STRUKTUR TERBAIK NEURAL NETWORK MENGGUNAKAN ALGORITMA *BACKPROPAGATION* DAN 4 FUNGSI AKTIVASI DALAM MEMPREDIKSI INDEKS HARGA SAHAM GABUNGAN (IHSG)

(Skripsi)

Oleh

NABILLA DEFIZD PUTRI
NPM 1717031035

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2021

ABSTRACT

THE BEST NEURAL NETWORK STRUCTURE USING BACKPROPAGATION ALGORITHM AND 4 ACTIVATION FUNCTIONS FOR INDONESIA COMPOSITE INDEX (IDX) PREDICTION

By

NABILLA DEFIZD PUTRI

Stocks are one of the most popular financial market instruments. One of the parameters that can be seen in stock investment is Indonesian Composite Index (IDI) as known as IDX *Composite*. In this study, the best network structure will be determined using *backpropagation* algorithm to produce a high level of accuracy in predicting IDX closing price. There are 4 activation functions, sigmoid activation function, tanh, ReLU and leaky ReLU an also 2 threshold values 0.01 and 0.001. So the best network structure is obtained by the form of 2 input layers, 1 hidden layer contains 2 nodes using the leaky ReLU activation function and a 0,01 threshold value. The best network structure produces an 0,244838 MSE value and 0.8686% MAPE value or 9.1314%. of accuracy level.

Keywords: IDX Prediction, Neural Network, Backpropagation, Neural Network Structure.

ABSTRAK

STRUKTUR TERBAIK NEURAL NETWORK MENGGUNAKAN ALGORITMA BACKPROPAGATION DAN 4 FUNGSI AKTIVASI DALAM MEMPREDIKSI INDEKS HARGA SAHAM GABUNGAN (IHSG)

Oleh

NABILLA DEFIZD PUTRI

Saham merupakan salah satu instrumen pasar keuangan yang paling populer. Salah satu parameter yang dapat dilihat dalam investasi saham adalah pergerakan indeks harga saham gabungan atau IHSG. Dalam penelitian ini akan ditentukan struktur jaringan terbaik menggunakan Algoritma *Backpropagation* dengan tujuan menghasilkan tingkat keakuratan yang tinggi dalam memprediksi harga penutupan IHSG. Akan digunakan 4 fungsi aktivasi yaitu fungsi aktivasi sigmoid, tanh, ReLU dan leaky ReLU serta 2 nilai *threshold* sebesar 0,01 dan 0,001. Sehingga didapatkan struktur jaringan terbaik berupa 2 *input layer,* 1 *hidden layer* berisi 2 *nodes* dengan menggunakan fungsi aktivasi leaky ReLU dan nilai *threshold* sebesar 0,01. Dengan menggunakan struktur jaringan terbaik didapatkan nilai MSE sebesar 0,244838 dan nilai MAPE sebesar 0,8686% atau dengan akurasi sebesar 99,1314%.

Kata kunci: Prediksi Indeks Harga Saham Gabungan (IHSG), *Neural Network*, *Backpropagation,* Struktur *Neural Network.*

STRUKTUR TERBAIK NEURAL NETWORK MENGGUNAKAN ALGORITMA *BACKPROPAGATION* DAN 4 FUNGSI AKTIVASI DALAM MEMPREDIKSI INDEKS HARGA SAHAM GABUNGAN (IHSG)

Oleh

NABILLA DEFIZD PUTRI

Skripsi

Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA MATEMATIKA

Pada

Jurusan Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Lampung

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2021

Judul Skripsi	:	**STRUKTUR TERBAIK NEURAL NETWORK MENGGUNAKAN ALGORITMA BACKPROPAGATION DAN 4 FUNGSI AKTIVASI DALAM MEMPREDIKSI INDEKS HARGA SAHAM GABUNGAN (IHSG)**
Nama Mahasiswa	:	Nabilla Defizd Putri
Nomor Pokok Mahasiswa	:	**1717031035**
Program Studi	:	**Matematika**
Fakultas	:	**Matematika dan Ilmu Pengetahuan Alam**

MENYETUJUI

1. Komisi Pembimbing

Dian Kurniasari, S.Si., M.Sc.
NIP.196903051996032001

Ir. Warsono, M.S., Ph.D.
NIP. 196302161987031003

2. Ketua Jurusan Matematika

Dr. Aang Nuryaman, S.Si., M.Si.
NIP. 197403162005011001

MENGESAHKAN

1. Tim Penguji

Ketua : **Dian Kurniasari, S.Si., M.Sc.** .....

Sekertaris : **Ir. Warsono, M.S., Ph.D.** .....

Penguji
Bukan Pembimbing : **Prof. Drs. Mustofa Usman, M.A., Ph.D.** .....

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Dr. Eng. Suripto Dwi Yuwono, S.Si., M.T.
NIP. 197407052000031001

Tanggal Lulus Ujian Skripsi : **06 Agustus 2021**

PERNYATAAN SKRIPSI MAHASISWA

Yang bertanda tangan di bawah ini :

Nama Mahasiswa	: **Nabilla Defizd Putri**
Nomor Pokok Mahasiswa	: **171731035**
Jurusan	: **Matematika**
Judul Skripsi	: **STRUKTUR TERBAIK NEURAL NETWORK MENGGUNAKAN ALGORITMA BACKPROPAGATION DAN 4 FUNGSI AKTIVASI DALAM MEMPREDIKSI INDEKS HARGA SAHAM GABUNGAN (IHSG)**

Dengan ini menyatakan bahwa penelitian ini adalah hasil pekerjaan saya sendiri dan apabila kemudian hari terbukti bahwa skripsi ini merupakan hasil salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan akademik yang berlaku.

Bandar Lampung, Agustus 2021

Penulis

Nabilla Defizd Putri

RIWAYAT HIDUP

Penulis bernama Nabilla Defizd Putri, dilahirkan di Kota Bandar Lampung, Provinsi Lampung pada 14 Maret 2000. Penulis merupakan anak tunggal dari pasangan Bapak Hafisd dan Ibu Puji Dewi Margareta.

Penulis mengawali pendidikan di Taman Kanak-Kanak (TK) Beringin Raya pada tahun 2004-2005. Kemudian menempuh pendidikan Sekolah Dasar (SD) di SDN Gambir 01 Pagi Jakarta pada tahun 2005-2011. Melanjutkan ke Sekolah Menengah Pertama di SMPN 5 Jakarta lulus pada tahun 2014. Sekolah Menengah Atas di SMAN 1 Jakarta dan lulus pada tahun 2017.

Pada tahun 2017 penulis terdaftar sebagai mahasiswa S1 Jurusan Matematika FMIPA UNILA melalui jalur SNMPTN. Selama menjadi mahasiswa penulis juga aktif dalam organisasi Himpunan Mahasiswa Matematika (HIMATIKA) FMIPA UNILA. Pada tahun 2020 penulis melakukan Kuliah Praktik (KP) di PT. Lotte Shopping Indonesia dan Kuliah Kerja Nyata (KKN) di Desa Way Rilau, Kecamatan Cukuh Balak, Kabupaten Tanggamus.

KATA INSPIRASI

"Maka sesungguhnya Bersama kesulitan itu ada kemudahan"
(Q.S. Al-Insyirah: 5)

"Sesungguhnya Allah tidak akan mengubah keadaan suatu kaum sebelum mereka mengubah keadaan diri mereka sendiri"
(Q.S. Ar-Ra'd: 11)

"Sesungguhnya jika kamu bersyukur, niscaya Aku akan menambah (nikmat) kepadamu, tetapi jika kamu mengingkari (nikmat-Ku), maka pasti azab-Ku sangat berat"
(Q.S. Ibrahim: 7)

"However difficult life may seem, there is always something you can do, and succeed at. It matters that you don't just give up"
(Stephen Hawking)

PERSEMBAHAN

Dengan mengucapkan rasa syukur atas segala puji dan kehadirat Allah SWT. yang telah melimpahkan nikmat serta hidayah-Nya sehingga skripsi ini dapat diselesaikan. Serta tak lupa juga sholawat serta salam selalu tercurahkan kepada junjungan kita Nabi Muhammad SAW. Dengan penuh ketulusan, penulis mempersembahkan karya kecil ini untuk:

Ibunda Puji Dewi Margareta

Seorang ibu yang selalu memberikan dukungan dalam setiap keputusan dan keadaan, yang menerima segala kekurangan serta selalu memberikan doa pada setiap langkahku.

Dosen Pembimbing dan Pembahas

Terima kasih kepada dosen pembimbing dan pembahas yang sangat berjasa, selalu membantu, memberikan arahan, masukan dan ilmu yang sangat bermanfaat.

Sahabat-sahabatku

Almamater Tercinta Universitas Lampung

SANWACANA

Puji syukur kehadirat Allah SWT. atas rahmat dan hidayah-Nya, shalawat serta salam selalu tercurahkan kepada baginda besar Nabi Muhammad SAW. sehingga penulis dapat menyelesaikan skripsi dengan judul "Struktur Terbaik Neural Network Menggunakan Algoritma Backpropagation Dan 4 Fungsi Aktivasi Dalam Memprediksi Indeks Harga Saham Gabungan (IHSG)". Penulis menyadari bahwa skripsi ini tidak akan terwujud tanpa adanya bantuan, bimbingan serta saran dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati penulis ingin menyampaikan banyak terima kasih kepada:

1. Ibu Dian Kurniasari, S.Si., M.Sc. selaku dosen pembimbing I yang selalu memberikan arahan, bantuan, bimbingan, motivasi dan saran yang mendukung sehingga penulis dapat menyelesaikan skripsi ini.
2. Bapak Ir. Warsono, M.S., Ph.D. selaku dosen pembimbing II atas saran dan masukan kepada penulis sehingga dapat menyelesaikan skripsi ini.
3. Bapak Prof. Drs. Mustofa Usman, M.A., Ph.D. selaku dosen penguji yang telah memberikan kritik dan saran yang membangun selama proses penyusunan skripsi.
4. Bapak Agus Sutrisno, S.Si., M.Si. selaku dosen pembimbing akademik yang telah memberikan bimbingan dan arahan selama masa perkuliahan.
5. Bapak Dr. Aang Nuryaman, S.Si., M.Si. selaku Ketua Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
6. Bapak Dr. Eng. Suripto Dwi Yuwono, S.Si., M.T. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
7. Seluruh dosen, staff, karyawan Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

8. Mama dan Papa yang selalu memberikan doa, dukungan, kasih sayang dan motivasi kepada penulis.
9. Amora Squad; Beta, Umroh, Rafa, Viona dan Arina yang telah menemani penulis dalam berbagai keadaan, terima kasih atas semua cerita dan kenangan selama masa perkuliahan.
10. Fariz, Dwi, Stefani, Yulica, Chaterina dan Dini yang telah memberikan dukungan dan motivasi selama ini.
11. Seluruh pihak terkait yang telah banyak membantu dan tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat banyak kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan guna penyempurnaan skripsi ini.

Bandar Lampung, 06 Agustus 2021

Penulis,

Nabilla Defizd Putri

DAFTAR ISI

Halaman

DAFTAR TABEL

Tabel Halaman

DAFTAR GAMBAR

Gambar Halaman

I. PENDAHULUAN

1.1 Latar Belakang dan Masalah

Dalam kehidupan sehari-hari, sebagian besar aktivitas yang dilakukan berhubungan dengan ilmu matematika dan dapat dimodelkan menjadi model matematika. Ilmu matematika dan statistik sangat bermanfaat dalam kehidupan sehari-hari salah satunya yaitu untuk memperkirakan sesuatu yang mungkin terjadi di masa depan atau biasa disebut prediksi.

Prediksi atau peramalan merupakan seni dan ilmu untuk memperkirakan kejadian di masa depan, sehingga hasil dari peramalan dapat digunakan oleh pemangku kebijakan dalam mengambil kebijakan strategis untuk menyelesaikan persoalan dimasa mendatang (Utami dan Darsyah, 2015). Proses peramalan tentu akan menghasilkan *error* yang menjadikan hasil peramalan tidak dapat dikatakan benar-benar akurat, namun proses peramalan dapat membantu untuk memberikan gambaran tentang masa depan.

Saat ini kegiatan investasi baik saham, emas, *cryptocurrency* maupun reksadana tidak lagi menjadi hal yang asing. Dalam investasi saham terdapat dua analisa untuk menentukan saham mana yang layak dimiliki oleh investor, analisa teknikal dan analisa fundamental (Wijiyanti dkk, 2005). Analisa teknikal identik menggunakan perangkat statistik seperti grafik dan rumus matematika. Salah satu parameter yang dapat dilihat dalam analisa tenikal adalah pergerakan indeks harga saham gabungan atau IHSG.

Indeks harga saham gabungan dapat dikatakan sebagai rata-rata pergerakan harga saham yang ada dalam Bursa Efek Indonesia (BEI), saat indeks harga saham gabungan menurun artinya terdapat sejumlah perusahaan yang mengalami penurunan harga saham, begitu pula saat IHSG mengalami kenaikan. Analisis dan teknik prediksi sangat bermanfaat bagi investor untuk mengatahui waktu yang tepat kapan saham harus dibeli, ditahan atau dijual.

Indeks harga saham gabungan dapat diprediksi menggunakan model matematika, namun pergerakan indeks harga saham gabungan sangat cepat sehingga data yang dihasilkan bersifat fluktuatif atau tidak konstan. Data yang memiliki fluktuasi tinggi akan menyulitkan proses analisa sehingga muncul permasalahan bagaimana cara menentukan model matematika yang tepat untuk memprediksi indeks harga saham gabungan.

Beberapa penelitian tentang prediksi indeks harga saham gabungan telah dilakukan oleh para peneliti. Dalam penelitian Susanti dan Adji (2020), digunakan metode ARIMA untuk memprediksi indeks harga saham gabungan lalu didapatkan kesimpulan bahwa model yang didapatkan tidak dapat digeneralisasi untuk periode yang terlalu jauh. Dalam penelitian Lawrence (1998), digunakan metode *neural network* untuk memprediksi harga pasar saham dan didapatkan bahwa *neural network* lebih unggul dibandingkan semua metode lain. Selain itu, dalam penelitian Hidayat (2016), digunakan algoritma *backpropagation* dengan fungsi aktivasi sigmoid dan disimpulkan bahwa nilai MSE yang didapatkan tidak terlalu maksimal sehingga disarankan untuk dapat meningkatkan akurasi yang lebih baik lagi.

Oleh karena itu, penulis akan mencoba menerapkan algoritma *backpropagation* dengan menggunakan 4 fungsi aktivasi yaitu sigmoid, tanh, ReLU dan leaky ReLU.untuk mencari nilai *error* terkecil sehingga dapat dihasilkan prediksi indeks harga saham gabungan terbaik. Penerapan *neural network* dengan algoritma *backpropagation* diharapkan dapat bermanfaat dalam memprediksi indeks harga saham karena *neural network* memiliki tingkat ketelitian yang tinggi dan memliki kelebihan pada prediksi hubungan non linear.

1.2 Tujuan Penelitian

Adapun tujuan dari penelitian ini diantaranya:

1. Menemukan struktur jaringan terbaik guna memprediksi indeks harga saham gabungan
2. Melakukan prediksi indeks harga saham gabungan menggunakan model *neural network* dengan algoritma *backpropagation*
3. Melihat hubungan antara *running time* dan spesifikasi perangkat yang digunakan

1.3 Manfaat Penelitian

Adapun manfaat penelitian ini yaitu:

1. Sebagai rujukan pengembangan ilmu matematika dalam memprediksi indeks harga saham gabungan serta dapat menjadi bahan pertimbangan dan informasi tambahan bagi peneliti yang akan melakukan penelitian tentang indeks harga saham gabungan.
2. Sebagai bahan pertimbangan bagi investor, calon investor maupun pihak-pihak lain yang memiliki kepentingan terhadap indeks harga saham gabungan.

II. TINJAUAN PUSTAKA

2.1 Peramalan (*Forecasting*)

Peramalan atau *forecasting* merupakan proses sistematis yang digunakan dengan tujuan mengetahui suatu peristiwa di masa yang akan datang. Berdasarkan horizon waktu, peramalan diklasifikasikan menjadi 3 bagian yaitu (Herjanto, 2009):

1. Peramalan jangka panjang adalah peramalan yang mencakup waktu yang lebih besar dari 18 bulan.
2. Peramalan jangka menengah adalah peramalan yang mencakup waktu antara 3 sampai 18 bulan
3. Peramalan jangka pendek adalah peramalan untuk jangka waktu kurang dari 3 bulan

Selain itu, terdapat dua pendekatan untuk melakukan peramalan yaitu dengan pendekatan kuantitatif dan pendekatan kualitatif. Untuk pendekatan kuantitatif dapat digunakan metode-metode statistika seperti model *casual* dan *time series*, sedangkan untuk pendekatan kualitatif dapat dilakukan berdasarkan opini atau pendapat dari pihak bersangkutan.

2.2 Runtun Waktu (*Time Series*)

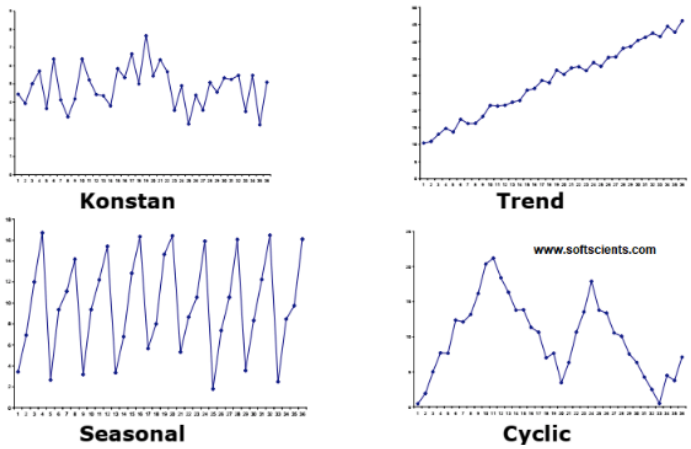
Time series merupakan salah satu metode peramalan yang mempelajari pola gerakan suatu variabel pada suatu interval waktu tertentu. *Time Series* biasanya

memerlukan data *historical* atau data dari masa lalu, untuk mendapatkan pola tertentu. Ketika data runtun waktu digambarkan menjadi sebuah plot, akan terlihat pola-pola tertentu.

2.2.1 Pola Data Time Series

Menurut Makridakis (1997), langkah penting dalam memilih metode perkiraan yang tepat adalah mempertimbangkan jenis pola data, sehingga metode yang paling sesuai dengan pola tersebut dapat digunakan. Empat jenis pola data rangkaian waktu dapat dibedakan: horizontal, musiman, siklis, dan tren.

Pola horizontal muncul ketika nilai data berfluktuasi di sekitar rata-rata konstanta atau disebut "stasioner" dalam rata-ratanya, pola musiman muncul ketika seri dipengaruhi oleh faktor musiman, pola siklis muncul ketika data menunjukkan kenaikan dan penurunan yang bukan berasal dari periode tetap dan pola tren muncul ketika terjadi fluktuasi jangka panjang.



Gambar 1. Pola Data *Time Series*.

Dalam *time series* terdapat koefisien kovarians dan korelasi yang berfungsi untuk mengukur sejauh mana hubungan linier antara dua variabel. Dengan demikian, koefisien ini dapat digunakan untuk mengidentifikasi faktor penjelas. *Autocovariance* dan *autocorrelation* adalah metode sebanding dengan tujuan yang sama dalam *time series.* Oleh karena itu, ACF (*Autocorrelation Function*) dan PACF (*Partial Autocorrelation Function*) sangat diperlukan dalam proses peramalan menggunakan metode runtun waktu.

2.2.2 Autokorelasi

Autokorelasi merupakan suatu korelasi antara X_t dan X_{t+k} (dimana k merupakan *lag).* Autokorelasi biasanya muncul pada data *time series* karena berdasarkan sifatnya data sekarang dipengaruhi oleh data pada masa-masa sebelumnya (Firdaus, 2004).

$$\rho_k = \frac{Cov(X_t, X_{t+k})}{\sqrt{Var(X_t)}\sqrt{Var(X_{t+k})}} \tag{2.1}$$

Untuk mendefinisikan autokorelasi diperlukan definisi autokovarians terlebih dahulu, autokovarians dapat didefinisikan sebagai berikut:

$$\gamma_k = Cov(X_t, X_{t+k}) = E[(X_t - \mu)(X_{t+k} - \mu)] \tag{2.2}$$

ρ_k disebut sebagai autokorelasi (*autocorrelation function*) atau biasa disebut ACF. X_t dan X_{t+k} dikatakan tidak berkorelasi jika ρ_k bernilai nol. Koefisien korelasi memiliki rentang nilai antara -1 sampai dengan 1, dimana jika dua variabel memiliki hubungan negatif sempurna akan memiliki ρ_k bernilai -1 sedangkan apabila dua variabel memiliki hubungan positif sempurna akan memiliki ρ_k bernilai 1. Fungsi autokorelasi (ACF) juga digunakan untuk mengidentifikasi model *moving average* (MA).

Selain fungsi autokorelasi, dalam *time series* terdapat fungsi autokorelasi parsial (*partial autocorrelation function*) atau PACF. PACF digunakan untuk mengukur korelasi antara X_t dan X_{t+k}, ketika efek dari rentang atau jangka waktu *time lag* dihilangkan. Seperti ACF, PACF juga memiliki nilai dalam rentang -1 sampai

dengan 1. Jika ACF digunakan untuk mengidentifikasi model MA, maka PACF digunakan untuk mengidentifikasi model *autoregressive* (AR). Fungsi PACF dapat didefinisikan sebagai berikut:

$$\emptyset_{kk} = \frac{\begin{vmatrix} 1 & \rho_1 & \rho_2 & \cdots & \rho_{k-2} & \rho_1 \\ \rho_1 & 1 & \rho_1 & \cdots & \rho_{k-3} & \rho_2 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \cdots & \rho_1 & \rho_k \end{vmatrix}}{\begin{vmatrix} 1 & \rho_1 & \rho_2 & \cdots & \rho_{k-2} & \rho_{k-1} \\ \rho_1 & 1 & \rho_1 & \cdots & \rho_{k-3} & \rho_{k-2} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \cdots & \rho_1 & 1 \end{vmatrix}} \tag{2.3}$$

dimana $\emptyset_{kk}$ merupakan fungsi dari k, himpunan $\{\emptyset_{kk}\,; k = 0,1,2, \dots\}$ dinamakan PACF (Wei, 2006)

2.2.3 Proses White Noise

Menurut Wei (2006), sebuah proses $\{e_t\}$ dikatakan *white noise* jika merupakan serangkaian variabel acak yang tidak memiliki korelasi dan berdistribusi tertentu dengan rata-rata tetap $E(e_t) = \mu$ biasanya bernilai nol, $Var(e_t) = \sigma^2$ dan $Cov(e_t, e_{t+k}) = 0$ untuk semua $k \neq 0$. Dengan demikian proses dari *white noise* $\{e_t\}$ adalah stasioner dengan fungsi autokovarians dan konsep dasar dari proses *white noise* adalah bahwa ACF dan PACF bernilai nol.

2.3 Knowledge Discovery in Database (KDD)

Knowledge Discovery in Database dan *data mining* sering kali digunakan secara bergantian untuk menjelaskan proses penggalian atau penarikan informasi dalam suatu data tertentu, padahal kedua istilah ini merupakan dua hal yang berbeda secara konsep. Menurut Fayyad (1996), proses KDD secara garis besar dapat dijelaskan sebagai berikut:

1. Seleksi Data

Pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai.

2. Preprocessing Data

Preprocessing data mencakup membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak (tipografi).

3. Transformasi Data

Transformasi data digunakan untuk merubah skala pengukuran dari bentuk asli ke dalam bentuk lain sehingga data dapat digunakan untuk analisis dan asumsi-asumsi tertentu. Terdapat beberapa cara untuk merubah *dataset* yaitu dengan *min-max normalization, z-score normalization* dan *decimal scaling.* Proses *min-max normalization* akan menghasilkan nilai dengan rentang (0,1).

$$x^* = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{2.4}$$

dengan:

x^* : nilai x baru

x_{min} : nilai minimum dari x

x_{max} : nilai maksimum dari x

$$Z = \frac{X - \mu}{\sigma} \tag{2.5}$$

dengan:

x : nilai yang diamati

μ : rata-rata

σ : standar deviasi

$$v^* = \frac{v}{10^j} \tag{2.6}$$

dengan j merupakan bilangan bulat terkecil sehingga nilai maksimum $|v^*| < 1$.

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. Interpretasi/Evaluasi

Pola informasi yang dihasilkan dari proses perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan.

2.4 Statistical Learning

Statistical learning sebagai alat pembelajaran diklasifikasikan menjadi 3 bagian yaitu *supervised learning*, *semi supervised learning* dan *unsupervised learning* (James, et al., 2017).

1. *Supervised Learning*

Supervised learning atau proses pembelajaran terawasi merupakan pendekatan *machine learning* yang didefinisikan oleh penggunaan data terlabel. Sekumpulan data dirancang untuk mengawasi algoritma dalam mengklasifikasikan atau memprediksi data secara akurat. Dengan menggunakan *input* dan *output* berlabel, model dapat mengukur akurasinya serta dapat belajar dari waktu ke waktu.

2. *Unsupervised Learning*

Unsupervised learning merupakan kebalikan dari metode *supervised learning,* dimana pada metode *machine learning* ini data yang diolah tidak memiliki label dan system tidak mengetahui *output* yang benar. Tujuannya adalah untuk mengeksplorasi data dan menemukan struktur di dalamnya.

3. *Semi-supervised Learning*

Semi-supervised Learning dapat dikatakan sebagai kombinasi dari *supervised* dan *unsupervised learning*, dimana dalam *semi-supervised learning* dapat digunakan data berlabel maupun tanpa label. *Semi-supervised learning* sangat berguna ketika sulit untuk mendapatkan fitur yang relevan dari data dan ketika data yang dimiliki berjumlah besar. *Semi-supervised learning* sangat ideal untuk gambar medis, di mana sejumlah kecil data pelatihan dapat menyebabkan peningkatan akurasi yang signifikan. Misalnya, ahli radiologi dapat melabeli subset kecil *CT scan* untuk tumor atau penyakit sehingga mesin dapat memprediksi lebih akurat pasien mana yang mungkin membutuhkan lebih banyak perhatian medis.

Sedangkan, proses evaluasi dalam *machine learning* dibagi menjadi 2 bagian yaitu proses *training* dan *testing*. Dalam proses *training* algoritma *machine learning* akan merubah parameter pada dirinya untuk menyesuaikan dengan data yang diberikan sehingga *machine learning* dapat menghasilkan suatu informasi dari data yang telah diberikan. Setelah itu, performa algoritma akan diuji menggunakan *testing set* (data *testing*) untuk mengevaluasi performa dari data tersebut.

2.5 Fungsi Aktivasi

Fungsi aktivasi merupakan fungsi matematis yang digunakan untuk mendapatkan nilai *output* berdasarkan nilai input, seperti namanya fungsi ini dapat mengaktifkan neuron. Kuncinya di sini adalah bahwa informasi diproses melalui fungsi aktivasi. Setiap fungsi aktivasi meniru neuron otak karena mereka tergantung pada kekuatan sinyal input. Hasil dari pemrosesan ini kemudian ditimbang dan didistribusikan ke neuron di lapisan berikutnya. Intinya, neuron saling mengaktifkan melalui jumlah tertimbang. Ini memastikan kekuatan koneksi antara dua neuron berukuran sesuai dengan berat informasi yang diproses (Lewis, 2017).

Menurut Fausett (1994), fungsi aktivasi yang digunakan pada algoritma *backpropagation* harus memiliki beberapa karakteristik penting yaitu kontinu, terdifferensial dan monoton. Jaringan saraf tiruan memiliki beberapa fungsi aktivasi seperti fungsi sigmoid, fungsi tanh, fungsi linear, fungsi biner, fungsi hard limit, fungsi ReLU, fungsi leaky ReLU, dan sebagainya. Fungsi aktivasi yang akan digunakan pada kasus ini yaitu fungsi aktivasi sigmoid, tanh, ReLU dan leaky ReLU.

2.5.1 Fungsi Aktivasi Sigmoid

Fungsi sigmoid (atau logistik) adalah pilihan populer. Fungsi sigmoid mengambil angka dalam kisaran antara 0 dan 1. Secara khusus, angka negatif besar menjadi 0 dan angka positif besar menjadi 1 (Lewis,2017).

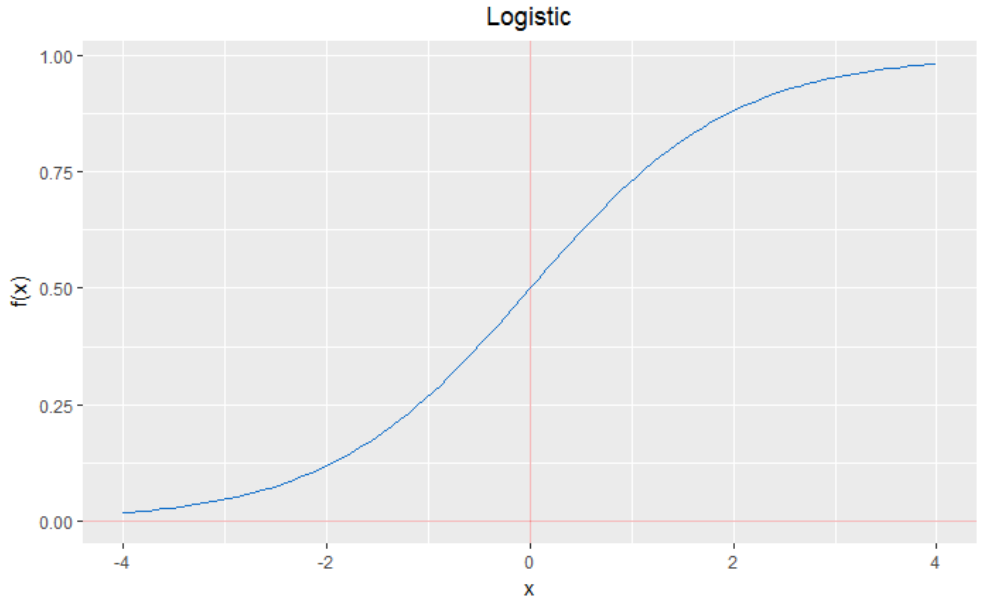
$$Sigmoid(x) = \frac{1}{1+e^{-x}} \tag{2.7}$$

dengan menggunakan aturan turunan

$$f'(x) = \frac{u(x)}{v(x)} = \frac{u'(x)v(x) - u(x)v\prime(x)}{[v(x)]^2}$$

$$\begin{aligned}
\frac{d}{dx}S(x) &= \frac{d}{dx}\frac{1}{1+e^{-x}} \\
&= \frac{e^{-x}}{(1+e^{-x})^2} \\
&= \frac{1-1+e^{-x}}{(1+e^{-x})^2} \\
&= \frac{1+e^{-x}}{(1+e^{-x})^2} - \frac{1}{(1+e^{-x})^2} \\
&= \frac{1}{(1+e^{-x})} - \frac{1}{(1+e^{-x})^2} \\
&= \frac{1}{(1+e^{-x})}\left(1 - \frac{1}{(1+e^{-x})}\right) \\
&= S(x)(1 - S(x))
\end{aligned} \tag{2.8}$$

Jika digambarkan menjadi sebuah grafik, fungsi sigmoid akan membentuk huruf "S".



Gambar 2. Fungsi Aktivasi Sigmoid.

2.5.2 Fungsi Aktivasi TanH

Fungsi aktivasi alternatif dari sigmoid adalah fungsi aktivasi tanh. Seperti pada sigmoid, fungsi tanh juga berbentuk "S" tetapi fungsi ini menghasilkan nilai -1 hingga 1. Maka dapat dikatakan bahwa fungsi *tanh* hampir sama dengan fungsi sigmoid, namun memiliki rentang nilai yang lebih luas sehingga lebih efektif untuk pemodelan nonlinear yang kompleks. Bentuk dari fungsi tanh adalah sebagai berikut:

$$Tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{2.9}$$

$$Tanh(x) = \frac{sinh\ (x)}{cosh\ (x)} \tag{2.10}$$

dengan menggunakan fungsi turunan

$$f'(x) = \frac{u(x)}{v(x)} = \frac{u'(x)v(x)-u(x)v\prime(x)}{[v(x)]^2}$$

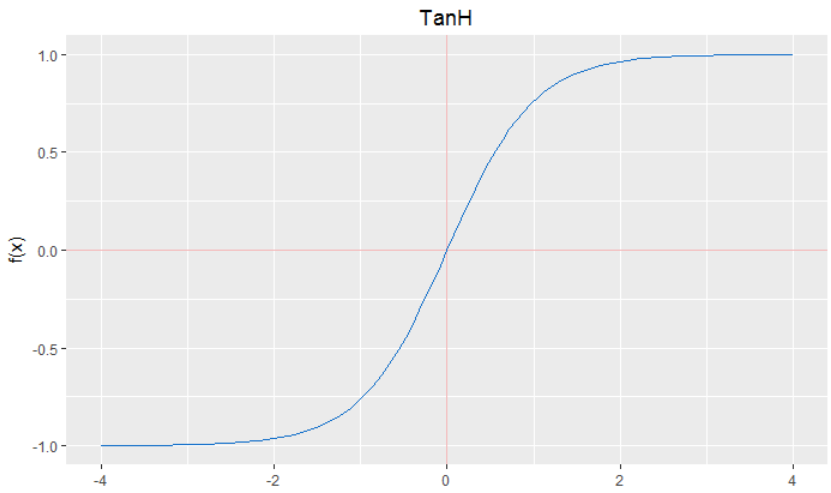
$$\frac{d}{dx}tanh(x) = \frac{d}{dx}\frac{sinh\ (x)}{cosh\ (x)}$$

$$= \frac{cosh^2(x)-sinh^2(x)}{cosh^2(x)}$$

$$= \frac{cosh^2(x)}{cosh^2(x)} - \frac{sinh^2(x)}{cosh^2(x)}$$

$$= 1 - tanh(x) \qquad (2.11)$$

Jika digambarkan menjadi sebuah grafik, fungsi tanh akan membentuk huruf "S" seperti fungsi aktivasi sigmoid.



Gambar 3. Fungsi Aktivasi TanH.

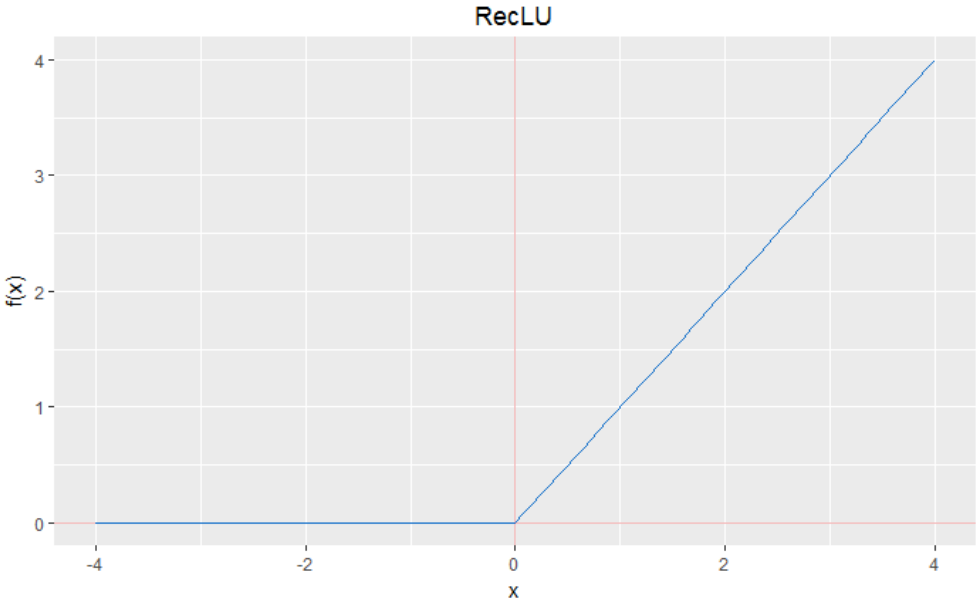
2.5.3 Fungsi Aktivasi *Rectified Linear Unit* (ReLU)

ReLU (*Rectified Linear Unit)* merupakan fungsi aktivasi yang digunakan untuk menormalisasikan nilai yang dihasilkan *layer.*

$$f(x) = \max(0, x) \qquad (2.12)$$

Dari persamaan (2.12), diketahui bahwa fungsi aktivasi ReLU akan menghasilkan nilai $f(x) = 0$ apabila x bernilai 0 atau kurang dari 0. Sehingga setiap input negatif yang diberikan akan selalu dipetakan menjadi 0. Konvergensi proses pelatihan jika

menggunakan fungsi ReLU lebih cepat hingga 6 kali dibandingkan fungsi tanh (Krizhevsky, 2012)

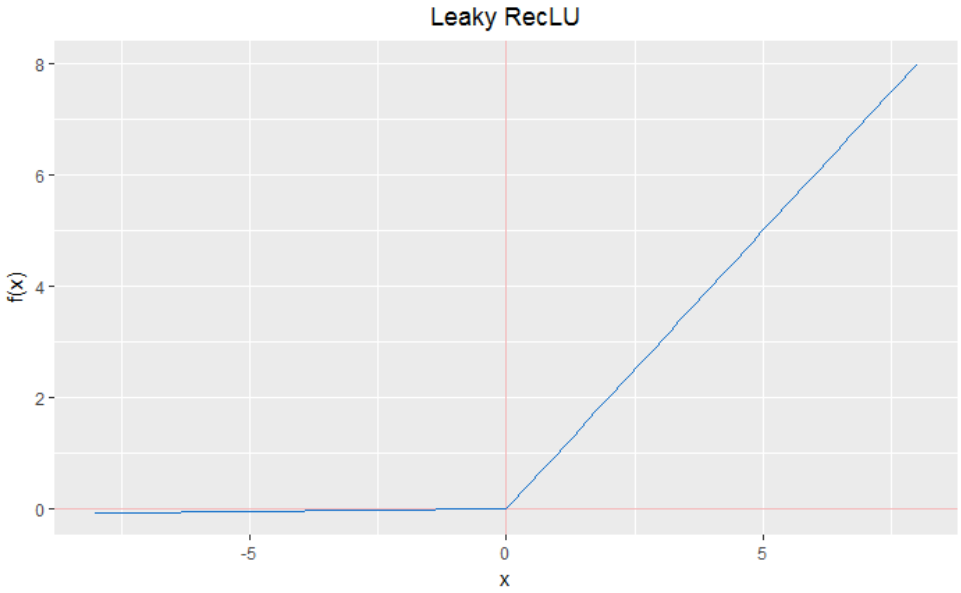


Gambar 4. Fungsi Aktivasi ReLU.

2.5.4 Fungsi Aktivasi *Leaky ReLU*

Menurut Sharma et al (2020), leaky ReLU merupakan versi improvisasi dari fungsi ReLU di mana untuk nilai negatif x, alih-alih mendefinisikan nilai fungsi ReLU sebagai nol, didefinisikan sebagai komponen linier yang sangat kecil dari x. fungsi leaky ReLU dapat diekspresikan secara matematis sebagai:

$$f(x)\begin{cases}0{,}01x; x < 0\\ x; x \geq 0\end{cases} \qquad (2.13)$$



Gambar 5. Fungsi Aktivasi Leaky ReLU.

2.6 Artificial Neural Network

2.6.1 Konsep Dasar Artificial Neural Network

Sejarah perkembangan *artificial neural network* atau biasa disebut jaringan saraf tiruan, telah berkembang sejak tahun 1940 dimana para ilmuan mulai menemukan bahwa psikologi dari otak sama dengan metode pemrosesan yang dilakukan komputer. Jaringan saraf tiruan merupakan salah satu algoritma klasifikasi yang menerapkan prinsip kerja jaringan saraf manusia. Sama hal nya seperti sistem saraf manusia, jaringan ini juga memiliki neuron yang berfungsi sebagai penghantar informasi. Sesuai dengan prinsip kerjanya, jaringan ini terdiri dari tiga lapisan yaitu lapisan *input,* lapisan tersembunyi dan lapisan *output.* Pada masing-masing lapisan, akan diberikan bobot yang berfungsi untuk mentransformasi nilai input menjadi nilai *output.*

Menurut Ripley (1996), jaringan saraf adalah prosesor terdistribusi paralel besar-besaran yang memiliki kecenderungan alami untuk menyimpan pengetahuan pengalaman dan membuatnya tersedia untuk digunakan. Ini menyerupai otak dalam dua hal:

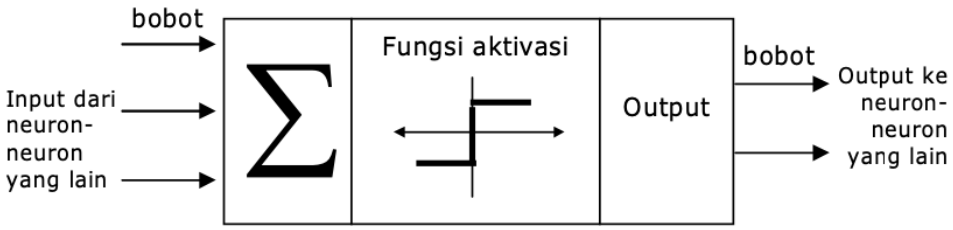
1. Pengetahuan diperoleh oleh jaringan melalui proses pembelajaran.
2. Kekuatan koneksi interneuron yang dikenal sebagai bobot sinaptik digunakan untuk menyimpan pengetahuan.

Secara umum, jaringan saraf tiruan ditentukan oleh 3 hal:

1. Arsitektur jaringan, sebagai pola hubungan antar neuron.
2. Algoritma, merupakan metode untuk menentukan bobot penghubung atau bisa juga disebut proses *training*/*learning*
3. Fungsi aktivasi

2.6.2 Struktur Artificial Neural Network

Jaringan saraf tiruan memiliki beberapa jenis yang berbeda, namun hampir semua jenis memiliki komponen yang sama. Seperti pada otak manusia, jaringan saraf tiruan memiliki *neuron.* Jika pada otak manusia terdapat dendrit yang memiliki fungsi sebagai penerima rangsangan, pada jaringan saraf terdapat lapisan input. Informasi yang didapat oleh lapisan input, akan diteruskan melalui *neuron* dengan bobot tertentu.

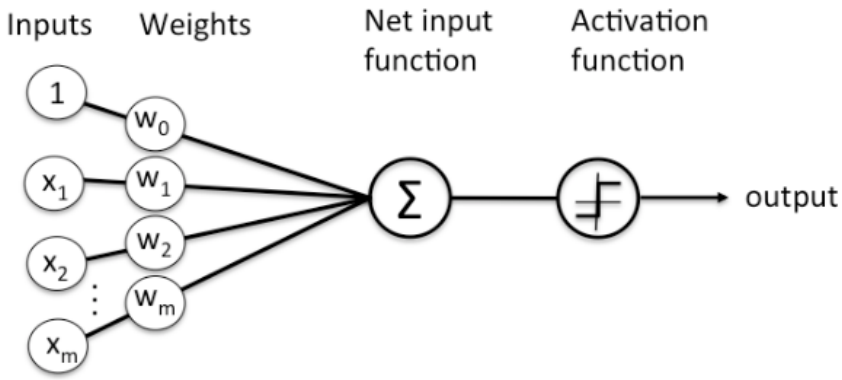


Gambar 6. Alur *Neural Network*.

Pada Gambar 6. terlihat bahwa input ini akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang yang datang. Hasil penjumlahan ini kemudian akan dibandingkan dengan suatu nilai ambang batas (*threshold)* tertentu melalui fungsi aktivasi setiap *neuron*. Apabila input tersebut melewati suatu nilai ambang tertentu, maka neuron tersebut akan diaktifkan, tetapi jika tidak, maka *neuron* tersebut tidak akan diaktifkan. Apabila *neuron* tersebut diaktifkan, maka *neuron* tersebut akan mengirimkan *output* melalui bobot-bobot *output* nya ke semua *neuron* yang berhubungan dengannya.

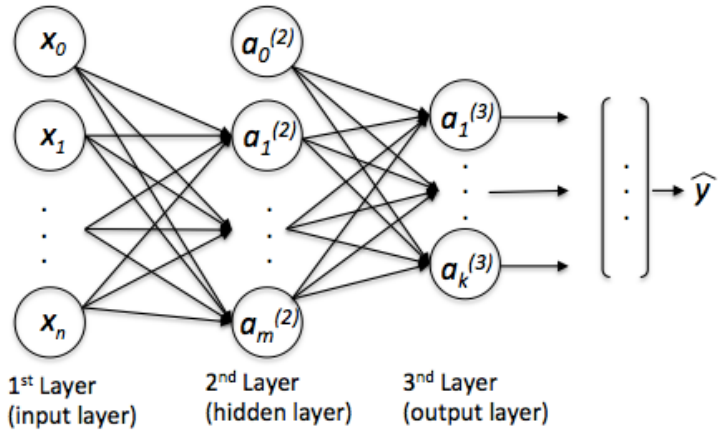
2.6.3 Perceptron

Menurut Shynk (1990), *perceptron* merupakan suatu kombinasi linear yang memetakan *output* ke satu dari dua nilai diskrit. Hubungan antara bobot dan *threshold* dapat diperbaiki atau diadaptasi menggunakan berbagai algoritma pembelajaran. Dalam *neural network,* terdapat dua jenis perceptron yaitu *single-layer perceptron* dan *multi-layer perceptron*. *Single-layer perceptron* adalah sebuah jaringan saraf tiruan yang hanya terdiri dari *layer input* dan *output.*



Gambar 7. *Single-Layer Perceptron.*

Pada *single-layer perceptron,* sinyal input x_m diskalakan oleh satu set bobot yang telah disesuaikan w_m untuk menghasilkan *output* y. Sedangkan, *multi-layer perceptron* merupakan varian lain dari *perceptron* yang memiliki satu atau lebih *hidden layer* antara *input layer* dan *output layer* (Ramchoun dkk, 2016)



Gambar 8. *Multi-Layer Perceptron.*

2.7 Algoritma Backpropagation

Algoritma *backpropagation* merupakan salah satu algoritma dalam *multi-layer perceptron.* Pelatihan *backpropagation* meliputi 3 fase yaitu fase maju, fase mundur, dan fase modifikasi bobot. Algoritma ini merupakan salah satu metode pembelajaran dengan supervisi (*supervsied learning*).

Menurut Kulkarni (2011), *backpropagation* hanyalah sebuah implementasi dari *gradient descent*. Algoritma ini dapat digunakan apabila diketahui turunan dari *error* yang berhubungan dengan bobot. Jika dinotasikan E_m merupakan *error* pada *training* m, *gradient descent* menunjukkan bahwa diperbaharui bobot W_{ij}dengan jumlah berikut (dimana η adalah konstanta).

$$\Delta w_{ij} = -\eta \frac{\partial E_m}{\partial w_{ij}} \quad (2.14)$$

dengan:

Δw_{ij} : perubahan bobot sambungan dari unit i ke unit j

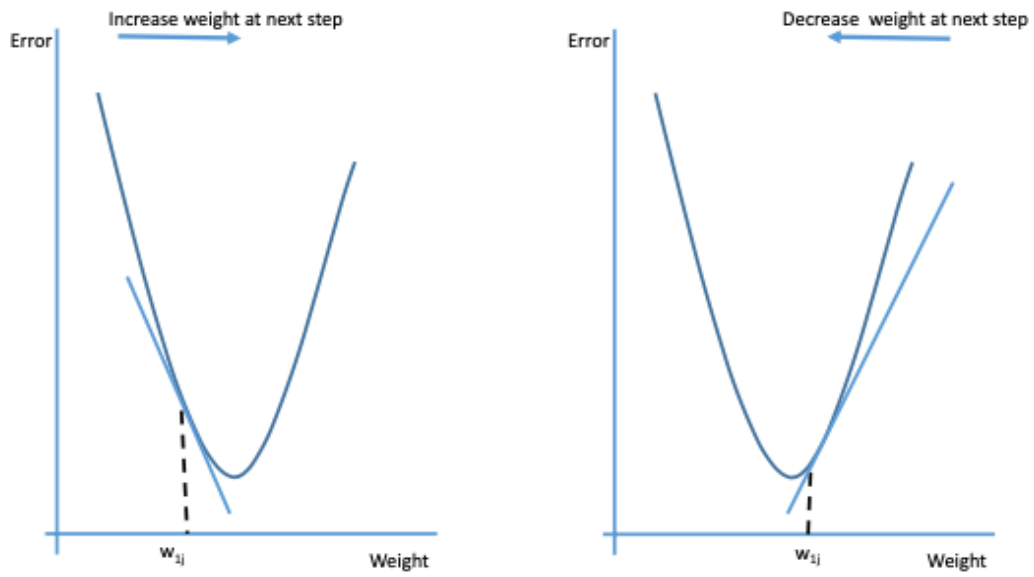
η : *learning rate*

∂E_m : turunan dari *error* pada *training m*

∂w_{ij} : turunan dari bobot

2.7.1 Gradient Descent

Menurut Lewis (2017), *gradient descent* merupakan salah satu algoritma popular yang digunakan untuk mengoptimasikan / melakukan optimasi pada *neural network* dengan cara melakukan perubahan (*update*) bobot dan bias secara berulang/iteratif berdasarkan arah gradien dari *loss function* nya hingga didapatkan nilai yang minimum.



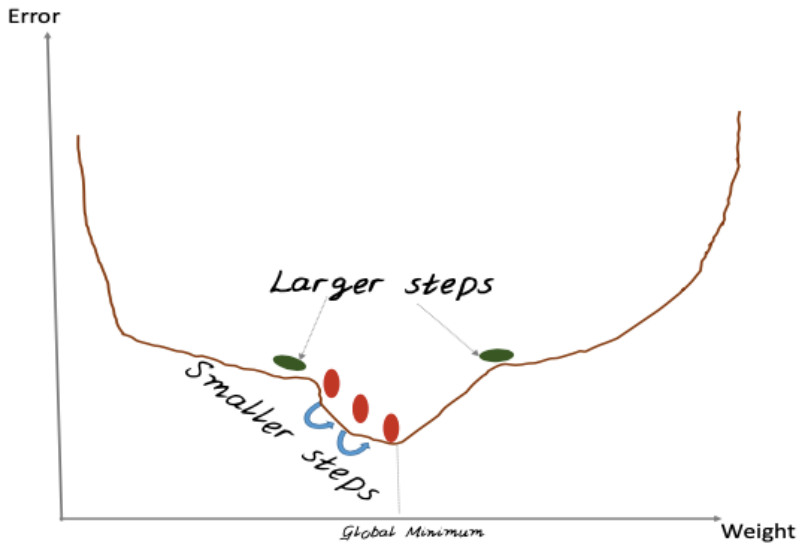
Gambar 9. Konsep Dasar *Gradient Descent.*

Dengan melakukan perubahan bobot dan bias secara iteratif sehingga nilai minimum dapat dicapai, atau dapat dilihat pada Gambar 9. konsep dasar pada *gradient descent* yaitu:

- Jika turunan parsial bernilai negatif, maka bobot akan meningkat (terlihat pada sisi kiri dari Gambar)
- Jika turunan parsial bernilai positif, maka bobot akan menurun (terlihat pada sisi kanan dari Gambar)
- Parameter *learning rate* berperan dalam menentukan ukuran dari iterasi yang digunakan hingga mencapai nilai minimum.

2.7.2 Learning Rate

Learning rate merupakan salah satu parameter yang dapat memberikan efek secara signifikan dalam *neural network*. Secara khusus, menggunakan nilai *learning rate* yang kecil akan mempercepat konvergensi terutama dalam kasus yang kompleks. *Learning rate* juga secara langsung mempengaruhi kecepatan proses *training* (Wilson and Martinez, 2001).



Gambar 10. Ilustrasi *Error* Berdasarkan *Learning Rate.*

Berdasarkan Gambar 10. dapat dilihat bahwa jika menggunakan *learning rate* yang terlalu besar maka akan menyebabkan perubahan yang besar pula, sehingga dapat menyebabkan terlewatnya *global minimum* yang diinginkan. Namun, jika menggunakan nilai *learning rate* yang terlalu kecil, akan menyebabkan proses yang sangat lambat untuk mencapai *global minimum.* Oleh karena itu, harus ditentukan nilai *learning rate* yang sesuai untuk mencapai *global minimum* yang diinginkan.

2.7.3 Epoch, Batch dan Iterasi

Dalam *machine learning* terdapat istilah *Epoch, Batch dan* Iterasi, lalu apakah perbedaan dari ketiga istilah tersebut. Dikatakan satu *Epoch* adalah ketika seluruh dataset sudah melalui proses *forward* dan *backward* dalam *neural network* sebanyak satu kali. Namun, karena satu *epoch* masih terlalu besar, maka akan dibagi lagi ke dalam satuan kecil yaitu *batch.*

Menurut Lewis (2017), proses *batching* adalah salah satu pendekatan umum untuk mempercepat komputasi jaringan saraf. Proses ini melibatkan perhitungan gradien pada beberapa contoh pelatihan (*batch*). Oleh karena itu, *batch* sendiri dapat diartikan sebagai jumlah sampel data yang disebarkan dalam *neural network. Batch* berisi beberapa contoh *training* dalam satu kali *forward pass / backward pass*. Untuk merasakan efisiensi komputasi *batching*, misalkan terdapat ukuran batch sebesar 500, dengan 1000 contoh *training*. Maka hanya diperlukan 2 iterasi untuk menyelesaikan 1 *epoch*.

2.7.4 Lapisan Tersembunyi (*Hidden Layer*)

Dikarenakan algoritma *backpropagation* merupakan bagian dari *multi layer perceptron,* maka dalam struktur jaringannya terdapat *hidden layer* yang harus diperhatikan. Jumlah *hidden layer* juga memberikan dampak yang cukup signifikan bagi nilai *error* yang dihasilkan dari proses *training* dan *testing* dalam

backpropagation. Seperti hal nya penggunaan1 *hidden layer,* tentu tidak akan sama dengan penggunaan 2 *hidden layer.* Penggunaan 2 *hidden layer* tentu akan menambah iterasi saat proses *looping* atau proses *backward.* Terdapat beberapa *rule* dalam menentukan *hidden layer* yang optimal dari berbagai peneliti.

Menurut Heaton (2017), kasus yang menggunakan lebih dari 2 *hidden layer* jarang terjadi dalam *deep learning*. 2 atau kurang dari 2 sudah cukup untuk sekumpulan data sederhana. Namun, dengan kumpulan data kompleks yang melibatkan data *time series* atau visi komputer, *layer* tambahan sangat membantu. Tabel 1. merangkum kemampuan beberapa arsitektur lapisan secara umum.

Tabel 1. Menentukan Jumlah *Hidden Layer*

Jumlah *hidden layer*	Hasil
0	Hanya mampu mewakili fungsi linear
1	Dapat memperkirakan fungsi apa pun yang berisi pemetaan berkelanjutan dari satu ruang terbatas ke ruang lain.
2	Dapat merepresentasikan batas keputusan sembarang terhadap akurasi sembarang dengan fungsi aktivasi rasional dan dapat memperkirakan pemetaan terhadap akurasi apa pun.
>2	*layer* tambahan dapat mempelajari representasi kompleks (semacam rekayasa fitur otomatis)

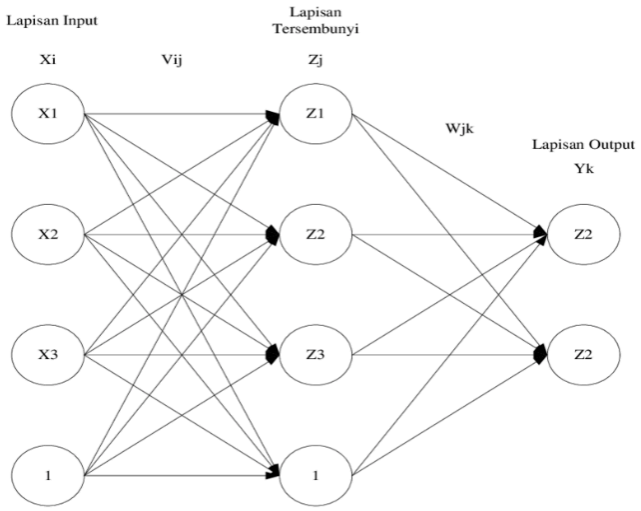
Selain menentukan jumlah *hidden layer*, perlu ditentukan juga seberapa banyak *nodes* atau neuron yang akan digunakan dalam *hidden layer* tersebut untuk memutuskan arsitektur jaringan secara keseluruhan. Menggunakan terlalu sedikit neuron di *hidden layer* akan menghasilkan sesuatu yang disebut *underfitting*. *Underfitting* terjadi ketika ada terlalu sedikit neuron di *hidden layer* untuk mendeteksi sinyal secara memadai dalam kumpulan data yang rumit.

Namun, menggunakan neuron dengan jumah banyak juga akan mengakibatkan masalah seperti *overfitting*. *Overfitting* terjadi ketika jaringan saraf memiliki begitu banyak kapasitas pemrosesan informasi sehingga jumlah informasi terbatas yang terkandung dalam set *training* tidak cukup untuk melatih semua neuron di *hidden layer.* Berdasarkan Jeff Heaton *rule* dalam (Heaton, 2017):

1. Jumlah *nodes* pada *hidden layer* harus berada di antara ukuran *input layer* dan *output layer*.
2. Jumlah *nodes* pada *hidden layer* harus bernilai $\frac{2}{3}$ dari jumlah *input layer* ditambah dengan *output layer* $\left(\frac{2}{3}(input + output)\right)$.
3. Jumlah *nodes* pada *hidden layer* harus kurang dari dua kali lipat jumlah *input layer.*

2.7.5 Propagasi Maju, Mundur dan Modifikasi Bobot

Algoritma *backpropagation* merupakan algoritma dengan *multi layer perceptron* yang artinya pada algoritma ini terdapat *layer input, hidden layer,* dan *layer output.* Secara umum arsitektur jaringan algoritma *backpropagation* dapat dilihat pada Gambar 11.

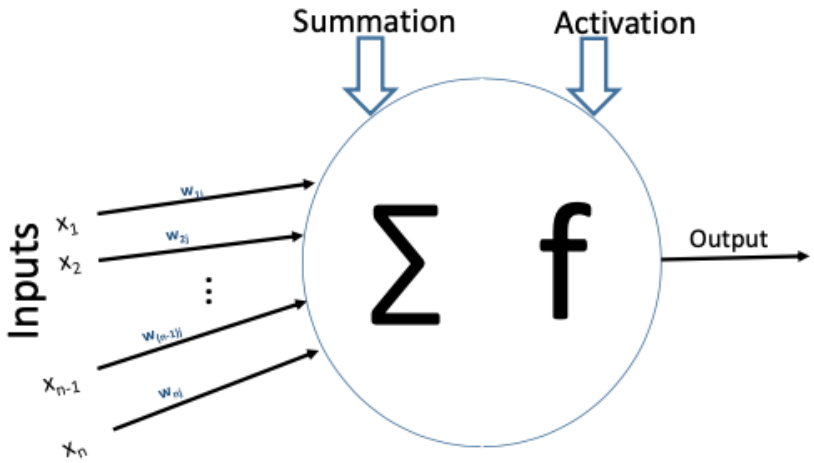


Gambar 11. Arsitektur Jaringan *Backpropagation.*

Dalam algoritma *backpropagation* terdapat 3 fase yaitu fase propagasi maju (*forward pass),* propagasi mundur (*backward pass)* dan *update* parameter bobot serta bias.

1. Propagasi maju (*forward pass*)

Secara umum proses *forward pass* dilakukan dengan meneruskan sinyal input (x_i) menuju *hidden layer* menggunakan fungsi aktivasi yang telah ditentukan. Keluaran dari unit tersembunyi (z_j) kemudian dipropagasi maju lagi ke *hidden layer* selanjutnya (jika ada) dengan fungsi aktivasi yang telah ditentukan, dan seterusnya hingga menghasilkan keluaran jaringan (y_k). Kemudian keluaran jaringan (y_k) dibandingkan dengan target yang ditentukan (t) dan *error* yang dihasilkan merupakan selisih antara (y_k) dan (t). Jika *error* yang dihasilkan lebih kecil dari batas toleransi (*threshold*).



Gambar 12. Proses Propagasi Maju (*Forward Pass*).

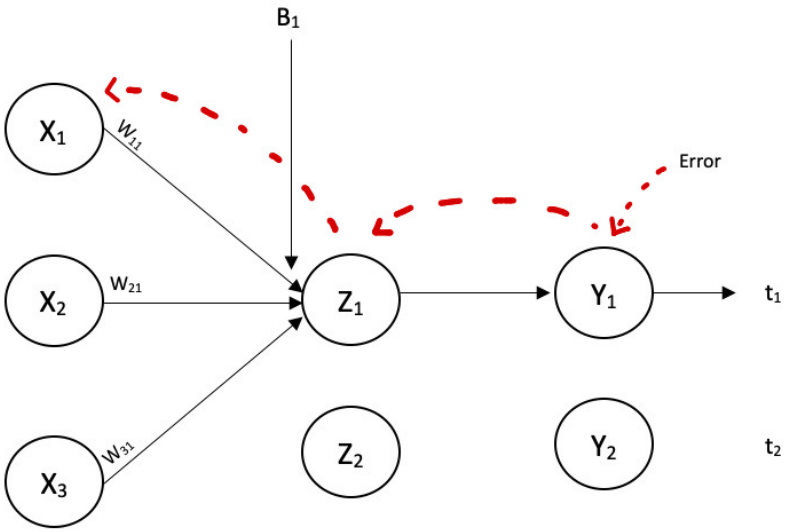
2. Propagasi mundur dan modifikasi bobot

Setelah didapatkan nilai *error,* dapat dilakukan optimasi menggunakan proses *gradient descent.* Bentuk utama untuk memperbaiki suatu bobot yaitu:

$$w_{baru} = w_{lama} - \eta \frac{\partial E}{\partial w} \tag{2.15}$$

Persamaan juga berlaku untuk memperbaiki nilai bias

$$b_{baru} = b_{lama} - \eta \frac{\partial E}{\partial b} \tag{2.16}$$



Gambar 13. Proses *Backward* dan Modifikasi Bobot.

Untuk menghitung $\frac{\partial E}{\partial w}$ dapat digunakan konsep *chaining,* proses *backward chaining* digambarkan dengan garis merah pada Gambar 13. Nilai E diperoleh dengan menggunakan rumus MSE (*mean square error*)

$$E = \frac{1}{2}(t_j - y_k)^2 \tag{2.17}$$

Pada persamaan tidak terdapat variabel w_{jk} tetapi variabel ini bisa diapatkan dengan melakukan *backward* atau "jalan mundur" dimana

$$y_k = \sigma(z_j) \tag{2.18}$$

Variabel y diperoleh dengan menerapkan sebuah fungsi aktivasi terhadap variabel z. Sedangkan untuk variabel z dapat dihitung dengan

$$z_j = \sum_{i=1}^{n} w_{ij} x_i + b_j \tag{2.19}$$

Pada persamaan terlihat bahwa variabel w_{jk} terdapat dalam perhitungan z_j yang secara tidak langsung berpengaruh terhadap nilai E. Hal ini yang disebut dengan konsep *chaining* atau rantaian. Setelah mengetahui hubungan antara E dan w_{jk} akan dilakukan proses turunan dengan menggunakan konsep turunan.

$$\frac{\partial f(x)}{\partial x} = \frac{\partial f(x)}{\partial z} \cdot \frac{\partial z}{\partial x}$$

$$\frac{\partial E}{\partial w_{ij}} = \left(\frac{\partial E}{\partial y_k}\right) \cdot \left(\frac{\partial y_k}{\partial w_{ij}}\right)$$

$$\frac{\partial E}{\partial y_k} = \frac{\partial}{\partial y_k}\left(\frac{1}{2}(t_j - y_k)^2\right)$$
$$= 2 * \frac{1}{2}(t_j - y_k)^{2-1} - 1 + 0$$
$$= -(t_j - y_k) \quad (2.20)$$

Pada persamaan y_k tidak terdapat variabel w_{ij}, oleh karena itu untuk menghitung $\frac{\partial y_k}{\partial w_{ij}}$ diperlukan persamaan (2.19)

Dengan menggunakan konsep turunan, maka didapatkan

$$\frac{\partial y_k}{\partial w_{ij}} = \frac{\partial y_k}{\partial z_j} \cdot \frac{\partial z_j}{\partial w_{ij}}$$
$$\frac{\partial y_k}{\partial z_j} = \sigma(z_j)\left(1 - \sigma(z_j)\right) = y_k(1 - y_k)$$
$$\frac{\partial z_j}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}}\left(\sum_{i=1}^{n} w_{ij} x_i + b_j\right)$$
$$= x_i \quad (2.21)$$

Dari perhitungan-perhitungan di atas, didapatkan

$$\frac{\partial E}{\partial w_{ij}} = -(t_j - y_k) * y_k(1 - y_k) * x_i \quad (2.22)$$

Sehingga untuk memodifikasi bobot dapat dilakukan dengan

$$w_{new} = w_{old} - \eta \frac{\partial E}{\partial w_{ij}} \quad (2.23)$$

Selain melakukan modifikasi bobot, akan ditentukan pula nilai modifikasi bias. Proses modifikasi nilai bias serupa dengan cara modifikasi nilai bobot menggunakan konsep *chaining* atau rantaian. Untuk memodifikasi bias b_j dapat dilakukan dengan persamaan

$$\frac{\partial E}{\partial b_j} = \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial b_j} \quad (2.24)$$

Dimana nilai dari $\frac{\partial E}{\partial y_k}$ telah diperoleh pada persamaan (2.20). Untuk menemukan nilai $\frac{\partial y_k}{\partial b_j}$ digunakan konsep turunan

$$\frac{\partial y_k}{\partial b_j} = \frac{\partial y_k}{\partial z_j} \cdot \frac{\partial z_j}{\partial b_j} \quad (2.25)$$

Dimana nilai dari $\frac{\partial y_k}{\partial z_j}$ telah diperoleh pada persamaan (2.21).

$$\frac{\partial z_j}{\partial b_j} = \frac{\partial\left(\sum_{i=1}^{n} w_{ij} x_i + b_j\right)}{\partial b_j} = 1 \quad (2.26)$$

Dari perhitungan-perhitungan di atas, didapatkan

$$\frac{\partial E}{\partial b_j} = \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial z_j} \cdot \frac{\partial z_j}{\partial b_j}$$
$$= -(t_j - y_k) * y_k(1 - y_k) * 1 \qquad (2.27)$$

Sehingga modifikasi bias b_j dapat dilakukan dengan

$$b_{new} = b_{old} - \eta \frac{\partial E}{\partial b_j} \qquad (2.28)$$

2.8 Evaluasi Kinerja Program dan Algoritma

Setelah dilakukan proses *training* dan *testing* menggunakan algoritma *backpropagation*, akan dilihat seberapa baik algoritma yang diguanakan dalam proses analisis. Proses evaluasi akan dilihat berdasarkan *running time* pada proses *training* dan akan dilihat *error* yang dihasilkan pada proses *testing.*

2.8.1 *Running Time* dan Spesifikasi Perangkat

Menurut Aho & Ullman (1992), setiap permasalahan yang dapat diselesaikan, dapat diselesaikan menggunakan lebih dari satu algoritma. Ketika sebuah program akan dijalankan berulang kali, efisiensi dan algoritma yang mendasarinya menjadi penting. Umumnya, akan dikaitkan antara efisiensi dan waktu yang dibutuhkan program untuk berjalan, meskipun ada sumber daya lain yang kadang-kadang harus dikonservasi oleh program, seperti:

1. Jumlah ruang penyimpanan yang diambil oleh variabelnya.
2. Jumlah *traffic* yang dihasilkan pada jaringan komputer.
3. Jumlah data yang harus dipindahkan ke dan dari *disk* komputer.

Dalam penelitian ini digunakan *notebook* atau laptop merek *apple* dengan jenis *macbook air,* dengan spesifikasi *hardware* sebagai berikut:

- *Processor name* : *dual-core intel core i5*
- *Processor speed* *:* 1,6 *Hz*
- *Memory* *:* SSD 128GB
- RAM : 4GB LPDDR3

2.8.2 Validasi Model

Menurut Hanke & Wichern (2014), teknik peramalan yang menggunakan data kuantitatif dengan data runtun waktu tertentu., terdapat *error* / kesalahan yang dihasilkan oleh teknik tersebut. Oleh sebab itu, dibutuhkan metode untuk mengukur seberapa besar *error* yang dapat dihasilkan oleh metode-metode forecasting untuk dipertimbangkan kembali sebelum dibuat keputusan. Metode-metode yang dapat digunakan dalam mengevaluasi *error* pada teknik forecasting yaitu MSE (*mean square error)*, RMSE *(root mean square error)*, MAD *(mean absolute derivation)*, MAPE *(mean absolute percentage error)* dan MPE *(mean percentage error)*.

Metode yang paling umum digunakan yaitu MSE *(mean square error)* dan RMSE *root mean square error)*. MSE mengukur rata-rata kuadrat dari kesalahan antara target yang diamati dan nilai yang diprediksi. Semakin kecil MSE, semakin baik hasil prediksi yang dihasilkan.

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(\hat{y}_t - y_t)^2 \tag{2.29}$$

dengan:

$\hat{y}_t$: nilai prediksi pada periode *t*

y_t : nilai aktual pada periode *t*

Karena MSE diukur dalam satuan kuadrat, maka RMSE dapat ditafsirkan sebagai jarak rata-rata, antara nilai yang diprediksi dan diamati, diukur dalam satuan variabel target.

$$RMSE = \sqrt{MSE} \tag{2.30}$$

2.9 Indeks Harga Saham Gabungan (IHSG)

Indeks Harga Saham Gabungan (IHSG) secara internasional disebut Indonesia composite index (IDI) atau IDX *Composite* merupakan salah satu indeks dalam Bursa Efek Indonesia (BEI). Indeks harga saham gabungan digunakan untuk melihat situasi dan perkembangan pasar modal secara keseluruhan dan bukan untuk melihat kondisi suatu perusahaan tertentu.

IHSG dapat dikatakan sebagai rata-rata harga dari seluruh harga saham perusahaan yang terdaftar dalam Bursa Efek. Apabila nilai IHSG naik, kemungkinan sebagian besar harga saham meningkat, namun bukan berarti tidak ada perusahaan yang mengalami penurunan harga saham. Maka dapat dikatakan bahwa fluktuasi IHSG desebabkan oleh fluktuasi harga-harga saham milik emiten yang terdaftar dalam BEI. Tidak hanya sebagai tolak ukur dalam kegiatan investasi, IHSG juga memiliki peranan yang besar dalam menunjukkan pertumbuhan ekonomi di Indonesia.

Metodologi perhitungan IHSG sama dengan cara menghitung indeks bursa saham lainnya, yaitu menggunakan rata-rata berdasarkan jumlah saham di bursa atau *Market Value Weighted Average Index.* IHSG didapatkan dengan persamaan:

$$IHSG = \frac{Nilai\ pasar}{Nilai\ dasar} x\ 100 \tag{2.31}$$

Dimana nilai dasar merupakan kumulatif jumlah saham pada hari dasar dikalikan dengan harga pada hari dasar. Sedangkan, nilai pasar merupakan kumulatif jumlah saham yang tercatat dikalikan dengan harga pasar.

$$Nilai\ pasar = p_1 q_1 + p_2 q_2 + \cdots + p_n q_n \tag{2.32}$$

dengan:

p_n : harga yang terjadi untuk emiten ke-n

q_n : jumlah saham yang digunakan untuk emiten ke-n

n : jumlah emiten yang tercatat di bursa efek

Menurut Sudirman (2015), secara garis besar terdapat 3 (tiga) faktor utama yang berpengaruh terhadap pergerakan IHSG yaitu: faktor domestik, faktor asing, dan

faktor aliran modal ke Indonesia. Faktor domestik berupa faktor-faktor fundamental suatu negara seperti inflasi, pendapatan nasional, jumlah uang yang beredar, suku bunga, maupun nilai tukar rupiah. Berbagai faktor fundamental tersebut dianggap dapat berpengaruh pada ekspektasi investor yang pada akhirnya berpengaruh terhadap pergerakan indeks. Faktor asing merupakan salah satu implikasi dari bentuk globalisasi dan semakin terintegrasinya pasar modal di seluruh dunia.

Seperti hal nya saat kasus *Corona Virus Desease-19* (COVID-19) pertama kali ditemukan di Indonesia, IHSG mengalami penurunan sejak bulan Maret 2020 dan kembali menguat di bulan November 2020. COVID-19 sangat memberikan dampak yang serius bagi pergerakan ekonomi di Indonesia. Oleh karena itu, teknik peramalan sangat dibutuhkan para investor untuk mengetahui nilai IHSG di masa yang akan datang berdasarkan nilai-nilai IHSG yang sudah ada di masa lalu.

III. METODOLOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

Penelitian ini dilakukan pada semester ganjil tahun akademik 2020/2021, bertempat di Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung.

3.2 Data Penelitian

Data yang digunakan adalah data sekunder yang diperoleh dari https://finance.yahoo.com/ mengenai data histori indeks harga saham gabungan selama 8 tahun yang terhitung sejak Januari 2013 sampai dengan Juni 2021.

3.3 Metode Penelitian

Dalam penelitian ini akan dicari model *neural network* untuk memprediksi harga penutupan IHSG menggunakan algoritma *backpropagation* dengan bantuan *software R-studio*. Selain itu akan dievaluasi berdasarkan nilai kesalahan terkecil dari beberapa fungsi aktivasi yang digunakan. Adapun langkah-langkah penelitian ini yaitu:

1. Memasukkan (input) data indeks harga saham gabungan yang didapatkan dari https://finance.yahoo.com/ ke dalam *software R-studio.*

2. Mengeskplorasi dan mempersiapkan data (mengubah data menjadi format *time series)*
3. Membuat plot dari data untuk meilhat pola data termasuk dalam jenis horizontal, musiman, siklis atau tren.
4. Melakukan *scaling* data.
5. Melakukan *training* dan *testing* data dengan jumlah tertentu menggunakan *package CaTools*.
6. Mengaplikasikan algoritma *backpropagation.*
7. Melakukan *testing* pada *output.*
8. Melihat validasi model.
9. Melihat tinggat akurasi dari model.
10. Menggunakan fungsi aktivasi alternatif untuk mendapatkan nilai *error* terkecil.
11. Membangun model *multi-layer perceptron* menggunakan *package Neuralnet.*
12. Denormalisasi hasil dari proses *neural network.*
13. Mengaplikasikan *neural network* untuk menemukan model *time series.*

Perhitungan algoritma *backpropagation* secara manual adalah sebagai berikut:

Langkah 1: Inisialisasi bobot dengan nilai *random* yang cukup kecil

- Fase Maju (*feed forward pass*)

Langkah 2: Lapisan *input* menerima sinyal $X_i, i = 1,2,3, \dots, n$ kemudian meneruskan sinyal tersebut ke lapisan berikutnya yaitu lapisan tersembunyi

Langkah 3: Tiap-tiap unit tersembunyi $Z_j, j = 1,2,3, \dots, p$ menjumlahkan sinyal-sinyal terbobot,

$$Z_{-in_j} = b1_j + \sum_{i=1}^{n} X_i W_{ij} \quad (3.1)$$

digunakan fungsi aktivasi untuk menghitung sinyal *output*.

$$Z_j = f(Z_{-in_j}) \quad (3.2)$$

dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya.

Langkah 4: Tiap unit *output* $Y_k, k = 1,2,3, \dots, m$ menjumlahkan sinyal-sinyal *input* terbobot,

$$Y_{-in_k} = b2_k + \sum_{i=1}^{p} Z_j W_{jk} \quad (3.3)$$

digunakan fungsi aktivasi untuk menghitung sinyal *output*.

$$Y_k = f(Y_{-in_k}) \quad (3.4)$$

- Fase Mundur (*backward pass*)

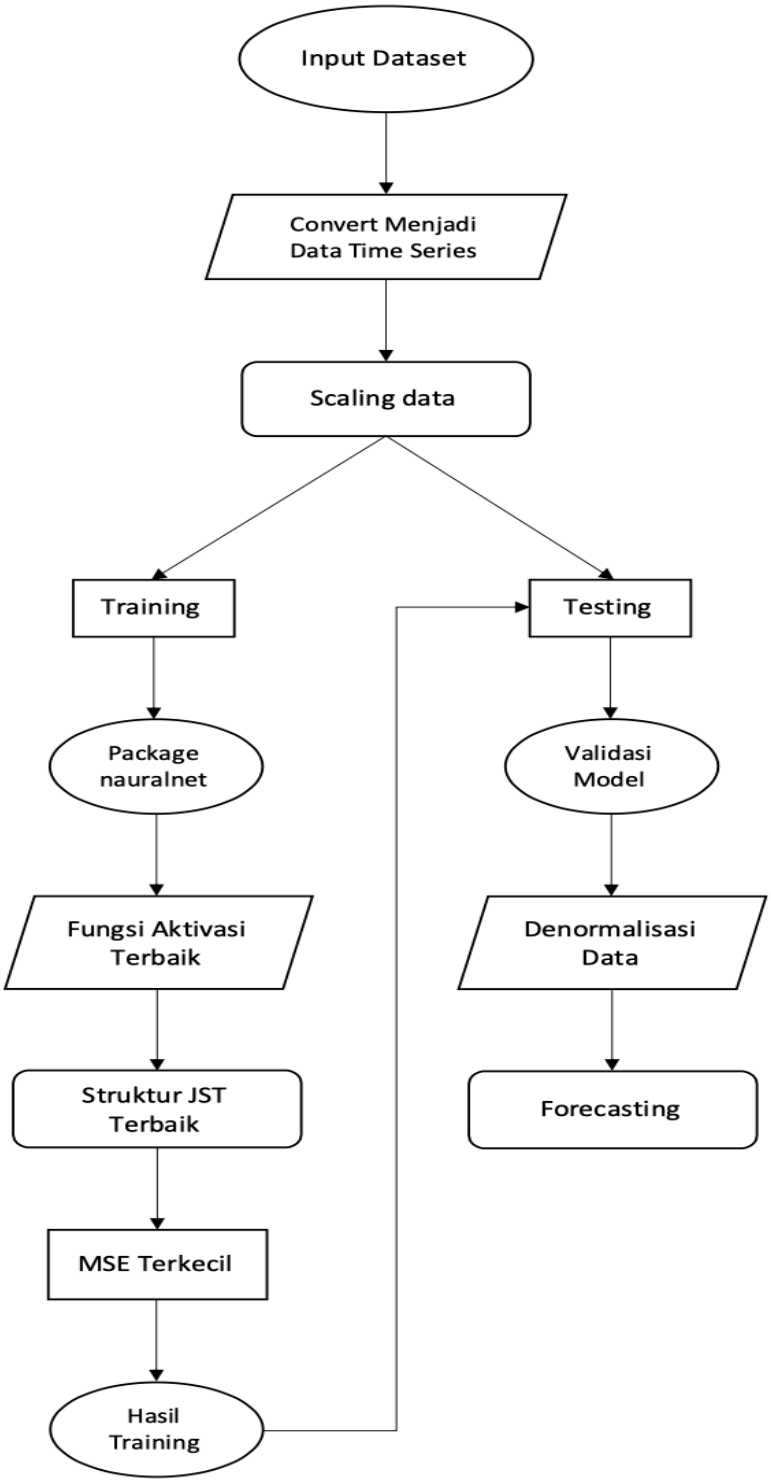
Langkah 5: Tiap unit *output* Y_k, $k = 1,2,3,\dots,m$ menerima target pola yang berhubungan dengan pola *input.*

Langkah 6: Tiap lapisan tersembunyi $Y_j, j = 1,2,3,\dots,p$ menghitung δ_j kemudian digunakan untuk menghitung bobot terkoreksi dan bias antara *input* dan *hidden layer.*

Langkah 7: Masing-masing lapisan *output* $Z_k, k = 1,2,3,\dots,m$ memperbaharui nilai bobot dan bias $j = 1,2,3,\dots,p$ dan setiap *hidden layer* memperbaharui pembobot dan bias $i = 1,2,3,\dots,n$ sehingga didapatkan bobot dan bias yang baru.

Langkah 8: Uji syarat (konvergen) terpenuhi, maka iterasi berakhir.

Secara singkat proses *neural network* menggunakan *software R-Studio* adalah sebagai berikut:



Gambar 14. *Flowchart* Proses *Neural Network* Menggunakan *R-Studio*

V. KESIMPULAN

1. Model *neural network* dengan menggunakan algoritma *backpropagation* diterapkan pada data harga penutupan IHSG selama 8 tahun yang terhitung sejak Januari 2013 sampai dengan Juni 2021 dengan variabel *input* yang digunakan yaitu data *lag* 1 dan *lag* 2 dari data IHSG. Diperoleh struktur jaringan terbaik berupa 2 *input layer,* 1 *hidden layer* berisi 2 *nodes* dengan menggunakan fungsi aktivasi leaky ReLU, nilai *threshold* sebesar 0,01 serta proses *scaling min-max normalization.* Model *neural network* menggunakan struktur jaringan terbaik yang telah ditentukan menghasilkan nilai MAPE sebesar 0,8686% atau dengan akurasi sebesar 99,1314%. Proses penentuan struktur jaringan terbaik dilakukan berdasarkan Heaton *rules.*
2. Dengan menggunakan *notebook* atau laptop merek *apple* dengan jenis *macbook air* dan spesifikasi *hardware* sebagai berikut:
   - *Processor name* : *dual-core intel core i5*
   - *Processor speed* *:* 1,6 *Hz*
   - *Memory* *:* SSD 128GB
   - RAM : 4GB LPDDR3

Didapatkan waktu rata-rata dan waktu maksimum dalam melakukan 45 kali proses *training* menggunakan *threshold* 0,01 adalah sebesar 6,565 detik dan 24,468, sedangkan waktu rata-rata dan waktu maksimum dalam melakukan 24 kali proses *training* menggunakan *threshold* 0,001 adalah sebesar 11,6304 detik dan 38,8760 detik. Semua proses *training* hanya memerlukan waktu kurang dari 1 menit.

DAFTAR PUSTAKA

Aho, A.V. and Ullman, J.D. 1992. *Foundation of Computer Science.* C Edition.

Fayyad, U., Shapiro, G.P., and Smyth, P. 1996. *Knowledge Discovery and Data Mining: Towards a Unifying Framework.* AAI Press.

Fausett, L. 1994. *Fundamentals of Neural Network: Architectures, Algorithms and Applications.* New Jersey: Prencite Hall.

Firdaus, M. 2004. *Ekonometrika: Suatu Pendekatan Aplikatif.* Bumi Aksara, Jakarta.

Hanke, J.E. and Wichern, D. 2014. *Business Forecasting.* 9th Edition. Pearson, USA.

Heaton, J. The Number of Hidden Layers. 2017.

Herjanto, E. 2008. *Manajemen Operasi.* Ed. Ke-3. Grasindo, Jakarta.

Hidayat, R. 2016. Prediksi Harga Saham Menggunakan Neural Network. Jurnal Gema Aktualia. **5**

James, G., *et al.* 2017. *An Introduction to Statistical Learning: Applications in R.* London: Springer.

Krizhevsky, A., Sutskever, I., and Hinton, G.E. 2012. ImageNet Classification with Deep Convulational Neural Network. Neural Information Processing System. **25**(2). Canada.

Kulkarni, S. and Harman, G. 2001. *An Elementary Introduction to Statistical Learning Theory.* Wiley and Sons, Canada.

Lawrence, R. 1997. Using Neural Networks to Forecast Stock Market Prices. Department of Computer Science, University of Manitoba.

Lewis, N.D. 2017. *Neural Network for Time Series Forecasting with R.* CrateSpace Independent Publishing Platform, US.

Makridakis, S., Wheelwright, S.C., and Hyndman, R.J. 1997. *Forecasting: Methods and Applications.* 3rd Edition. Wiley and Sons, Canada.

Ramchoun, H., *et al.* 2016. Multilayer Perceptron: Architecture Optimization and Training. International Journal of Interactive Multimedia and Artificial Intelligence. **4**(1).

Ripley, B.D. 1996. *Pattern Recognition and Neural Networks.* 1st Edition. Cambridge University Press, England.

Sudirman. 2015. *Pasar Modal dan Manajemen Portofolio.* Sultan Amai Press, Gorontalo.

Susanti, R. dan Adji, A.R. 2020. Analisis Peramalan IHSG dengan Time Series Modeling ARIMA. **17**(1).

Shynk, J.J. 1990. Performance Surface of a Single-Layer Perceptron. IEEE Transactions on Neural Networks. **1**(3).

Sharma, S., Sharma S., and Athaiya, A. 2020. Activation Function in Neural Networks. IJEAST. **4**(12):310-316.

Utami, T.W. dan Darsyah, M.Y. 2015. Peramalan Data Saham dengan Model Winter's. **3**(2). Semarang.

Wei, W.W.S. 2006. *Time Series Analysis: Univariate and Multivariate Methods.* 2nd Edition. New Jersey: Pearson Prentice Hall.

Wijiyanti, I., Anastasia, N., dan Gunawan, Y.W. 2005. Analisis Faktor Fundamental dan Risiko Sistematik Terhadap Harga Saham Properti di BEJ. *E*-jurnal Akutansi dan Keuangan. **5**(2).

Wilson, D.R., and Martinez, T.R. 2001. The Need for Small Learning Rates on Large Problems. Proceedings of the 2001 International Joint Conference on Neural Networks. 115-119.