telah ditentukan, DTREG menggunakan algoritma gradien konjugat untuk mengoptimalkan nilai berat.

Sebagian besar algoritma latihan mengikuti siklus ini untuk memperbaiki nilai beban:

1. Memasukkan nilai penduga untuk suatu kasus pada jaringan menggunakan serangkaian berat yang bersifat sementara.
2. Menghitung selisih antara nilai target yang diduga dan nilai target yang sebenarnya untuk kasus tersebut. Ini adalah galat pendugaan.
3. Merata-ratakan informasi galat sepanjang serangkaian kasus pelatihan.
4. Menyebarkan galat dengan arah berlawanan / membelakangi pada jaringan dan menghitung gradien (vektor turunan) perubahan pada galat berdasarkan perubahan nilai berat.
5. Melakukan pengaturan berat untuk mengurangi kesalahan atau galat.

Setiap siklus disebut sebagai epos. Oleh karena informasi galat disebarkan dengan arah berlawanan pada jaringan, jenis metode pelatihan ini disebut penyebaran dengan arah berlawanan atau membelakangi (Penyebaran Ke Belakang / Backward Propagation). Algoritma pelatihan penyebaran ke belakang dijelaskan pertama kali oleh Rumelhart dan McClelland pada tahun 1986 itu merupakan metode praktis pertama pada pelatihan jaringan saraf.

Prosedur yang asli menggunakan algoritma penurunan gradien untuk mengatur berat menuju konvergensi / pendekatan menggunakan gradien. Akibat sejarah tersebut, istilah “penyebaran ke belakang / backpropagation” atau “backprop” seringkali digunakan untuk melambangkan algoritma pelatihan jaringan saraf menggunakan penurunan gradien sebagai algoritma inti. Hal tersebut sedikit disayangkan karena penyebaran ke belakang terhadap informasi galat pada jaringan digunakan hampir pada semua algoritma pelatihan, beberapa diantaranya jauh lebih baik daripada metode penurunan gradien.

Penyebaran ke belakang yang menggunakan penurunan gradien seringkali berkonvergensi dengan sangat lambat atau tidak berkonvergensi sama sekali. Pada permasalahan skala besar, keberhasilannya tergantung dari parameter tingkat pembelajaran yang ditentukan oleh penggunanya dan parameter momentum atau daya dorong. Tidak ada cara yang otomatis dalam memilih parameter-parameter ini, dan jika nilai yang salah dispesifikasi, konvergensinya bisa sangat lambat, atau bisa jadi tidak berkonvergensi sama sekali. Sementara metode penyebaran ke belakang dengan penurunan gradien masih digunakan pada banyak program jaringan saraf, metode tersebut tidak lagi dianggap sebagai algoritma tercepat atau terbaik.

DTREG menggunakan algoritma gradien konjugat untuk mengatur nilai berat menggunakan gradien selama penyebaran ke belakang terhadap galat di sepanjang jaringan. Jika dibandingkan dengan penurunan gradien, algoritma gradien konjugat mengambil jalur yang lebih langsung menuju serangkaian nilai berat optimal. Biasanya, gradien konjugat bersifat sangat cepat dan lebih mantap daripada penurunan gradien. Gradien konjugat juga tidak memerlukan pengguna untuk menentukan tingkat pembelajaran dan parameter momentum.

Algoritma gradien konjugat tradisional menggunakan gradien untuk memperhitungkan arah pencarian. Algoritma tersebut juga menggunakan sebuah algoritma garis pencarian seperti Metode Brent untuk menemukan ukuran langkah optimal di sepanjang sebuah garis pada arah pencarian. Garis pencarian mengatasi perlunya untuk menghitung matriks Hessian pada turunan kedua, tapi algoritma tersebut memerlukan perhitungan galat pada titik-titik yang banyak di sepanjang garis. Algoritma gradien konjugat dengan garis pencarian (CGL) telah berhasil digunakan pada banyak program jaringan saraf, dan dianggap sebagai metode terbaik yang belum pernah ditemukan.

DTREG memberikan algoritma gradien konjugat tradisional dengan garis pencarian, namun juga menawarkan algoritma terbaru, Gradien Konjugat Berskala (Sherrod, 2010). Algoritma gradien konjugat berskala menggunakan pendekatan

numeris untuk turunan kedua (matriks Hessian), namun mengatasi ketidakstabilan dengan menggabungkan pendekatan daerah kepercayaan model dari algoritma Levenberg-Marquardt dengan pendekatan gradien konjugat. Hal ini memungkinkan gradien konjugat berskala untuk menghitung ukuran langkah optimal pada arah pencarian tanpa perlu menampilkan garis pencarian terpogram yang mahal yang digunakan pada algoritma gradien konjugasi tradisional. Tentu saja, ada biaya yang diperlukan dalam menduga turunan kedua.

Pengujian yang ditunjukkan oleh Moller menunjukkan algoritma gradien konjugat berskala yang berkonvergensi sampai dua kali kecepatan gradien konjugat tradisional dan mencapai 20 kali kecepatan metode penyebaran ke belakang menggunakan penurunan gradien. Pengujian Moller juga menunjukkan bahwa gradien konjugat berskala jarang mengalami kegagalan untuk berkonvergensi dibandingkan gradien konjugat tradisional atau metode penyebaran ke belakang menggunakan penurunan gradien (Sherrod, 2010).

#### 2.1.2.2 *Radial Basis Function (RBF)*

Jaringan RBF memiliki tiga lapis , seperti pada Gambar 2.3:

**a. Lapisan Input**

Terdapat satu sel saraf pada lapisan input untuk setiap variabel penduga. Sel saraf atau neuron input (atau pengolahan sebelum lapisan input) menstandarkan kisaran nilai dengan mengurangkan median dan dibagi dengan jangkauan antarkuartil. Lalu, sel saraf input memasukkan nilai-nilai kepada setiap sel saraf pada lapisan tersembunyi.

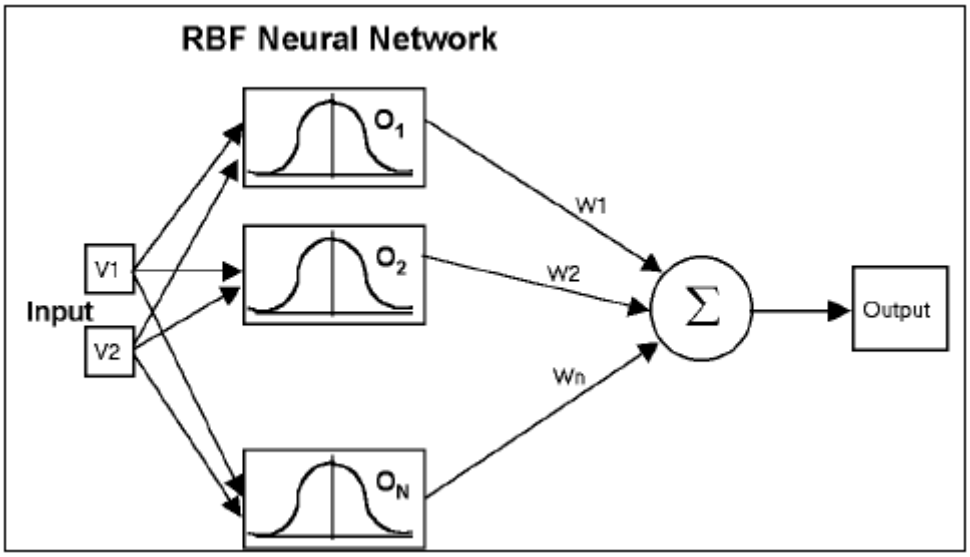
**b. Lapisan Tersembunyi**

Lapisan ini memiliki sejumlah neuron dengan jumlah yang bervariasi (jumlah optimalnya ditentukan melalui proses pelatihan). Setiap neuron terdiri dari fungsi basis radial yang berada di tengah pada suatu titik dengan dimensi sebanyak jumlah variabel penduga. Sebaran (radius) dari fungsi RBF dapat berbeda untuk setiap dimensinya. Nilai tengah dan sebaran ditentukan melalui proses pelatihan. Jika ditunjukkan dengan vektor x dari nilai input pada

lapisan input, neuron tersembunyi menghitung jarak Euclidean kasus pengujian dari titik pusat neuron dan kemudian menerapkan fungsi inti RBF terhadap jarak tersebut dengan menggunakan nilai-nilai sebaran. Nilai yang dihasilkan dilewatkan melalui lapisan penjumlahan.

**c. Lapisan Penjumlahan**

Nilai yang berasal dari suatu neuron pada lapisan tersembunyi dikalikan dengan sebuah berat yang dikenakan pada neuron dan dilewatkan menuju lapisan penjumlahan yang menambahkan nilai-nilai yang sudah diberi beban (sudah dikalikan dengan berat) dan menyediakan hasil penjumlahan ini sebagai output atau hasil keluaran dari jaringan.

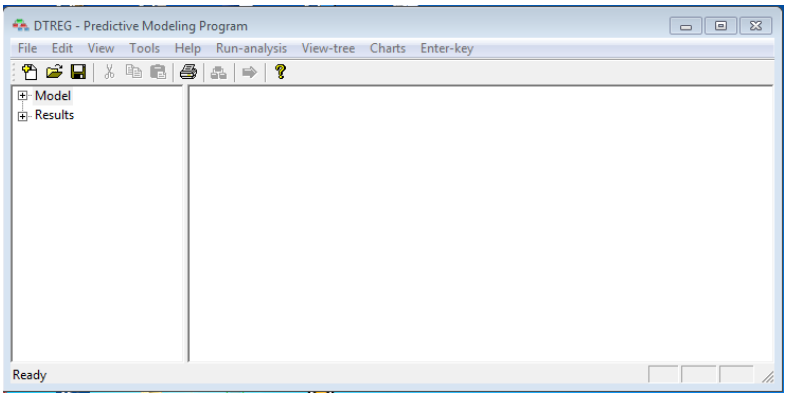



**Gambar 2.3. Jaringan Saraf *Radial Basis Function (RBF)***

### 2.1.3 DTReg

DTReg adalah sebuah program analisis statistik yang kuat yang menghasilkan pohon keputusan klasifikasi dan regresi yang model data dan dapat digunakan untuk memprediksi nilai-nilai (www.dtreg.com). DTReg adalah aplikasi yang kuat yang diinstal dengan mudah pada sistem Windows. DTReg membaca file Comma Separated Value (CSV) data yang mudah dibuat dari hampir semua sumber data. Setelah Anda membuat file data Anda, hanya feed ke DTReg, dan biarkan DTReg melakukan semua pekerjaan membuat pohon keputusan dan pemangkasan untuk ukuran optimal. Bahkan analisis kompleks dapat diatur dalam beberapa menit.

DTREG menggunakan V-kali lipat cross-validasi untuk menentukan ukuran pohon yang optimal. Prosedur ini menghindari masalah "*overfitting*" di mana pohon yang dihasilkan sesuai dengan data pelatihan dengan baik tetapi tidak memberikan prediksi yang akurat dari data baru. DTReg menggunakan teknik canggih yang melibatkan "splitter pengganti" untuk menangani kasus dengan nilai-nilai yang hilang. Hal ini memungkinkan kasus dengan beberapa nilai yang tersedia dan beberapa nilai yang hilang untuk dimanfaatkan semaksimal ketika membangun model. Hal ini juga memungkinkan DTReg untuk memprediksi nilai-nilai kasus yang nilai hilang. DTReg dapat menampilkan pohon keputusan yang dihasilkan pada layar, menulis ke. Jpg atau png file disk atau. Mencetaknya. Ketika dicetak, DTREG menggunakan teknik canggih untuk pagenating pohon yang melintasi beberapa halaman.




**Gambar 2.4 Tampilan Awal Aplikasi DTReg**

Berikut beberapa hal yang harus dipersiapkan dalam penggunan DTReg :

a. Input Variabel

Ketika membangun sebuah model (bukan time series) normal, input harus terdiri dari nilai-nilai untuk satu variabel target dan satu atau lebih variabel prediktor. Ketika membangun sebuah model time series, masukan dapat terdiri dari nilai-nilai hanya satu variabel - variabel sasaran yang nilainya harus dimodelkan dan diperkirakan.

b. Lag Variabel

Sebuah variabel lag memiliki nilai dari beberapa variabel lain seperti terjadi beberapa jumlah periode sebelumnya. Sebagai contoh, di sini adalah seperangkat nilai-nilai untuk variabel Y, lag pertama dan lag kedua:

| Y | Y_Lag_1 | Y_Lag_2 |
|---|---|---|
| 3 | ? | ? |
| 5 | 3 | ? |
| 8 | 5 | 3 |
| 6 | 8 | 5 |

Perhatikan bahwa lag nilai untuk pengamatan sebelum awal seri tidak diketahui.

DTREG menyediakan generasi otomatis variabel lag. Pada halaman Time Series Properti dalam DTReg anda dapat memilih variabel yang memiliki variabel lag dihasilkan dan seberapa jauh kembali nilai-nilai lag yang menjalankan. Anda juga dapat membuat variabel untuk rata-rata bergerak, tren linear dan lereng pengamatan sebelumnya. Berikut adalah contoh dari Page Properti Variabel menunjukkan variabel lag dihasilkan untuk Penumpang:

Variables

| Variable | Target | Predictor | Weight | Categorical | Character |
|---|---|---|---|---|---|
| Passengers | ☒ | ☐ | ☐ | ☐ | ☐ |
| Passengers_Lag_1 | ☐ | ☒ | ☐ | ☐ | ☐ |
| Passengers_Lag_2 | ☐ | ☒ | ☐ | ☐ | ☐ |
| Passengers_Delta_2 | ☐ | ☒ | ☐ | ☐ | ☐ |
| Passengers_LTrend_2 | ☐ | ☒ | ☐ | ☐ | ☐ |
| Passengers_Slope_2 | ☐ | ☒ | ☐ | ☐ | ☐ |
| Passengers_SMA_2 | ☐ | ☒ | ☐ | ☐ | ☐ |
| Passengers_LMA_2 | ☐ | ☒ | ☐ | ☐ | ☐ |
| Passengers_EMA_2 | ☐ | ☒ | ☐ | ☐ | ☐ |

Pada layar ini, Anda dapat memilih yang dihasilkan variabel yang ingin Anda gunakan sebagai prediktor untuk model. Meskipun tergoda untuk menghasilkan banyak variabel dan menggunakan semua dari mereka dalam

model, model kadang-kadang lebih baik dapat dihasilkan dengan menggunakan nilai lag hanya merupakan kelipatan dari periode siklus seri

c. intervensi variabel

Suatu peristiwa luar biasa yang terjadi selama time series dikenal sebagai intervensi. Contoh intervensi adalah perubahan suku bunga, suatu aksi teroris atau pemogokan buruh. Seperti peristiwa mengacaukan time series dengan cara yang tidak dapat dijelaskan dengan sebelumnya (lag) pengamatan.

DTREG memungkinkan Anda untuk menentukan variabel prediktor tambahan selain variabel target. Anda bisa memiliki variabel untuk suku bunga, produk domestik bruto, inflasi, dll Anda juga dapat memberikan variabel dengan nilai 0 untuk semua baris sampai dengan awal pemogokan buruh, maka 1 untuk baris selama pemogokan, kemudian menurun nilai setelah akhir pemogokan. Variabel ini disebut variabel intervensi, mereka ditetapkan dan digunakan sebagai variabel prediktor biasa. DTREG dapat menghasilkan nilai lag untuk variabel intervensi seperti untuk variabel target.

d. *Trend removal* dan *stationary time series*

Sebuah time series dikatakan stasioner jika kedua artinya (nilai tentang yang berosilasi), dan varians nya (amplitudo) tetap konstan melalui waktu. Klasik Box-Jenkins ARMA model hanya bekerja memuaskan dengan time series stasioner, jadi bagi jenis model adalah penting untuk melakukan transformasi pada seri untuk membuatnya diam.

Model yang dikembangkan oleh DTREG kurang sensitif terhadap non-stasioner seri waktu model ARMA, tetapi mereka biasanya mendapatkan keuntungan dengan membuat seri stasioner sebelum membangun model. DTREG termasuk fasilitas untuk menghapus tren dari seri waktu dan menyesuaikan amplitudo.

e. Memilih jenis model untuk *time series*

DTREG memungkinkan Anda untuk menggunakan jenis berikut model untuk time series : *(1) Decision tree, (2) TreeBoost, (3) Multilayer perceptron*

*neural network, (4) General regression neural network (GRNN), (5) RBF neural network, (6) Cascade correlation network, (7) Support vector machine (SVM), (8) Gene expression programming, (9) GMDH neural networks.*

Percobaan telah menunjukkan bahwa pohon keputusan biasanya tidak bekerja dengan baik karena mereka melakukan pekerjaan yang buruk memprediksi nilai berkelanjutan. *Gene expression programming (GEP)* merupakan metode yang baik untuk seri waktu karena fungsi yang dihasilkan sangat umum, dan mereka dapat menjelaskan tren dan perubahan varians. *General regression neural networks (GRNN)* and *GMDH neural networks* juga tampil sangat baik dalam tes. *Multilayer perceptrons* kadang-kadang bekerja sangat baik, tetapi mereka lebih temperamental untuk melatih. Jadi rekomendasi terbaik adalah untuk selalu berusaha GEP and GRNN, dan kemudian mencoba jenis lain dari model jika Anda punya waktu. Jika Anda menggunakan model GEP, yang terbaik adalah untuk mengaktifkan fitur untuk memungkinkan untuk berkembang konstanta numerik.

## 2.2 Tinjauan Studi

Pada penelitian tesis ini, peneliti melakukan tinjauan studi dengan beberapa jurnal yang berhubungan dengan tema penelitian yang dipilih.

1. *Performance Analysis of stock Price Prediction Using Artificial Neural Network*

   Penelitian yang dilakukan oleh Sureshkumar (2012), dalam penelitian ini membahas tentang penerapan model *multilayer perceptron* dengan menggunakan dari alat memprediksi *Neural Networks.* Hasil dari analisis menunjukkan bahwa *Neural Networks* menawarkan kemampuan untuk memprediksi saham harga lebih akurat daripada alat yang ada lainnya dan teknik. Analisis ini dapat digunakan untuk mengurangi persentase kesalahan dalam memprediksi harga saham masa depan. Hal ini meningkatkan peluang bagi para investor untuk memprediksi harga lebih akurat dengan mengurangi persentase kesalahan dan karenanya meningkatkan keuntungan mereka di pasar saham. Perbedaan dengan penelitian yang sedang peneliti

ambil adalah peneliti mencoba mengkomparasikan dengan model *Radial Basis Function (RBF)* untuk mengukur keakurasian yang lebih tinggi.

2. *Neural Network And Investor Sentiment Measures For Stock Market Trend Prediction.*

   Penelitian ini dilakukan oleh Salim Lahmiri (2011), dalam penelitian ini membahas tentang arsitektur jaringan syaraf digunakan untuk memprediksi masa depan tren dari NYSE dan tiga saham dari teknologi sektor. Jaringan saraf yang digunakan adalah Back-Propagation jaringan saraf (BPNN) dan Probabilistik Jaringan Syaraf (PNN), algoritma Genetik (GA) digunakan untuk mengoptimalkan topologi dari BPNN tersebut dan tindakan sentimen investor tindakan juga digunakan sebagai masukan untuk jaringan saraf. untuk mengidentifikasi langkah-langkah secara statistik berhubungan dengan masing-masing perusahaan dan ekuitas pasar indeks. Perbedaan dengan penelitian yang sedang peneliti ambil adalah peneliti belum mengkomparasikan mengkomparasikan dengan model *Radial Basis Function (RBF)* untuk membandingkan keakurasian harga saham.

3. *Accuracy Driven Artificial Neural Networks in Stock Market Prediction.*

   Penelitian ini dilakukan oleh Selvan Simon dan Arun Raoot (2012), dimana pada penelitian ini Jaringan Syaraf Tiruan memainkan peran penting di *Stock Market Prediction*. Model *Neural Network* telah mengungguli model tradisional lainnya. *Multilayer Layer Perceptron* dengan pelatihan BP ditemukan menjadi *Neural Network* yang paling banyak digunakan model di prediksi harga saham. Ada variasi dalam model JST. Sangat penting untuk memilih model JST yang tepat mengingat masalah target. Berbagai teknik statistik dapat digunakan untuk sebelum memproses data untuk meningkatkan kinerja. Juga, algoritma khusus dapat dikombinasikan dengan Jaringan Syaraf Tiruan untuk meningkatkan akurasi. Yang dapat diadaptasi dari penelitian tersebut adalah bahwa Jaringan syaraf tiruan dengan model *Multilayer Perceptron (MLP)* adalah pendekatan yang paling populer

dalam memprediksi harga saham yang akan datang dan peneliti akan mengkomparasikan dengan model *Radial Basis Function (RBF).*

*4.* Prediksi Harga Saham Dengan Menggunakan Algoritma *Hybrid Neural Network*

Penelitian yang dilakukan oleh Chairisni Lubis, et all (2005), dalam penelitian ini membahas tentang prediksi harga saham di masa datang yang ditentukan dari pembelajaran pola fluktuasi harga saham tersebut di masa lampau. Pada penelitian ini, akan diteliti keakuratan dua algoritma Neural Network yang sering digunakan dalam memprediksi harga saham dan satu algoritma *hybrid* yang ingin diketahui kelebihannya dibandingkan kedua algoritma tersebut. Algoritma pertama adalah algoritma *Backpropagation Network*, dimana algoritma ini mempunyai keakuratan prediksi yang tinggi bila data pembelajarannya relatif stabil. Algoritma kedua adalah algoritma *Self Organizing Maps* Kohonen, dimana algoritma ini mempunyai keakuratan prediksi yang cukup tinggi walaupun pola data pembelajarannya bersifat fluktuatif. Algoritma ketiga adalah algoritma *Hybrid*, dimana algoritma ini merupakan kombinasi dari kedua algoritma diatas, dan diharapkan dapat memberikan keakuratan prediksi yang sangat tinggi, baik untuk data pembelajaran yang bersifat stabil maupun fluktuatif. Yang dapat diadaptasi dari penelitian tersebut adalah peneliti dalam menentukan harga saham yang akan datang ditentukan dari pembelajaran pola fluktuasi harga saham pada masa lampau.

Berikut Tabel Perbandingan Tinjauan Studi :

## Tabel 2.1. Perbandingan Tinjauan Studi

| Nama Penulis | Tahun | Resume | Perbedaan dengan Penelitian ini |
|---|---|---|---|
| Sureshkumar | 2012 | Penelitian ini hanya menggunakan model *Multilayer Perceptron* | Penelitian ini menggunakan dua model yaitu : *Multilayer Perceptron & Radial Basis Function* |
| Salim Lahmiri | 2011 | Penelitian ini menggunakan variabel sentimen investor dan harga saham | Penelitian menggunakan variabel historical harga saham |
| Selvan Simon dan Arun Raoot | 2012 | Penelitian ini menggunakan *Multilayer Perceptron* karena metode paling populer | Penelitian ini mengadaptasi menggunakan *Multilayer Perceptron* sebagai metode paling populer |
| Chairisni Lubis, et all | 2005 | Penelitian ini menggunakan pembelajaran pola fluktuasi harga saham pada masa lampau. | Penelitian ini mengadaptasi pola fluktuasi historical harga saham |

## 2.3 Tinjauan Organisasi / Obyek Penelitian

PT. Agra Argo Lestari, adalah perusahaan yang bergerak dibidang perkebunan khususnya produk minyak kelapa sawit. Ini dimulai sebagai sebuah unit usaha PT Astra International Tbk untuk mengembangkan 2.000 hektar perkebunan singkong. Kemudian berubah menjadi perkebunan karet setelah melihat munculnya permintaan karet. Kemudian, mengingat bisnis yang menjanjikan minyak sawit, PT Astra International Tbk berkelana ke industri kelapa sawit.

Pada tahun 1984, PT Astra International Tbk mengakuisisi PT Tunggal Perkasa Plantations, yang telah memiliki 15.000 hektar perkebunan kelapa sawit di propinsi Riau. Keputusan bisnis yang terbukti tepat waktu dan tepat sebagai bisnis kelapa sawit telah tumbuh kokoh, memberikan dampak positif terhadap unit bisnis Perusahaan. Akhirnya, PT Astra International Tbk memutuskan untuk berputar unit bisnis ke anak perusahaan kelapa sawit baru dengan nama PT Suryaraya Cakrawala baru pada tanggal 3 Oktober 1988. Sebagai salah satu anak perusahaan PT Astra International Tbk, nama PT Suryaraya Cakrawala diubah menjadi PT Astra Agro Niaga pada tahun 1989.

Didorong oleh tumbuhnya bisnis kelapa sawit, PT Astra Agro Niaga melakukan merger dengan PT Suryaraya Bahtera tahun 1997 dan berubah nama menjadi PT Astra Agro Lestari. Nama PT Astra Agro Lestari yang dikenal publik dan berkelanjutan sampai mencapai usia 23 tahun di 2011.

PT Astra Agro Lestari mencatatkan sahamnya untuk pertama kalinya pada tanggal 9 Desember 1997 tentang Bursa Efek Jakarta dan Bursa Efek Surabaya, yang keduanya telah dilebur ke dalam Bursa Efek Indonesia (BEI). Selama Penawaran Umum Perdana (IPO), Perusahaan menawarkan 125.800.000 saham kepada masyarakat dengan harga Rp 1.550 per saham. Pada akhir 2011, perdagangan saham Perusahaan di BEI ditutup pada harga Rp 21.700 per saham. Setelah dalam operasi selama 31 tahun, PT Astra Agro Lestari Tbk kini memiliki total luas 266.706 hektar minyak kelapa sawit, yang terdiri dari kebun inti dan

plasma (perkebunan plasma) di Sumatera, Kalimantan dan Sulawesi, dengan usia rata-rata 14 tahun. Sampai akhir 2011, Perusahaan telah dipekerjakan 26.473 orang.

PT Astra Agro Lestari Tbk telah berhasil mengubah bisnis kelapa sawit menjadi kesempatan emas, karena telah menunjukkan melalui pertumbuhan yang kuat, sekaligus menjaga optimisme untuk mempertahankan prestasi di masa depan, yang menjanjikan bisnis yang lebih berkembang dari kelapa sawit bagi Perusahaan .

## 2.4 Kerangka Pemikiran

Penelitian ini di lakukan untuk mengamati dan menganalisa pola harga saham dengan memanfaatkan data harga saham yang lalu dengan menerapkan menerapkan *Neural Network.* Berikut adalah kerangka pemikiran dalam penelitian yang dibuat:

| Flow | Keterangan |
|---|---|
| Permasalahan | 1. Bagaimana memanfaatkan data yang ada, agar diperoleh suatu manfaat bagi investor?<br>4. Apakah pendekatan neural network dapat digunakan untuk memprediksi harga saham PT. Astra Argo Lestari yang akan datang?<br>5. Apakah dengan menggunakan neural network dapat digunakan untuk merekomendasikan investor tentang saham yang akan datang? |
| Pendekatan | *Neural Network* |
| Pengembangan | *DTReg* |
| Penerapan | PT . Astra Argo Lestari, Tbk |
| Pengujian | *Multilayer Perceptron (MLP) dan Radial Basis Function (RBF)* |
| Hasil | *Neural Network* dapat bermanfaat dalam menentukan harga saham yang akan datang pada PT. Astra Agro Lestari yang dapat dimanfaatkan oleh para investor. |

**Gambar 2.5 Kerangka Pemikiran**

# BAB 3

# METODE PENELITIAN

## 3.1 Perancangan Penelitian

Pada bab ini, percobaan dengan menggunakan data finansial yang sebenarnya dilakukan untuk menunjukkan kemapuan neural network dalam memprediksi data finansial. Dua jenis *Neural Network* yang berbeda, yaitu *Multilayer Perceptron (MLP)* dan *Radial Basis Function (RBF),* penelitian ini dilakukan untuk memperoleh model terbaik.

Berbagai macam kombinasi pra-pengolahan data, arsitektur jaringan, fungsi aktivasi, dan algoritma pelatihan dibandingkan pada setiap model *Neural Network.*

Penelitian ini dilakukan pada periode antara tanggal 1 Januari 2006 sampai 30 Desember 2011. Peneliti menggunakan data dari PT. Astra Agro Lestari Tbk yang merupakan perusahaan perkebunan khususnya kelapa sawit di Indonesia. Data tersebut terdiri dari harga-harga saham yang mengalami penutupan setiap hari yang diperoleh dari website finansial yahoo (http://finance.yahoo.com/).

## 3.2 Metode Penelitian

Penelitian ini disusun berdasarkan metodologi yang secara garis besar meliputi kegiatan berikut:

1. Identifikasi masalah dan analisa kebutuhan.

   Identifikasi masalah yang berkaitan dengan pemilihan variabel *input* dan *output*. Setelah diidentifikasi kemudian merumuskan masalah yang selanjutnya dilakukan menganalisa kebutuhan untuk memecahkan masalah tersebut.

2. Pengumpulan data

Pada tahap ini, pengumpulan data yang dibutuhkan dapat dilakukan dengan beberapa cara seperti studi literatur, observasi, dan melakukan tanya jawab.

Data sampel dibagi menjadi sub – sub bagian . Sub-sub bagian tersebut adalah data untuk *training* model, data untuk validasi model, dan terakhir data untuk *testing* keakuratan model.

Porsi terbesar adalah data untuk *training*, dimana data ini digunakan untuk mendefinisikan parameter model (melatih kemampuan model). Data untuk validasi digunakan dengan tujuan untuk menguji kemampuan model selama proses pembentukannya. Sedangkan data untuk *testing* digunakan untuk memeriksa keakuratan peramalan model jika diuji pada data di luar sampel.

3. Eksperimen

   Dalam eksperimen ini dilakukan penentuan metode-metode yang akan digunakan, kemudian data *training* diaplikasikan ke dalam metode tersebut, setelah itu dilakukan pengujian menggunakan DTReg.

   6. Implementasi

      Menerapkan model yang dihasilkan ke dalam sistem untuk menganalisa data historis harga saham.

   7. Evaluasi

      Evaluasi dilakukan untuk mengetahui harga saham yang akurat dalam pendugaan harga saham.

   8. Pembuatan tesis

      Setelah tahapan penelitian selesai kemudian dituangkan dalam bentuk laporan yang berbentuk tesis.

## 3.3 Perangkat Penelitian

Spesifikasi perangkat keras dan perangkat lunak yang digunakan untuk mendukung penelitian adalah:

### *3.3.1 Hardware*

Dalam menjalankan sebuah penelitian tentunya membutuhkan spesifikasi minimum dari perangkat atau *hardware* dalam menjalankan suatu aplikasi. Spesifikasi perangkat keras minimum yang dibutuhkan adalah system operasi Windows XP Service Pack 2, processor Intel® Pentium IV, Harddisk 250Gb dan RAM 512MB, serta monitor SVGA.

### *3.3.2 Software*

Penggunaan *Neural Network* yang digunakan dalam penelitian dalam pendugaan harga saham tidak dapat dibuktikan secara langsung keakuratannya tanpa adanya sebuah *software* yang digunakan dalam penelitian tersebut. *DTReg* dengan *Graphical User Interface(GUI)* adalah sebuah *software editor* yang digunakan dalam pengembangan penelitian ini.

Kebutuhan akan *hardware* dan *software* dapat dilihat secara spesifikasi pada tabel 3.1.

**Tabel 3.1. Spesifikasi Perangkat keras dan Perangkat Lunak**

| Spesifikasi kebutuhan | Tipe |
|---|---|
| Prosesor | Pentium IV |
| Memori | 512 MB |
| Monitor | SVGA |
| Sistem operasi | Windows XP |
| Aplikasi | DTReg |

## 3.4 Jadwal Penelitian

Penelitian ini dilakukan dalam beberapa tahapan, berikut waktu yang diperlukan dalam melakukan pengerjaan dalam melakukan penelitian yaitu:

1. Identifikasi masalah dan analisa kebutuhan.

   Dalam melakukan penelitian diperlukan waktu untuk melakukan identifikasi masalah serta menganalisa kebutuhan diperlukan waktu tiga minggu.

2. Pengumpulan data

   Pada tahap ini, pengumpulan data yang dibutuhkan dapat dilakukan dengan beberapa cara seperti studi literatur, observasi, dan melakukan tanya jawab. Waktu yang diperlukan dalam melakukan pengumpulan data adalah selama tiga minggu.

3. Eksperimen

   Dalam eksperimen ini dilakukan penentuan metode-metode yang akan digunakan, kemudian data *training* diaplikasikan ke dalam metode tersebut, eksperimen diperlukan waktu selama tiga minggu, yang dapat dimulai pada minggu terakhir pada saat pengumpulan data.

9. Implementasi

   Menerapkan model yang dihasilkan ke dalam sistem untuk menganalisa data historis harga saham.Waktu yang diperlukan adalah selama tiga minggu, yang dapat dimulai pada minggu terakhir pada tahap eksperimen.

10. Evaluasi

    Evaluasi dilakukan untuk mengetahui harga saham yang akurat dalam pendugaan harga saham.Waktu yang diperlukan pada tahap evaluasi adalah selama tiga minggu, yang dapat dimulai pada minggu terakhir pada tahap implementasi.

11. Pembuatan tesis

    Setelah tahapan penelitian selesai kemudian dituangkan dalam bentuk laporan yang berbentuk tesis. Pembuatan tesis ini memerlukan waktu selama empat minggu, yang dimulai sejak minggu kedua pada tahap melakukan evaluasi.

Dari penjelasan diatas dapat digambarkan dalam tabel sebagai berikut:

**Tabel 3.2 Jadwal Penelitian**

| No. | Jenis Kegiatan | Minggu Ke | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 1 | Identifikasi masalah dan analisa kebutuhan | | | | | | | | | | | | |
| 2 | Pengumpulan data | | | | | | | | | | | | |
| 3 | Eksperimen | | | | | | | | | | | | |
| 4 | Implementasi | | | | | | | | | | | | |
| 5 | Evaluasi | | | | | | | | | | | | |
| 6 | Pembuatan Tesis | | | | | | | | | | | | |

# BAB 4

# HASIL ANALISA DAN PEMBAHASAN

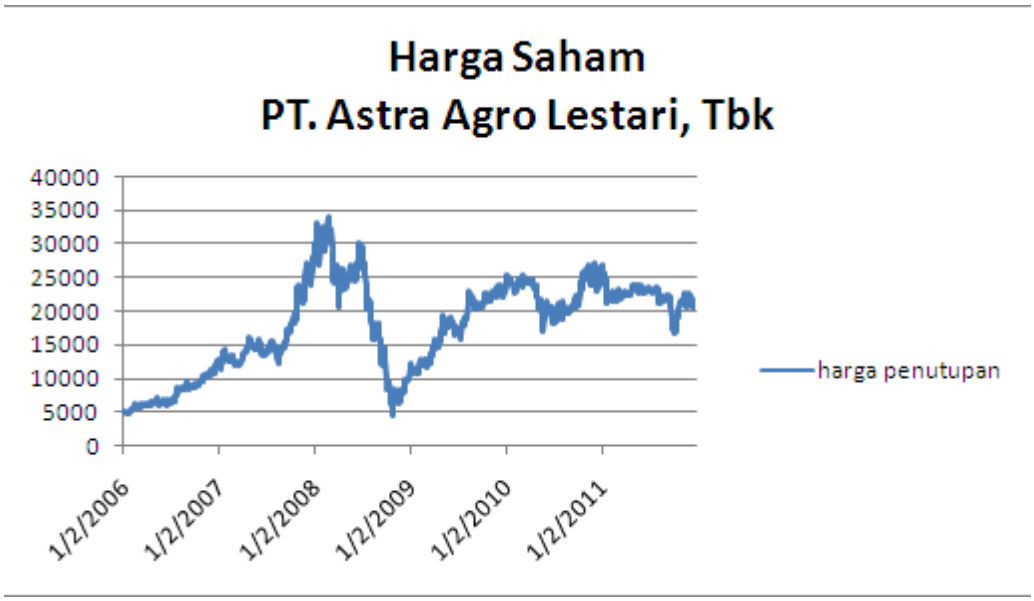
## 4.1 Hasil Penelitian

Pada bab ini, percobaan dengan menggunakan data finansial yang sebenarnya dilakukan untuk menunjukkan kemmapuan jaringan neural dalam memprediksi data finansial. Dua Jenis jaringan neural yang berbeda, yaitu *Multilayer Perceptron (MLP)* dan *Radial Basis Function (RBF)*, dilakukan untuk memperoleh model terbaik.

Berbagai macam kombinasi pra-pengolahan data, topologis/arsitektur jaringan, fungsi aktivasi, dan algoritma pelatihan dibandingkan pada setiap model *Neural Network* .

### 4.1.1 Data

Penelitian ini dilakukan pada periode antara tanggal 1 Januari 2006 sampai 30 Desember 2011. Sehingga, rangkaian data berisi sebanyak 1459 titik data dalam seri waktu. Kami menggunakan data dari PT. Astra Agro Lestari, Tbk. yang merupakan perusahaan perkebunan kelapa sawit di Indonesia. Data tersebut terdiri dari harga-harga saham yang mengalami penutupan setiap hari yang diperoleh dari website finansial yahoo (http://finance.yahoo.com/). Waktu dipilih dari tanggal 1 Januari 2006 s.d. 30 Desember 2011 karena saham minyak mentah pada tahun 2008 mengalami krisis moneter dunia yang mengakibatkan anjloknya harga saham minyak mentah. Seri data mentah ditunjukkan pada gambar 4.1 dan tabel 4.1. menunjukkan statistik deskriptif dari harga-harga penutupan dari data yang digunakan pada penelitian ini.




**Gambar 4.1 Harga Saham PT. Astra Agro Lestari, Tbk**

**Tabel 4.1 Statistik Deskriptif Data Mentah**

VAR00001

| | | |
|---|---|---|
| N | Valid | 2503 |
| | Missing | 0 |
| Mean | | 11388.0144 |
| Median | | 8750.0000 |
| Std. Deviation | | 9241.92546 |
| Skewness | | .387 |
| Std. Error of Skewness | | .049 |
| Kurtosis | | -1.361 |
| Std. Error of Kurtosis | | .098 |
| Minimum | | 900.00 |
| Maximum | | 34000.00 |

### 4.1.2 Pemilihan Variabel Input

Langkah pertama adalah pemilihan variabel input. Pemilihan variabel input pada model jaringan neural merupakan faktor kritis terhadap kinerja jaringan neural. Pada permasalahan pendugaan seri waktu, input biasanya merupakan pengamatan masa lalu terhadap serangkaian data dan outputnya adalah nilai masa depan. Oleh karena penelitian ini menggunakan model seri waktu univariat, variabel input yang dipilih pada model ini merupakan pengamatan yang tertinggal terhadap seri waktu yang akan diprediksi. *Neural Network* menunjukkan pemetaan fungsi berikut ini:

$$y_{t+1} = f(y_t, y_{t-1}, \ldots, y_{t-p}) \qquad [1]$$

Dimana Yt-1 adalah nilai Y untuk pengamatan sebelumnya, Yt-2 adalah nilai dua pengamatan lalu, dll, dan et mewakili suara yang tidak mengikuti pola yang dapat diprediksi (ini disebut kejutan acak). Nilai variabel yang terjadi sebelum pengamatan saat ini disebut nilai lag. Jika deret waktu mengikuti pola yang berulang, maka nilai Yt biasanya sangat berhubungan dengan Yt-siklus di mana siklus adalah jumlah pengamatan dalam siklus teratur. Misalnya, pengamatan bulanan dengan siklus tahunan sering dapat dimodelkan oleh Yt = f (Yt-12).

Tujuan membangun sebuah model time series adalah sama dengan tujuan untuk jenis lain dari model prediksi yang adalah untuk menciptakan sebuah model sedemikian rupa sehingga kesalahan antara nilai prediksi variabel target dan nilai aktual adalah sekecil mungkin. Perbedaan utama antara model time series dan jenis lain dari model ini adalah bahwa lag nilai dari variabel target yang digunakan sebagai variabel prediktor, sedangkan model tradisional menggunakan variabel lain sebagai prediktor, dan konsep nilai lag tidak berlaku karena pengamatan tidak mewakili urutan kronologis.

### 4.1.3 Pra-pengolahan Data

Pada beberapa literatur, untuk mengurangi jumlah data yang abnormal dan untuk meningkatkan kecepatan konverjensi pada jaringan, sebelum datanya digunakan dalam pelatihan, maka seringkali diberi skala/dinormalkan terlebih dahulu. Misalnya, harga-harga stok harian dapat dibagi dengan harga historis maksimal . Meski begitu, pada penelitian ini, pemberian skala atau normalisasi data tidak diperlukan.

### 4.1.4 Pembagian Kelompok Data

Kelompok data dibagi kedalam kelompok pelatihan dan kelompok validasi. Kelompok pelatihan dimulai dari permulaan data sampai tanggal 1 Januari 2006 dan sisanya digunakan untuk validasi. Hal ini berdasarkan petunjuk dari Kaastra bahwa kelompok data pelatihan sebaiknya merupakan kelompok data yang paling banyak jumlahnya dan digunakan oleh jaringan neural untuk mempelajari pola yang terdapat pada data, sedangkan kelompok data validasi sebaiknya berasal dari pengamatan terkini yang berdekatan. Kelompok data validasi digunakan untuk memilih parameter jaringan yang optimal pada model. Pelatihan dan validasi terjadi secara bersamaan, dan dua kelompok data digunakan untuk menyelidiki nilai parameter pada model.

## 4.2. Pengujian Model

### 4.2.1 Model Perseptron Lapis Ganda

#### 4.2.1.1 Jumlah Input

Kritikalitas pada pemilihan variabel input terdapat pada pemilihan jumlah variabel masa lalu (lagged variables) dari variabel input. Balkin menyarankan bahwa nilai-nilai di masa lalu (lag) yang memungkinkan yang sebaiknya dipilih untuk data harian adalah lag pertama sampai keenam. Sehingga, lag pertama sampai keenam dipilih sebagai variabel input pada penelitian ini, dan kami dapatkan:

$$y_t = f(y_{t-1}, y_{t-2}, y_{t-3}, y_{t-4}, y_{t-5}, y_{t-6}) \qquad [2]$$

#### 4.2.1.2 Jumlah Lapisan Tersembunyi

DTREG mendukung sampai dua lapisan tersembunyi, sehingga satu atau paling banyak dua lapisan tersembunyi digunakan pada percobaan untuk menemukan model terbaik. Pada prakteknya, jaringan neural dengan satu dan terkadang dua lapisan tersembunyi banyak digunakan dan telah menunjukkan hasil yang sangat baik. Peningkatan jumlah lapisan tersembunyi juga meningkatkan waktu perhitungan dan terjadinya bahaya overfitting yang menyebabkan kinerja pendugaan diluar sampel (out-of-sample) yang buruk.

#### 4.2.1.3 Jumlah Neuron Tersembunyi

Metode pencarian yang sempurna digunakan untuk menentukan topologi jaringan yang terbaik. Metode ini melakukan pencarian yang sempurna pada semua topologi pada kisaran pencarian yang kita tentukan, mempekerjakan sampai 2 jaringan lapisan tersembunyi.

#### 4.2.1.4 Jumlah Output

Penentuan jumlah output terkadang sedikit lebih mudah karena terdapat alasan yang menarik, yaitu selalu menggunakan hanya satu neuron output. Jaringan neural dengan output yang banyak, terutama jika output ini sangat longgar, akan menghasilkan hasil yang lebih rendah jika dibandingkan dengan jaringan dengan output tunggal.

#### 4.2.1.5 Fungsi Transfer / Aktivasi

Sementara mayoritas para peneliti menggunakan fungsi aktivasi logistik (sigmoid) pada noktah tersembunyi, tidak ada suatu keharusan dimana aktivasi sebaiknya digunakan pada noktah output. Sejumlah peneliti menggunakan fungsi aktivasi logistik (simoid) secara sederhana terhadap semua noktah output dan noktah tersembunyi (Zhang, 1998). Pada penelitian ini, kombinasi fungsi aktivasi logistik (sigmoid) dan linear digunakan pada lapisan tersembunyi maupun lapisan output. Fungsi lainnya seperti transigmoid tidak tersedia di dalam software DTREG.

#### 4.2.1.6 Kriteria Evaluasi

Penelitian ini menggunakan empat kriteria evaluasi yang paling sering digunakan adalah ME (Galat Maksimal), MSE (Galat Kuadrat Rata-rata), MAE (Galat Absolut Maksimal), dan MAPE (Galat Persentase Absolut Rata-rata). Rumusnya dapat ditunjukkan sebagai berikut:

$$MSE = \frac{\sum (y_t - \hat{y}_t)^2}{T} \qquad [3]$$

$$MAE = \frac{\sum |y_t - \hat{y}_t|}{T} \qquad [4]$$

$$MAPE = \frac{1}{T} \sum \left| \frac{y_t - \hat{y}_t}{y_t} \right| \text{ x } 100 \qquad [5]$$

dimana yt merupakan pengamatan sebenarnya, ŷt adalah nilai yang diprediksi, dan T adalah jumlah prediksi.

#### 4.2.1.7 Algoritma Pelatihan

Dua metode pelatihan digunakan, yaitu gradien konjugat berskala dan gradien konjugat tradisional.

Dalam hal ini, dipilih model MLP yang menghasilkan galat pendugaan minimal pada kelompok data validasi. 48 model total yang memungkinkan seperti yang ditunjukkan pada Tabel 4.2. digunakan pada percobaan ini. Nilai-nilai dengan warna merah menunjukkan nilai minimal. Kami mendapatkan model nomor 2 sebagai model terbaik pada MLP karena memberikan nilai terendah dari ketiga kriteria yang berbeda dari serangkaian data validasi. Model nomor 2 dapat dijelaskan sebagai berikut:

- Penggunaan data mentah tanpa stabilisasi variansi dilakukan pada data
- Satu lapisan tersembunyi
- Fungsi aktivasi logistik (sigmoid) diterapkan pada lapisan tersembunyi dan

fungsi aktivasi linear diterapkan pada lapisan output

- Gradien konjugat Scaled sebagai metode pelatihan
- Model terbaik yang dipilih menggunakan pola 6-3-1 sebagai topologi jaringan (6 noktah input, 3 neuron tersembunyi, dan 1 oktah output).

**Tabel 4.2 Semua topologi MLP yang memungkinkan**

| Model | 1 Hidden Layer | 2 Hidden layer | Hidden layer activation function | Output layer activation function | Training Metod | NN Architecture | Training data | | | | Validation Data | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | ME | MSE | MAE | MAPE | ME | MSE | MAE | MAPE |
| 1* | v | | Linear | Linear | Scaled | 6-2-1 | 14333.4 | 442965.6 | 377.4 | 2.387 | 647.8 | 161804.4 | 343.1 | 1.591 |
| 2* | v | | Linear | Logistic | Scaled | 6-3-1 | 14228.3 | 614438.0 | 539.0 | 4.314 | 147.2 | **6223.0** | **62.3** | **0.293** |
| 3* | v | | Logistic | Linear | Scaled | 6-7-1 | 14388.8 | 438307.8 | 377.2 | 2.414 | 257.5 | 25458.2 | 145.0 | 0.674 |
| 4* | v | | Logistic | Logistic | Scaled | 6-4-1 | 14374.1 | 445004.6 | 382.2 | 2.499 | 542.0 | 118994.4 | 297.1 | 1.378 |
| 5* | | v | Linear | Linear | Scaled | 6-2-4-1 | 14352.8 | 443924.8 | 379.1 | 2.407 | 519.7 | 102250.3 | 268.3 | 1.244 |
| 6* | | v | Linear | Logistic | Scaled | 6-2-4-1 | 14230.6 | 618614.5 | 540.9 | 4.315 | 160.4 | 8549.8 | 80.7 | 0.378 |
| 7* | | v | Logistic | Linear | Scaled | 6-9-4-1 | 14352.3 | 539081.1 | 457.9 | 3.182 | 304.2 | 50128.8 | 204.5 | 0.950 |
| 8* | | v | Logistic | Logistic | Scaled | 6-2-4-1 | 14601.0 | 1146800.0 | 743.8 | 5.799 | 923.1 | 348130.5 | 533.2 | 2.477 |
| 9* | v | | Linear | Linear | Traditional | 6-14-1 | 14360.3 | 444486.9 | 380.6 | 2.444 | 471.0 | 85132.3 | 248.1 | 1.151 |
| 10* | v | | Linear | Logistic | Traditional | 6-5-1 | 14226.5 | 615379.5 | 538.3 | 4.316 | 139.4 | 7817.6 | 82.1 | 0.384 |
| 11* | v | | Logistic | Linear | Traditional | 6-8-1 | 14387.3 | 434993.8 | 373.5 | 2.374 | 295.3 | 36201.3 | 171.0 | 0.794 |
| 12* | v | | Logistic | Logistic | Traditional | 6-3-1 | 14323.3 | 454168.2 | 386.1 | 2.555 | 601.7 | 134812.4 | 310.4 | 1.440 |

| Model | 1 Hidden Layer | 2 Hidden layer | Hidden layer activation function | Output layer activation function | Training Metod | NN Architecture | Training data | | | | Validation Data | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | ME | MSE | MAE | MAPE | ME | MSE | MAE | MAPE |
| 13* | | v | Linear | Linear | Traditional | 6-4-4-1 | 14350.9 | 433676.6 | 372.0 | 2.361 | 606.1 | 147068.4 | 325.9 | 1.511 |
| 14* | | v | Linear | Logistic | Traditional | 6-8-4-1 | 14229.0 | 616331.6 | 539.1 | 4.311 | 139.0 | 7382.4 | 77.9 | 0.365 |
| 15* | | v | Logistic | Linear | Traditional | 6-2-4-1 | 14392.0 | 437956.4 | 378.1 | 2.432 | 212.7 | 19893.3 | 130.7 | 0.608 |
| 16* | | v | Logistic | Logistic | Traditional | 6-7-4-1 | 14443.1 | 463856.7 | 401.3 | 2.702 | **128.3** | 11593.4 | 98.5 | 0.460 |
| 17** | v | | Linear | Linear | Scaled | 6-5-1 | 4423.7 | 303903.0 | **362.0** | 2.174 | 384.6 | 59369.0 | 208.4 | 0.967 |
| 18** | v | | Linear | Logistic | Scaled | 6-20-1 | 4297.6 | 371393.1 | 416.5 | 2.663 | 1024.5 | 442172.9 | 567.0 | 2.629 |
| 19** | v | | Logistic | Linear | Scaled | 6-4-1 | 4355.1 | 328248.6 | 383.1 | 2.373 | 359.9 | 52484.5 | 199.7 | 0.927 |
| 20** | v | | Logistic | Logistic | Scaled | 6-4-1 | 4330.8 | 384942.2 | 421.9 | 2.693 | 1109.4 | 543164.2 | 647.5 | 3.004 |
| 21** | | v | Linear | Linear | Scaled | 6-3-4-1 | 4428.5 | 323168.3 | 378.5 | 2.283 | 298.3 | 43397.0 | 182.6 | 0.848 |
| 22** | | v | Linear | Logistic | Scaled | 6-2-4-1 | 4300.5 | 374415.9 | 418.3 | 2.673 | 985.4 | 427868.1 | 567.9 | 2.634 |
| 23** | | v | Logistic | Linear | Scaled | 6-10-4-1 | 4557.6 | 511978.8 | 482.0 | 2.808 | 850.3 | 301314.6 | 497.2 | 2.309 |
| 24** | | v | Logistic | Logistic | Scaled | 6-2-4-1 | 5038.8 | 969765.2 | 653.9 | 4.325 | 974.8 | 512674.0 | 674.6 | 3.136 |
| 25** | v | | Linear | Linear | Traditional | 6-5-1 | 4424.0 | 305116.2 | 363.0 | 2.182 | 351.6 | 47455.2 | 187.4 | 0.870 |
| 26** | v | | Linear | Logistic | Traditional | 6-5-1 | 4301.1 | 375058.2 | 423.6 | 2.702 | 1022.6 | 407216.2 | 540.5 | 2.506 |

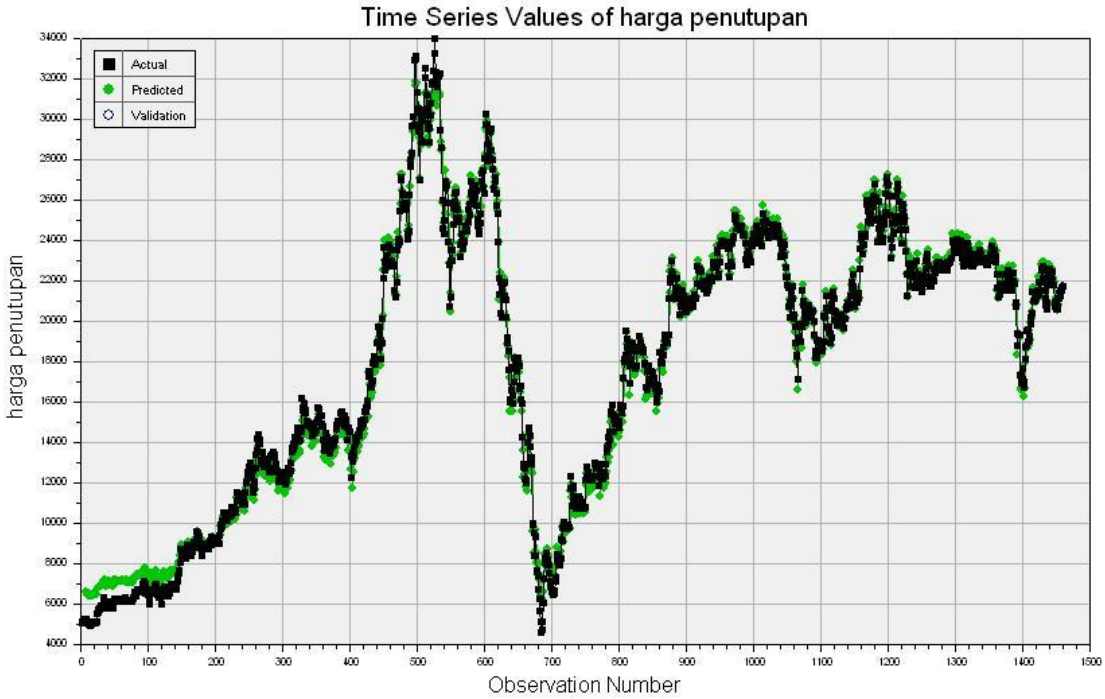
| Model | 1 Hidden Layer | 2 Hidden layer | Hidden layer activation function | Output layer activation function | Training Metod | NN Architecture | Training data | | | | Validation Data | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | ME | MSE | MAE | MAPE | ME | MSE | MAE | MAPE |
| 27** | v | | Logistic | Linear | Traditional | **6-8-1** | 4411.8 | **302638.6** | 364.2 | 2.213 | 471.4 | 93899.5 | 265.6 | 1.232 |
| 28** | v | | Logistic | Logistic | Traditional | 6-2-1 | 4446.3 | 302763.4 | 363.9 | 2.209 | 328.3 | 45822.5 | 185.4 | 0.860 |
| 29** | | v | Linear | Linear | Traditional | 6-5-4-1 | 4424.9 | 309826.1 | 368.4 | 2.223 | 278.6 | 31826.1 | 156.9 | 0.729 |
| 30** | | v | Linear | Logistic | Traditional | 6-3-4-1 | 4307.8 | 382045.9 | 419.6 | 2.673 | 870.0 | 324512.6 | 491.4 | 2.279 |
| 31** | | v | Logistic | Linear | Traditional | 6-4-4-1 | 4394.0 | 305689.3 | 367.5 | 2.230 | 630.6 | 157532.0 | 337.4 | 1.564 |
| 32** | | v | Logistic | Logistic | Traditional | 6-16-4-1 | 4416.2 | 336920.3 | 387.7 | 2.388 | 507.4 | 102580.8 | 267.0 | 1.238 |
| 33*** | v | | Linear | Linear | Scaled | 6-18-1 | 4108.7 | 303419.3 | 362.2 | **2.170** | 341.7 | 49123.9 | 194.8 | 0.905 |
| 34*** | v | | Linear | Logistic | Scaled | 6-2-1 | 4302.1 | 392660.4 | 428.2 | 2.706 | 990.9 | 403040.9 | 537.1 | 2.490 |
| 35*** | v | | Logistic | Linear | Scaled | 6-2-1 | 4103.9 | 306119.0 | 369.1 | 2.236 | 542.8 | 123572.7 | 305.0 | 1.415 |
| 36*** | v | | Logistic | Logistic | Scaled | 6-2-1 | 4429.6 | 427369.2 | 447.7 | 2.903 | 964.2 | 393201.2 | 535.3 | 2.482 |
| 37*** | | v | Linear | Linear | Scaled | 6-4-4-1 | 4115.1 | 304531.8 | 363.2 | 2.176 | 296.3 | 35406.2 | 170.7 | 0.793 |
| 38*** | | v | Linear | Logistic | Scaled | 6-2-4-1 | 4334.4 | 402231.6 | 432.0 | 2.725 | 981.8 | 391144.7 | 534.4 | 2.478 |
| 39*** | | v | Logistic | Linear | Scaled | 6-3-4-1 | 4107.2 | 319463.5 | 380.8 | 2.336 | 655.5 | 168248.9 | 342.5 | 1.588 |

| Model | 1 Hidden Layer | 2 Hidden layer | Hidden layer activation function | Output layer activation function | Training Metod | NN Architecture | Training data | | | | Validation Data | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | ME | MSE | MAE | MAPE | ME | MSE | MAE | MAPE |
| 40*** | | v | Logistic | Logistic | Scaled | 6-6-4-1 | 4722.8 | 505877.1 | 481.4 | 3.127 | 997.7 | 413953.9 | 546.9 | 2.536 |
| 41*** | v | | Linear | Linear | Traditional | 6-9-1 | 4116.8 | 308493.1 | 365.9 | 2.196 | 219.1 | 27399.4 | 149.6 | 0.696 |
| 42*** | v | | Linear | Logistic | Traditional | 6-9-1 | 4336.5 | 395798.2 | 428.6 | 2.704 | 961.2 | 379195.1 | 522.7 | 2.424 |
| 43*** | v | | Logistic | Linear | Traditional | 6-19-1 | 4120.8 | 306471.6 | 365.9 | 2.193 | 286.8 | 31281.5 | 158.9 | 0.738 |
| 44*** | v | | Logistic | Logistic | Traditional | 6-6-1 | 4166.4 | 340571.5 | 390.4 | 2.393 | 1056.4 | 492904.0 | 605.3 | 2.807 |
| 45*** | | v | Linear | Linear | Traditional | 6-10-4-1 | 4112.2 | 304679.4 | 362.8 | 2.175 | 313.2 | 42363.7 | 184.4 | 0.857 |
| 46*** | | v | Linear | Logistic | Traditional | 6-18-4-1 | 4266.2 | 389873.4 | 426.6 | 2.705 | 1053.6 | 468948.4 | 586.6 | 2.720 |
| 47*** | | v | Logistic | Linear | Traditional | 6-14-4-1 | **4099.7** | 307276.2 | 365.9 | 2.199 | 404.2 | 66435.8 | 224.3 | 1.041 |
| 48*** | | v | Logistic | Logistic | Traditional | 6-10-4-1 | 4176.2 | 349963.2 | 394.5 | 2.424 | 1618.7 | 1085600.0 | 909.2 | 4.218 |

Ket :

ME = *Maximum Error*
MSE = *Mean Squared Error*
MAE = *Mean Absolute Error*
MAPE = *Mean Absolute Percentage Error*

Dari tabel 4.2. topologi *Multilayer Perceptron (MLP)* yang memungkinkan, pada Model nomor dua ditemukan MSE, MAE dan MAPE terendah dari dari model-model yang lain.Maka dengan itu, peneliti mengambil model nomor dua sebagai model terbaik dalam *Multilayer Perceptron (MLP)*.

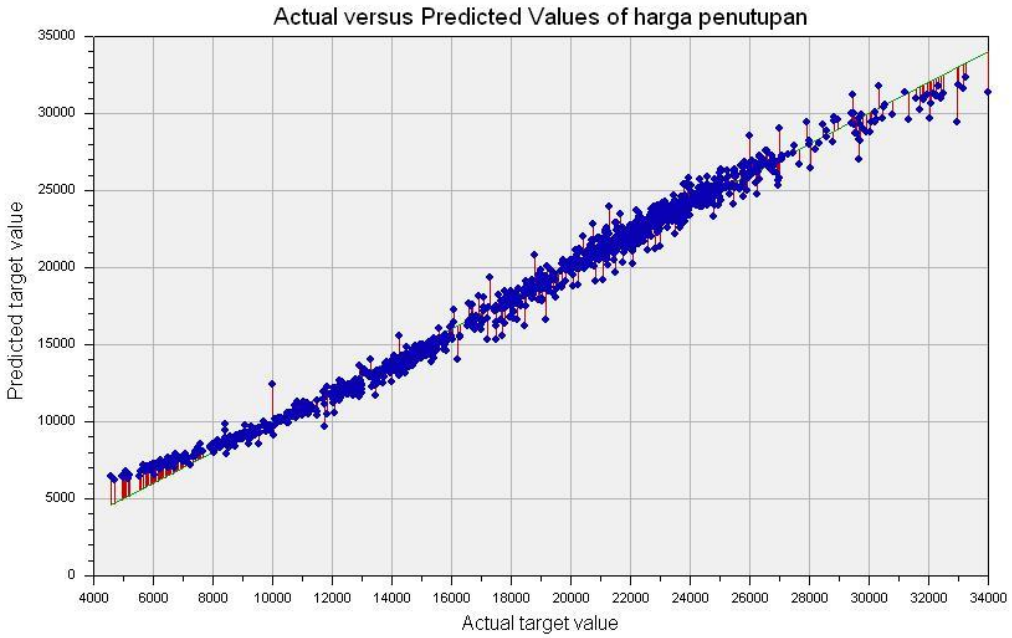



Sumbu x = Jumlah Pengamatan

Sumbu y = Harga

**Gambar 4.2. Nilai seri waktu harga pada MLP (6-3-1)**

Diagram seri waktu ditunjukkan pada Gambar 4.2. Kotak yang berwarna hitam menunjukkan nilai yang sebenarnya dari seri waktu, kotak yang berwarna hijau menunjukkan nilai yang diprediksi pada titik-titik yang bersesuaian dengan titik-titik pelatihan, dan bulatan terbuka dengan garis pinggir berwarna biru menunjukkan nilai yang diprediksi pada baris validasi yang digunakan pada pelatihan.




sumbu x = nilai target yang sebenarnya

sumbu y = nilai targel yang diprediksi

**Gambar 4.3. Nilai yang diprediksi terhadap nilai yang sebenarnya dalam hal harga pada MLP (6-3-1)**

Diagram yang sebenarnya terhadap diagram yang diprediksi (diagram residual) ditunjukkan pada Gambar 4.3. Diagram ini menampilkan suatu titik pada setiap baris data. Koordinat X dari suatu titik merupakan nilai target yang sebenarnya. Koordinat Y dari titik tersebut adalah nilai target yang diprediksi yang bersesuaian.

Dengan suatu model yang sempurna, nilai yang diprediksi akan sama dengan nilai yang sebenarnya, koordinat X dan Y pada setiap titik akan sama, dan semua titik akan ditempatkan pada garis diagonal dimana X = Y. Jika nilai yang diprediksi berbeda dengan nilai yang sebenarnya, titik-titik tersebut akan ditutupi dengan titik dari garis diagonal, dan jarak vertikal dari garis tersebut menuju ke titik tersebut merupakan galatnya (residual). Galatnya disimbolkan dengan garis vertikal berwarna merah.

### 4.2.1 Model Jaringan *Radial Basis Function* (RBF) Pada Prediksi Harga Saham

#### 4.2.2.1 Input Variabel dan Pra-pengolahan Data

Agar memiliki perbandingan yang realistik, jaringan neural RBF juga dirancang dengan enam input. Tiga proses pra-pengolahan data yang berbeda diusulkan dalam penelitian ini, yaitu:

Model *Radial Basis Function* yang pertama menggunakan data mentah, model yang kedua dilakukan penghilangan kecenderungan tanpa stabilisasi variansi dan percobaan ketiga dilakukan penghilangan kecenderungan dengan stabilisasi variansi

Dari tiga percobaan diatas akan dihasilkan model terbaik dari model *Radial Basis function.* Model terbaik dari percobaan diatas akan dikomparasikan dengan model lain dari *Neural Network.*

#### 4.2.2.2 Jaringan RBF Pelatihan

DTREG menggunakan algoritma pelatihan yang dikembangkan oleh Sheng Chen, Xia Hong, dan Chris J. Harris pada tahun 2005. Algoritma ini menggunakan suatu pendekatan yang evolusioner dalam menentukan titik pusat optimal dan tersebar pada setiap neuron. Algoritma ini juga menentukan ketika menghentikan penambahan neuron pada jaringan dengan memantau galat leave-one-out (LOO) (leave one out = meninggalkan yang satu itu (di situ); leave them out! = tinggalkan mereka!) yang diduga dan mengakhirinya ketika galat LOO cenderung meningkat akibat overfitting (Sherrod, 2010).

Perhitungan berat optimal antara neuron pada lapisan tersembunyi dan lapisan penjumlahan akhir dilakukan dengan regresi bubungan (ridge regression). Prosedur iteratif yang dikembangkan oleh Mark Orr pada tahun 1966 digunakan untuk menghitung regularisasi optimal untuk parameter Lambda yang meminimalkan galat validasi silang yang digeneralisasi (GCV) (Sherrod, 2010).

#### 4.2.2.3 Hasil Empiris

Model RBF No. 2 dipilih sebagai model terbaik karena menghasilkan galat pendugaan minimal (lihat Tabel 4.3.) yang ditunjukkan oleh nilai berwarna merah. Jumlah RBF satuan pada lapisan tersembunyi tunggal adalah sebanyak 25, dan parameter sebaran minimal (jangkauan RBF) adalah 0,01 dan maksimal 479.952. Parameter Lambda maksimal masing-masing adalah, 0.02256 dan 9.3274.1

**Tabel 4.3. Hasil pelatihan pada semua model RBF yang memungkinkan**

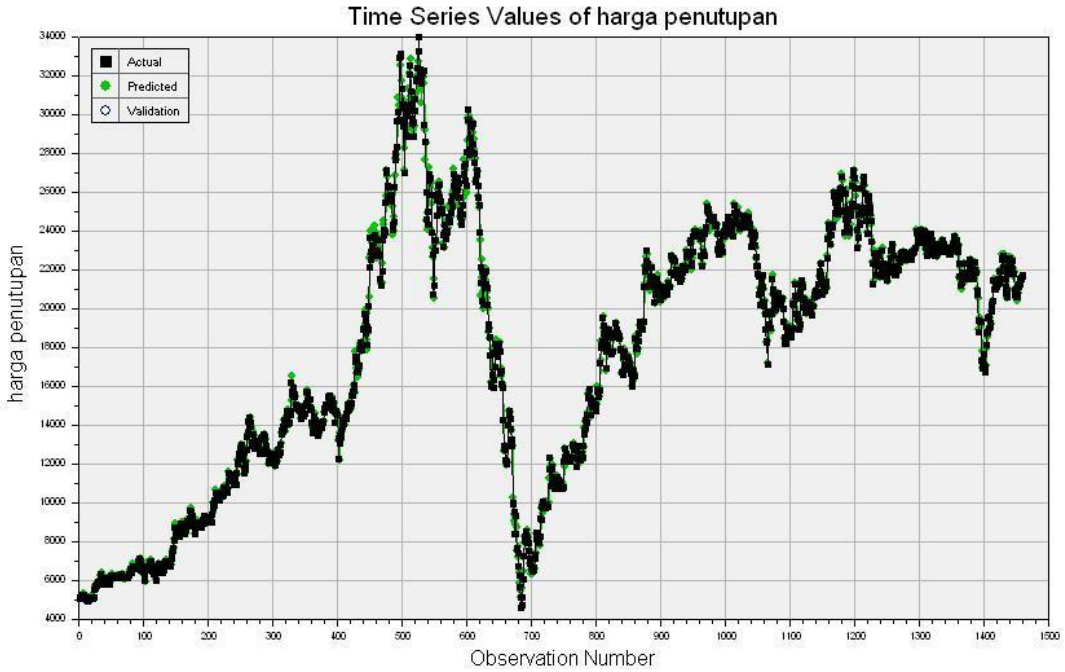
| RBF Model | Training data | | | | Validation Data | | | |
|---|---|---|---|---|---|---|---|---|
| | ME | MSE | MAE | MAPE | ME | MSE | MAE | MAPE |
| RBF Model #1 | 4817.1 | 308764.7 | 370.1 | 2.270 | 220.9 | 17261.8 | 100.0 | 0.470 |
| RBF Model #2 | 4419.4 | 275091.7 | 353.1 | 2.150 | 173.8 | 18195.6 | 130.9 | 0.610 |
| RBF Model #3 | 4088.9 | 286185.7 | 357.4 | 2.152 | 220.0 | 25513.9 | 143.8 | 0.671 |

RBF Model #1 = Menggunakan Data Mentah

RBF Model #2 = Menggunakan Trend Removal

RBF Model #3 = Menggunakan Trend Removal dan Stabilize Variance

Dari tabel 4.3. Hasil pelatihan pada semua model *Radial Basis Function (RBF)* yang memungkinkan yang memungkinkan, pada Model nomor dua ditemukan MSE, MAE dan MAPE terendah dari dari model-model yang lain.Maka dengan itu, peneliti mengambil model nomor dua sebagai model terbaik dalam *Radial Basis Function (RBF)*.



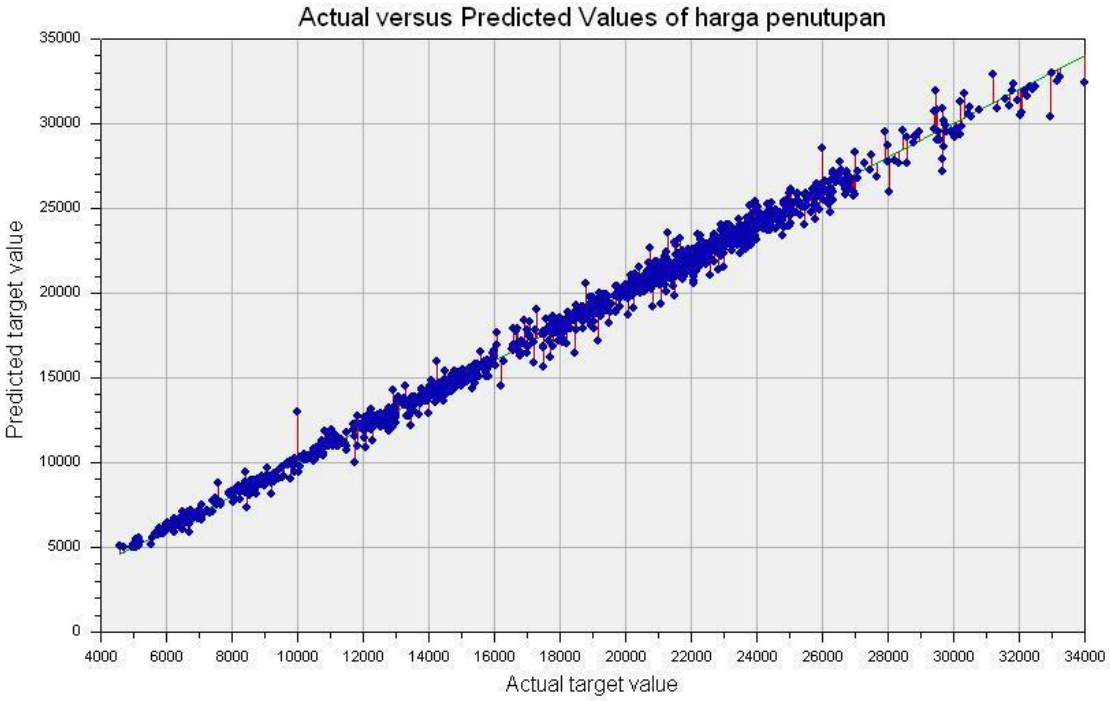

Sumbu x = Jumlah Pengamatan

Sumbu y = Harga

**Gambar 4.4 Nilai seri waktu harga pada model RBF #2**

Bagan seri waktu ditunjukkan pada Gambar 4.4. Kotak berwarna hitam menunjukkan nilai seri waktu yang sebenarnya, kotak berwarna hijau menunjukkan nilai yang diprediksi pada titik yang bersesuaian dengan titik-titik pelatihan, dan lingkaran terbuka dengan garis pinggir berwarna biru menunjukkan nilai yang diprediksi pada baris validasi yang tidak digunakan pada pelatihan.

Diagram nilai yang sebenarnya terhadap nilai yang diprediksi (diagram residual) ditunjukkan pada Gambar 4.5.




sumbu x = nilai target yang sebenarnya

sumbu y = nilai targel yang diprediksi

**Gambar 4.5 Nilai yang sebenarnya terhadap nilai yang diprediksi dalam hal harga pada model RBF #2**

Diagram yang sebenarnya terhadap diagram yang diprediksi (diagram residual) ditunjukkan pada Gambar 4.5. Diagram ini menampilkan suatu titik pada setiap baris data. Koordinat X dari suatu titik merupakan nilai target yang sebenarnya. Koordinat Y dari titik tersebut adalah nilai target yang diprediksi yang bersesuaian.

Dengan diagram residual diatas, dapat dilihat selisih harga saham yang sebenarnya dengan harga saham yang diprediksi menggunakan model *Radial Basis Function (RBF)* sehingga terlihat bahwa tingkat keeroran dari setiap harinya.

## 4.3. Hasil Perbandingan antara *Multilayer Perceptron* dan *Radial Basis Function*

Dari hasil penelitian diatas, Peneliti melakukan percobaan 48 model kemungkinan di *Multilayer Perceptron* dan 3 model kemungkinan di *Radial Basis* Function. Dari Model-model yang telah dilakukan, peneliti mencari *Mean Square Error (MSE)* yang paling kecil.

Dalam hal keakuratan pendugaan, Tabel 4.4 menunjukkan bahwa semua *Neural Network* Model *Radial Basis Function* mengungguli model *Multilayer Perceptron*. Semua kriteria pengukuran, yaitu ME, MSE, MAE, dan MAPE, pada model *Multilayer Perceptron* jauh lebih tinggi daripada pendugaan jaringan neural *Radial Basis Function*.

**Tabel 4.4 Perbandingan keakuratan pendagaan diantara model-model *neural network***

| MODEL | Training Data | | | | Validation Data | | | |
|---|---|---|---|---|---|---|---|---|
| | ME | MSE | MAE | MAPE | ME | MSE | MAE | MAPE |
| MLP | 14228.3 | 614438.0 | 539.0 | 4.314 | 147.2 | 6223.0 | 62.3 | 0.293 |
| RBF | 4419.4 | 275091.7 | 353.1 | 2.150 | 173.8 | 18195.6 | 130.9 | 0.610 |

Dari tabel 4.4. perbandingan model-model *neural* network , pada Model *Radial Basis Function (RBF)* ditemukan MSE, MAE dan MAPE terendah dari dari model *Multilayer Perceptron (MLP)*.Maka dengan itu, peneliti mengambil model *Radial Basis Function (RBF)* sebagai model terbaik dalam pendugaan harga saham.

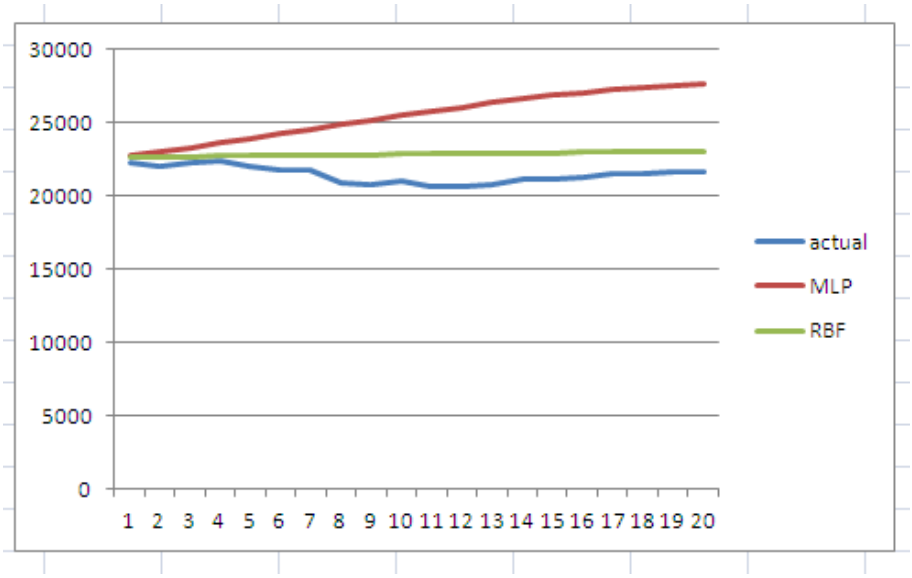
Dari Running menggunakan DTReg, berikut perbandingan harga saham sebenarnya dan prediksi harga saham menggunakan model *Neural Network* :

**Tabel 4.5. Hasil Running perbandingan harga sebenarnya dengan model *Neural Network***

| Row | actual | MLP | RBF |
|---|---|---|---|
| 1440 | 22250 | 22807 | 22651 |
| 1441 | 22000 | 23054 | 22635 |
| 1442 | 22250 | 23343 | 22680 |
| 1443 | 22450 | 23635 | 22719 |
| 1444 | 22050 | 23925 | 22758 |
| 1445 | 21800 | 24223 | 22782 |
| 1446 | 21800 | 24532 | 22801 |
| 1447 | 20900 | 24848 | 22820 |
| 1448 | 20750 | 25165 | 22840 |
| 1449 | 21000 | 25481 | 22860 |
| 1450 | 20600 | 25787 | 22881 |
| 1451 | 20600 | 26081 | 22901 |
| 1452 | 20750 | 26362 | 22920 |
| 1453 | 21150 | 26621 | 22940 |
| 1454 | 21100 | 26857 | 22959 |
| 1455 | 21250 | 27073 | 22979 |
| 1456 | 21500 | 27262 | 22998 |
| 1457 | 21500 | 27427 | 23017 |
| 1458 | 21600 | 27568 | 23036 |
| 1459 | 21700 | 27689 | 23055 |

Tabel 4.5. diatas dapat menunjukkan selisih harga yang sebenarnya dengan harga prediksi menggunakan model-model *neural network* dan harga yang paling mendekati dari harga saham sebenarnya memperoleh model terbaik dari penelitian ini.

Berikut grafik perbandingan harga saham sebenarnya dengan prediksi harga saham menggunakan *Neural Network :*




**Gambar 4.6. Perbandingan Harga Saham menggunakan *Neural Network***

Gambar 4.6 di bawah ini menunjukkan grafik garis bahwa model-model *neural network* model *Radial Basis Function* memberikan keakuratan yang lebih baik daripada model *Multilayer Perceptron*.

Garis hijau menunjukkan prediksi harga saham menggunakan model *Radial Basis Function* dan gari merah menunjukkan prediksi harga saham menggunakan model *Multilayer Perceptron.* Sedangkan garis biru menunjukkan harga saham actual. Dari diagram dapat dilihat komparasi model harga saham yang paling mendekati harga sebenarnya.

## 4.4 Implikasi Penelitian

Sebagai suatu penelitian dalam bidang sistem pendukung keputusan maka penelitian ini mempunyai implikasi terhadap pendugaan harga saham, terhadap manajerial PT. Astra Agro Lestari, Tbk dan terhadap penelitian-penelitian selanjutnya yang akan diuraikan dibawah ini.

### 4.4.1. Implikasi terhadap sistem pendukung keputusan untuk memprediksi harga saham

Dalam menjalankan sebuah penelitian tentunya membutuhkan spesifikasi minimum dari perangkat atau *hardware* dalam menjalankan suatu aplikasi. Spesifikasi perangkat keras minimum yang dibutuhkan adalah sistem operasi Windows XP Service Pack 2, processor Intel® Pentium Dual Core, *Harddisk* 250Gb dan RAM 1Gb, serta monitor SVGA.

Penggunaan *Neural Network* yang digunakan dalam penelitian dalam pendugaan harga saham tidak dapat dibuktikan secara langsung keakuratannya tanpa adanya sebuah *software* yang digunakan dalam penelitian tersebut. *DTReg* dengan *Graphical User Interface(GUI)* adalah sebuah *software editor* yang digunakan dalam pengembangan penelitian ini.

Dari hasil penelitian yang telah dilakukan menunjukan bahwa *neural network* model *radial basis function* memiliki tingkat keakuratan yang lebih tinggi dan mengandung implikasi bahwa model *radial basis function* lebih cocok diterapkan untuk memprediksi harga saham yang akan datang, sehingga dengan melihat hasil perbandingan tersebut, maka kedepanya dapat dijadikan oleh para investor sebagai salah satu metode yang tepat untuk memprediksi harga saham yang akan datang.

### 4.4.2. Implikasi terhadap aspek manajerial

Setelah menyelesaikan penelitian ini, dapat disimpulkan bahwa prediksi harga saham banyak mengalami perubahan dari hari ke hari. Oleh karena itu penelitian ini diharapkan dapat berimplikasi terhadap manajerial khususnya di Astra Agro Lestari. bahwa perlu adanya sosialisasi dan training dalam melakukan pendugaan harga saham yang akan datang.

### 4.4.3. Implikasi terhadap penelitan lanjutan

penelitian ini dilakukan dengan model *Multilayer Perceptron* dan *Radial Basis Function*, maka untuk lebih menguji tingkat keakurasian sistem yang telah dibangun, kiranya dimasa yang akan datang perlu adanya pengujian sistem ini apakah dengan penggunaan sistem ini terbukti dengan akurat dalam mempredikasi harga saham.

Oleh karena itu, penelitian ini dapat di eksplorasi lebih dalam lagi dengan menggunakan neural network model lainnya seperti, GMDH Polynomial, SVM, GACRH dll.

Penelitian ini juga harus dikembangkan secara periodik tiap 2 tahun dikarenakan pasar modal setiap tahun mempunyai variabel yang berbeda.

# BAB 5

# PENUTUP

## 5.1 Kesimpulan

Dalam penelitian ini menunjukan hasil bahwa penerapan *neural network* menggunakan model *Multilayer Perceptron dan Radial Basis Function* dalam pendugaan harga saham dengan pemanfaatan data historis di perusahaan.. Dari penelitian yang telah dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Dari hasil pengukuran menggunakan *Radial Basis Function* mempunyai nilai MSE yang paling minimal
2. *Neural Network* model *Radial Basis Function* lebih akurat dibandingkan *Multilayer Perceptron*

## 5.2 Saran

Berdasarkan hasil analisis dan pembahasan, maka agar penelitian ini bisa ditingkatkan, berikut adalah saran-saran yang diusulkan:

1. Spesifikasi peralatan diharapkan menggunakan minimal tahun 2000 dikarenakan software yang digunakan cukup berat untuk dijalankan
2. Para investor diharapkan diberikan sosialisasi tentang prediksi harga saham dengan menggunakan sistem pendukung keputusan ini.
3. Penelitian ini diharapkan dapat di eksplorasi lebih dalam lagi dengan menggunakan *neural network* model lainnya seperti, GMDH Polynomial, SVM, GARCH dll.
4. Penelitian ini diharapkan diharapkan dapat dikembangkan secara periodik setiap 2 tahun dikarenakan pasar modal setiap tahun mempunyai variabel yang berbeda.