

**PENERAPAN *ARTIFICIAL NEURAL NETWORK* (ANN) MENGGUNAKAN
ALGORITMA *BACKPROPAGATION* DENGAN MEMBANDINGKAN
EMPAT FUNGSI AKTIVASI DALAM MEMPREDIKSI HARGA EMAS**

(Skripsi)

Oleh

**RANTI VIDIA MAHYUNIS
NPM. 1817031076**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2022**

ABSTRACT

IMPLEMENTATION OF ARTIFICIAL NEURAL NETWORK (ANN) USING BACKPROPAGATION ALGORITHM BY COMPARING FOUR ACTIVATION FUNCTIONS IN PREDICTING GOLD PRICES

By

RANTI VIDIA MAHYUNIS

Gold is one of the main commodities traded in global financial markets and considered a safe asset. Gold price prediction is very important especially from a business perspective. The prediction of the gold price in this study is to determine the best ANN structure using Backpropagation Algorithm based on four activation functions. The goal is to produce good gold price prediction model with the smallest error difference and high accuracy. The activation functions used are the activation functions of Sigmoid, Tanh, ReLU, and Linear and learning rate values are 0.01 and 0.001. The structure of best ANN model obtained is in the form of four nodes in the input layer, one hidden layer with four nodes and the output layer using a Linear activation function and a learning rate of 0.01. Based on this structure, MSE value is 0.00051, MAPE value is 1.9798% and the accuracy is 98%.

Keywords: *Artificial Neural Network, Backpropagation, Gold Price Prediction, Activation Function, Model Structure*

ABSTRAK

PENERAPAN *ARTIFICIAL NEURAL NETWORK* (ANN) MENGGUNAKAN ALGORITMA *BACKPROPAGATION* DENGAN MEMBANDINGKAN EMPAT FUNGSI AKTIVASI DALAM MEMPREDIKSI HARGA EMAS

Oleh

RANTI VIDIA MAHYUNIS

Emas adalah salah satu komoditas utama yang diperdagangkan di pasar keuangan global dan dianggap sebagai aset yang aman. Prediksi harga emas sangat penting terutama dari perspektif bisnis. Prediksi harga emas dalam penelitian ini adalah menentukan struktur ANN terbaik menggunakan Algoritma *Backpropagation* berdasarkan empat fungsi aktivasi. Tujuannya menghasilkan model prediksi harga emas yang baik dengan selisih *error* terkecil dan akurasi yang tinggi. Fungsi aktivasi yang digunakan yaitu fungsi aktivasi *Sigmoid*, *Tanh*, *ReLU*, dan *Linear* serta nilai *learning rate*, yaitu 0,01 dan 0,001. Struktur model ANN terbaik yang diperoleh berupa empat *node* pada *input layer*, satu *hidden layer* dengan empat *node* dan *output layer* dengan menggunakan fungsi aktivasi *Linear* dan *learning rate* sebesar 0,01. Berdasarkan struktur tersebut, diperoleh nilai MSE sebesar 0.00051, nilai MAPE sebesar 1,9798% dan akurasi sebesar 98%.

Kata Kunci: *Artificial Neural Network*, *Backpropagation*, Prediksi Harga Emas, Fungsi Aktivasi, Struktur Model

**PENERAPAN *ARTIFICIAL NEURAL NETWORK* (ANN) MENGGUNAKAN
ALGORITMA *BACKPROPAGATION* DENGAN MEMBANDINGKAN
EMPAT FUNGSI AKTIVASI DALAM MEMPREDIKSI HARGA EMAS**

Oleh

RANTI VIDIA MAHYUNIS

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA MATEMATIKA**

Pada

**Jurusan Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2022**

Judul Skripsi : **PENERAPAN *ARTIFICIAL NEURAL NETWORK* (ANN) MENGGUNAKAN ALGORITMA *BACKPROPAGATION* DENGAN MEMBANDINGKAN EMPAT FUNGSI AKTIVASI DALAM MEMPREDIKSI HARGA EMAS**

Nama Mahasiswa : **Ranti Vidia Mahyunis**

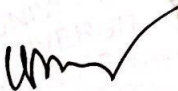
Nomor Pokok Mahasiswa : **1817031076**

Jurusan : **Matematika**

Fakultas : **Matematika dan Ilmu Pengetahuan Alam**

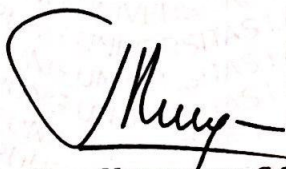


1. **Komisi Pembimbing**


Ir. Warsono, M.S., Ph.D.
NIP 19630216 198703 1 003


Dr. Aang Nuryaman, S.Si., M.Si.
NIP 19740316 200501 1 001

2. **Ketua Jurusan Matematika**


Dr. Aang Nuryaman, S.Si., M.Si.
NIP 19740316 200501 1 001

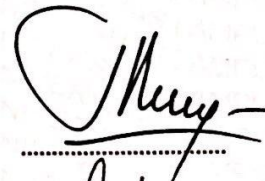
MENGESAHKAN

1. Tim Penguji

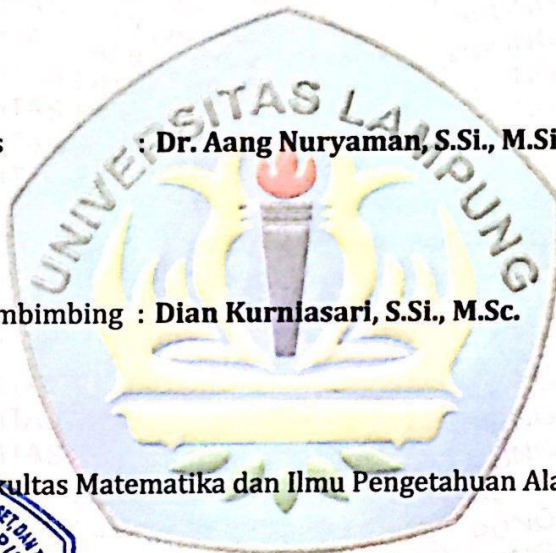
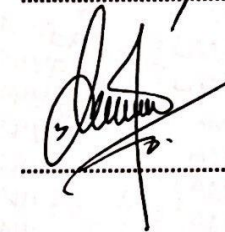
Ketua : Ir. Warsono, M.S., Ph.D.



Sekretaris : Dr. Aang Nuryaman, S.Si., M.Si.



**Penguji
Bukan Pembimbing : Dian Kurniasari, S.Si., M.Sc.**



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Dr. Eng. Sumpito Dwi Yuwono, M.T.

NIP. 19740705 200003 1 001

Tanggal Lulus Ujian Skripsi : 15 Agustus 2022

PERNYATAAN SKRIPSI MAHASISWA

Yang bertanda tangan di bawah ini:

Nama : Ranti Vidia Mahyunis
Nomor Pokok Mahasiswa : 1817031076
Jurusan : Matematika
Judul Skripsi : **Penerapan *Artificial Neural Network* Algoritma *Backpropagation* dengan Membandingkan Empat Fungsi Aktivasi dalam Memprediksi Harga Emas**

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, 15 Agustus 2022
Penulis,



Ranti Vidia Mahyunis

RIWAYAT HIDUP

Penulis bernama lengkap Ranti Vidia Mahyunis, lahir di Bandar Lampung pada tanggal 07 Februari 2000. Penulis merupakan anak ke-tujuh dari tujuh bersaudara yang lahir dari pasangan Bapak Mahyunis dan Ibu Marsiti.

Penulis memulai pendidikan di TK Nurul Islam pada tahun 2005-2006. Kemudian melanjutkan sekolah ke sekolah dasar di SD Negeri 3 Sukaraja pada tahun 2006-2012, sekolah menengah pertama di SMP Negeri 23 Bandar Lampung pada tahun 2012-2015, dan sekolah menengah atas di SMA Negeri 1 Bandar Lampung pada tahun 2015-2018. Pada tahun 2018 penulis diterima sebagai mahasiswa S1 di Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam melalui jalur SBMPTN (Seleksi Bersama Masuk Perguruan Tinggi Negeri).

Selama menjadi mahasiswa, penulis mengikuti organisasi dalam kampus antara lain menjadi anggota ESO Universitas Lampung periode kepengurusan 2019, Sekretaris Bidang Minat dan Bakat HIMATIKA periode 2020, dan Dewan Pembina Organisasi HIMATIKA periode 2021.

Pada Bulan Februari-Maret 2021, penulis melakukan Kerja Praktik (KP) di Bank BRI Kantor Cabang Teluk Betung Bandar Lampung di bagian Operasional *Back Office*. Sedangkan sebagai bentuk pengabdian mahasiswa kepada masyarakat, penulis melaksanakan Kuliah Kerja Nyata (KKN) periode kedua di Kelurahan Keteguhan, Kecamatan Teluk Betung Timur, Kota Bandar Lampung.

KATA INSPIRASI

“Dan karena rahmat-Nya, Dia jadikan untukmu malam dan siang, supaya kamu beristirahat pada malam itu dan supaya kamu mencari kebahagiaan dari karunia-Nya (pada siang hari) dan agar kamu bersyukur kepada-Nya.”

(Q.S. Al-Qashash: 73)

“Hai orang-orang yang beriman, mintalah pertolongan dengan sabar dan shalat. Sesungguhnya Allah beserta orang-orang yang sabar”

(Q.S. Al-Baqarah: 153)

“Ketahuilah bahwa kemenangan bersama kesabaran, kelapangan bersama kesempatan, dan kesulitan bersama kemudahan.”

(HR. Tirmizi)

“And when I’m hopeless, Allah is my only hope for holding on”

PERSEMBAHAN

Puji syukur senantiasa kita panjatkan kehadirat Allah SWT. yang telah memberikan nikmat, rahmat, taufik dan hidayah-Nya sehingga skripsi ini dapat terselesaikan dengan baik. Serta sholawat dan salam yang selalu tercurahkan kepada junjungan kita Nabi Muhammad SAW. Dengan penuh ketulusan serta rasa syukur, saya mempersembahkan karya ini kepada:

Bapak Mahyunis dan Ibu Marsiti (almh)

Segala perjuangan dan perjalanan hidup hingga titik ini saya persembahkan pada dua orang paling berharga dalam hidup saya. Terima kasih ayah atas doa, dukungan, dan nasihat yang selalu diberikan. Terima kasih ibu atas cinta dan kasih sayang yang selalu hadir dan selalu menjadi penguat untuk anakmu. Terima kasih telah menjadi orang tua yang sempurna. Terima kasih karena selalu ada untukku.

Dosen Pembimbing dan Penguji

Terima kasih kepada dosen pembimbing dan pembahas yang sudah sangat berjasa dalam membantu, memberikan arahan, masukan dan ilmu yang sangat berharga.

Sahabat-sahabatku

Terima kasih atas inspirasi, semangat, dorongan, dan dukungan yang telah kalian berikan. Terima kasih selalu menjadi pendengar dan penasihat yang baik.

Almamater Tercinta Universitas Lampung

SANWACANA

Alhamdulillahirobbil 'alamin, puji syukur kehadirat Allah SWT. atas segala limpahan rahmat, karunia serta izin-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Penerapan *Artificial Neural Network* (ANN) Menggunakan *Backpropagation* dengan Membandingkan Empat Fungsi Aktivasi dalam Memprediksi Harga Emas”. Shalawat serta salam selalu tercurahkan kepada baginda besar Nabi Muhammad SAW. Dalam penyusunan skripsi ini, tentu tak lepas dari bantuan, arahan, bimbingan serta saran dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati penulis ingin menyampaikan banyak terima kasih kepada:

1. Bapak Ir. Warsono, M.S., Ph.D. selaku dosen pembimbing I yang selalu memberikan arahan, bantuan, bimbingan, motivasi dan saran yang mendukung sehingga penulis dapat menyelesaikan skripsi ini.
2. Bapak Dr. Aang Nuryaman, S.Si., M.Si. selaku dosen pembimbing II atas saran dan masukan kepada penulis sehingga dapat menyelesaikan skripsi ini.
3. Ibu Dian Kurniasari, S.Si., M.Sc. selaku dosen penguji yang telah memberikan kritik dan saran yang membangun selama proses penyusunan skripsi.
4. Ibu Dr. Notiragayu, 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. Suropto 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. Kedua orang tua yang selalu memberikan doa, dukungan, kasih sayang dan motivasi kepada penulis.
9. Abang, uda, uni, mba, seluruh anggota keluarga yang telah memfasilitasi, memberikan dukungan dan semangat kepada penulis
10. Teman-teman terbaik yaitu aulia, sherli, dan intan yang telah menemani penulis dalam berbagai keadaan, dan menjadi tempat berbagi semua cerita, canda, tawa dan kenangan selama masa perkuliahan.
11. Teman-teman seperbimbingan yang telah memberikan saran, dukungan, dan semangat kepada penulis selama proses penyelesaian skripsi.
12. Teman-teman Kelas B atas kebersamaan dan canda tawa selama masa perkuliahan.
13. 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 untuk dijadikan perbaikan kedepannya.

Bandar Lampung, 15 Agustus 2022
Penulis,

Ranti Vidia Mahyunis

DAFTAR ISI

	Halaman
DAFTAR TABEL	xvi
DAFTAR GAMBAR	xvii
I. PENDAHULUAN	1
1.1 Latar Belakang dan Masalah	1
1.2 Tujuan Penelitian.....	3
1.3 Manfaat Penelitian.....	4
II. TINJAUAN PUSTAKA	5
2.1 Prediksi	5
2.2 <i>Data Mining</i>	5
2.2.1 Proses <i>Data Mining</i>	6
2.3 <i>Machine Learning</i>	8
2.4 <i>Artificial Neural Network</i>	10
2.4.1 Arsitektur <i>Artificial Neural Network</i>	11
2.4.2 Komponen dan dan Parameter Model <i>Artificial Neural Network</i>	13
2.4.2.1 Lapisan Tersembunyi (<i>Hidden Layer</i>)	13
2.4.2.2 Fungsi Aktivasi	14
2.4.2.3 <i>Learning Rate</i>	17
2.4.2.4 <i>Batch Size</i> dan <i>Optimizer</i>	18
2.4.2.5 <i>Epoch</i> dan <i>Iterasi</i>	18
2.5 Algoritma <i>Backpropagation</i>	19
2.6 <i>Learning Curves</i>	24
2.7 Validasi Model	27
III. METODE PENELITIAN	29
3.1 Tempat dan Waktu Penelitian	29
3.2 Data Penelitian.....	29
3.3 Metode Penelitian.....	30

IV. HASIL DAN PEMBAHASAN	33
4.1 Analisis Deskriptif	33
4.2 Data <i>Preprocessing</i>	34
4.2.1 Pemeriksaan Data Hilang	34
4.2.2 Transformasi Data	35
4.2.3 Pembagian Data	36
4.3 Membangun Model ANN dengan <i>Backpropagation</i>	36
4.3.1 Mendefinisikan Variabel <i>Input</i> dan <i>Output</i>	36
4.3.2 Komponen dan Parameter Jaringan	37
4.3.2.1 <i>Hidden Layer</i>	37
4.3.2.2 Banyak <i>Node</i> pada <i>Hidden Layer</i>	38
4.3.2.3 Menentukan <i>Learning Rate</i>	38
4.3.2.4 Maksimum <i>Epoch</i>	38
4.3.2.5 <i>Batch Size</i> , Iterasi, dan <i>Optimizer</i>	39
4.3.2.6 Fungsi Aktivasi	39
4.3.3 Membangun Model	39
4.3.3.1 Model Satu <i>Hidden Layer</i> dan <i>Learning Rate</i> 0.01	40
4.3.3.2 Model Dua <i>Hidden Layer</i> dan <i>Learning Rate</i> 0.01	41
4.3.3.3 Model Satu <i>Hidden Layer</i> dan <i>Learning Rate</i> 0.001	42
4.3.3.4 Model Dua <i>Hidden Layer</i> dan <i>Learning Rate</i> 0.001	43
4.3.4 Mengecek <i>Overfitting</i> dan <i>Underfitting</i> pada Model	45
4.3.4.1 Model Satu <i>Hidden Layer</i> dengan <i>Learning Rate</i> = 0.01 ...	45
4.3.4.2 Model Dua <i>Hidden Layer</i> dengan <i>Learning Rate</i> = 0.01	46
4.3.4.3 Model Satu <i>Hidden Layer</i> dengan <i>Learning Rate</i> = 0.001 ..	47
4.3.4.4 Model Dua <i>Hidden Layer</i> dengan <i>Learning Rate</i> = 0.001 ...	48
4.4 Prediksi dengan Data <i>Testing</i> Menggunakan Struktur Terbaik Model ANN	49
4.4.1 Hasil Prediksi Menggunakan Model Satu <i>Hidden Layer</i> dengan <i>Learning Rate</i> = 0,01	49
4.4.2 Hasil Prediksi Menggunakan Model Dua <i>Hidden Layer</i> dengan <i>Learning Rate</i> = 0,01	50
4.4.3 Hasil Prediksi Menggunakan Model Satu <i>Hidden Layer</i> dengan <i>Learning Rate</i> = 0,001	50
4.4.4 Hasil Prediksi Menggunakan Model Dua <i>Hidden Layer</i> dengan <i>Learning Rate</i> = 0,001	51
4.5 Visualisasi Hasil Prediksi	52
4.5.1 Visualisasi Hasil Prediksi Menggunakan Model Satu <i>Hidden Layer</i> dengan <i>Learning Rate</i> = 0,01	52
4.5.2 Visualisasi Hasil Prediksi Menggunakan Model Dua <i>Hidden Layer</i> dengan <i>Learning Rate</i> = 0,01	53
4.5.3 Visualisasi Hasil Prediksi Menggunakan Model Satu <i>Hidden Layer</i> dengan <i>Learning Rate</i> = 0,001	54

4.5.4 Visualisasi Hasil Prediksi Menggunakan Model Dua <i>Hidden Layer</i> dengan <i>Learning Rate</i> = 0,001	55
4.6 Akurasi dan Pengujian Model	57
V. KESIMPULAN DAN SARAN	58
5.1 Kesimpulan	58
5.2 Saran	59
DAFTAR PUSTAKA	60

DAFTAR TABEL

Tabel	Halaman
1. Kriteria Nilai MAPE	28
2. Data Harga Emas	29
3. Pemeriksaan Data Hilang pada Data Harga Emas	34
4. Hasil Transformasi <i>Min-Max</i> Data Harga Emas.....	35
5. Pembagian Variabel <i>Input</i> dan <i>Output</i>	37
6. Nilai <i>Loss</i> Model Satu <i>Hidden Layer</i> dengan LR (0.01)	40
7. Nilai <i>Loss</i> Model Dua <i>Hidden Layer</i> dengan LR (0.01).....	41
8. Nilai <i>Loss</i> Model Satu <i>Hidden Layer</i> dengan LR (0.001)	43
9. Nilai <i>Loss</i> Model Dua <i>Hidden Layer</i> dengan LR (0.001).....	44
10. Hasil Prediksi Menggunakan Model Satu <i>Hidden Layer</i> dengan LR = 0.01.....	49
11. Hasil Prediksi Menggunakan Model Dua <i>Hidden Layer</i> dengan LR = 0.01	50
12. Hasil Prediksi Menggunakan Model Satu <i>Hidden Layer</i> dengan LR = 0.001....	50
13. Hasil Prediksi Menggunakan Model Dua <i>Hidden Layer</i> dengan LR = 0.001	51
14. Pengujian Model dengan MAPE, MSE, dan Akurasi.....	57

DAFTAR GAMBAR

Gambar	Halaman
1. Arsitektur <i>Artificial Neural Network</i>	11
2. Jaringan Lapisan Tunggal	12
3. Jaringan <i>Multilayer</i>	13
4. Fungsi Aktivasi <i>Sigmoid</i>	15
5. Fungsi Aktivasi <i>TanH</i>	15
6. Fungsi Aktivasi <i>ReLU</i>	16
7. Fungsi Aktivasi <i>Linear</i>	17
8. Algoritma 1	20
9. Algoritma 2	22
10. Model <i>Underfit</i>	25
11. Model <i>Overfit</i>	26
12. Model <i>Good Fit</i>	26
13. Diagram Alir Proses Prediksi dengan <i>Artificial Neural Network</i>	32
14. Grafik <i>Trend</i> Harga Emas 2016-2022.....	33
15. Grafik Model Satu <i>Hidden Layer</i> dengan <i>Learning Rate</i> = 0,01	46
16. Grafik Model Dua <i>Hidden Layer</i> dengan <i>Learning Rate</i> = 0,01	46

17. Grafik Model Satu <i>Hidden Layer</i> dengan <i>Learning Rate</i> = 0,001.....	47
18. Grafik Model Dua <i>Hidden Layer</i> dengan <i>Learning Rate</i> = 0,001	48
19. Grafik Hasil Prediksi Menggunakan Model Satu <i>Hidden Layer</i> dengan <i>Learning Rate</i> = 0,01	53
20. Grafik Hasil Prediksi Menggunakan Model Dua <i>Hidden Layer</i> dengan <i>Learning Rate</i> = 0,01	54
21. Grafik Hasil Prediksi Menggunakan Model Satu <i>Hidden Layer</i> dengan <i>Learning Rate</i> = 0,001	55
22. Grafik Hasil Prediksi Menggunakan Model Dua <i>Hidden Layer</i> dengan <i>Learning Rate</i> = 0,001	56

I. PENDAHULUAN

1.1 Latar Belakang dan Masalah

Emas adalah salah satu komoditas utama yang diperdagangkan di pasar keuangan global. Emas dianggap sebagai aset yang aman dibandingkan dengan mata uang fiat karena nilai fisik yang dimilikinya dan dari zaman kuno investor berbondong-bondong ke aset emas ketika pasar tidak berkinerja baik. Penggunaan utama emas selain dari nilai investasinya adalah sehubungan dengan penggunaannya yang tinggi di pasar barang mewah dan perhiasan, elektronik dan industri teknologi tinggi di mana emas digunakan sebagai bahan konduktor kelas tinggi untuk memproduksi papan sirkuit cetak dan komponen semi konduktor.

Belakangan ini karena resesi ekonomi global yang disebabkan oleh pandemi *Covid-19*, investor berbondong-bondong membeli aset terkait emas karena mata uang global kehilangan nilainya dengan cepat dan sangat fluktuatif. Dalam skenario ini, memprediksi harga emas sangat penting baik dari perspektif bisnis maupun akademis karena akan memungkinkan para praktisi keuangan untuk memprediksi situasi keuangan di masa depan (Patalay & Bandlamudi, 2021).

Herjanto (2006) mengatakan bahwa prediksi adalah suatu proses memperkirakan sesuatu yang akan terjadi di masa depan berdasarkan informasi yang dimiliki sekarang. Prediksi tidak selalu memberikan jawaban yang pasti melainkan berusaha

untuk memberikan jawaban sedekat mungkin dengan kejadian. Makridakis dkk (1999) berpendapat bahwa prediksi bertujuan untuk mengenali pola sistematis serta menemukan pola hubungan kecenderungan yang didasarkan pada data historis. Pengumpulan data historis disini berperan sebagai tahapan pertama yang dilakukan untuk sebuah prediksi. Salah satu pendekatan dalam memprediksi yang telah banyak digunakan adalah *Artificial Neural Network* (ANN). Fausett (1994) mengatakan bahwa ANN adalah algoritma pemrograman yang diterapkan pada proses komputasi komputer dengan mengadaptasi struktur dan karakteristik jaringan syaraf biologis manusia. Stern (1996) berpendapat bahwa kelebihan dari ANN adalah tidak perlu adanya asumsi-asumsi yang harus dipenuhi dan metode ini memiliki ketelitian yang tinggi serta dapat membantu dalam menyederhanakan berbagai permasalahan yang tidak bisa diselesaikan dengan menggunakan pendekatan matematis. Stern (1996) juga berpendapat bahwa ANN mampu menggambarkan model baik *linear* maupun non *linear* dengan rentang yang cukup luas serta mampu memodelkan hubungan yang kompleks antara *input* dan *output* dalam menemukan pola pada data. Ada berbagai macam algoritma dalam ANN, salah satunya adalah algoritma *backpropagation*.

Algoritma *Backpropagation* adalah sebuah metode sistematis ANN yang menggunakan algoritma pembelajaran terawasi (*Supervised Learning*). Algoritma ini memiliki keunggulan yang sangat baik dalam melakukan pelatihan, terutama pada data berskala besar dan kompleks (Siregar dan Wanto, 2017). Proses dalam algoritma *backpropagation* dibedakan menjadi 3 tahap utama, yaitu *feed forward*, *backpropagation*, dan perubahan nilai bobot. Algoritma *backpropagation* juga merupakan algoritma pembelajaran pada ANN untuk mengurangi tingkat kesalahan dengan mengatur bobot berdasarkan output yang diinginkan (Lesnussa dkk, 2018). Salah satu komponen penting dalam membangun model ANN dengan menggunakan algoritma *backpropagation* adalah fungsi aktivasi. Fungsi aktivasi ini memiliki peranan sangat penting dalam suatu jaringan syaraf tiruan dimana penggunaannya tergantung sesuai kebutuhan dan target yang diinginkan serta fungsi aktivasi ini yang

akan menentukan besarnya bobot. Fungsi aktivasi berperan sebagai sinyal untuk menentukan *output* ke beberapa neuron lainnya (Puspitaningrum, 2006). Beberapa penelitian yang pernah dilakukan sebelumnya menggunakan model ANN dengan algoritma *backpropagation* antara lain: penelitian Lestari dkk (2019) dalam memprediksi jumlah wisatawan di Provinsi Nusa Tenggara Timur (NTB) dengan metode *backpropagation* memiliki nilai selisih *error* yang sangat rendah. Kemudian penelitian Untoro (2020) dalam memprediksi harga saham menggunakan ANN memberikan hasil yang maksimal dalam melakukan prediksi dengan persentase nilai *error* yang relatif rendah. Penelitian ini memiliki kesamaan dengan penelitian Nafi'iah (2016) khususnya dalam memprediksi harga emas menggunakan algoritma *backpropagation*. Penelitian Nafi'ah (2016) memprediksi harga emas dengan metode Regresi Linear, *Backpropagation* dan Fuzzy Mamdani, didapatkan struktur terbaik model yaitu 2-4-1, 2 *node input layer*, 4 *node hidden layer*, dan 1 *output layer* dengan akurasi sebesar 95%. Hal ini menandakan bahwa metode *backpropagation* lebih unggul dibanding dua metode lainnya karena memiliki akurasi tertinggi. Oleh karena itu, penelitian ini akan menerapkan model ANN dengan algoritma *backpropagation* dalam memprediksi harga emas menggunakan struktur yang berbeda yaitu dengan membandingkan 4 fungsi aktivasi serta melakukan penambahan komponen seperti *batch size*, *optimizer*, dan *learning rate* dalam membangun model. Berdasarkan struktur tersebut akan dikaji tingkat efektivitas, akurasi, selisih *error* terkecil, dan hasil prediksi harga emas.

1.2 Tujuan Penelitian

Adapun tujuan penelitian ini yaitu:

1. Memperoleh struktur terbaik model ANN menggunakan algoritma *Backpropagation* berdasarkan empat fungsi aktivasi yang digunakan.

2. Memprediksi harga emas berdasarkan model terbaik dengan empat fungsi aktivasi yang digunakan.
3. Mengkaji performa tingkat efektivitas, akurasi, dan selisih *error* terkecil berdasarkan empat fungsi aktivasi yang digunakan.

1.3 Manfaat Penelitian

Adapun manfaat penelitian ini yaitu:

1. Sebagai rujukan pengembangan ilmu matematika dalam memprediksi harga emas serta dapat menjadi bahan pertimbangan dan informasi tambahan bagi peneliti yang akan melakukan penelitian lebih lanjut terkait prediksi dan model yang telah didapatkan.
2. Sebagai bahan pertimbangan bagi investor, memprediksi harga emas sangat penting baik dari perspektif bisnis karena akan memungkinkan para praktisi keuangan untuk memprediksi situasi keuangan di masa depan.

II. TINJAUAN PUSTAKA

2.1 Prediksi

Kamus besar bahasa Indonesia menyebutkan bahwa prediksi adalah hasil dari kegiatan memprediksi atau memperkirakan nilai pada masa yang akan datang dengan menggunakan data masa lalu. Prediksi merupakan perpaduan antara seni dan ilmu dalam memperkirakan keadaan di masa yang akan datang, dengan memproyeksikan data-data masa lampau ke masa yang akan datang menggunakan model matematika maupun perkiraan yang subjektif (Montgomery, 1990). Herjanto (2006) berpendapat bahwa prediksi tidak selalu memberikan jawaban yang pasti melainkan berusaha untuk memberikan jawaban sedekat mungkin dengan kejadian. Prediksi bertujuan untuk mengenali pola sistematis serta menemukan pola hubungan kecenderungan yang didasarkan pada data historis. Pengumpulan data historis berperan sebagai tahapan pertama yang dilakukan untuk sebuah prediksi (Makridakis dkk, 2006).

2.2 *Data Mining*

Data mining adalah proses pencarian pengetahuan dan pola yang menarik dari sejumlah kumpulan data yang terdapat pada *database*, *data warehouse*, ataupun repositori informasi lainnya (Han dkk, 2012). Turban dkk (2005) berpendapat bahwa kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi

informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai *database* besar. Beberapa teknik pada data mining yang sering digunakan dalam berbagai penelitian antara lain yaitu: *Clustering*, *Classification*, *Association Rules*, *Neural Network* dan lain-lain. Data mining sering juga disebut sebagai *knowledge discovery in database* (KDD) (Santoso, 2007). Ayub (2007) berpendapat bahwa KDD adalah suatu proses terdiri atas pembersihan data (*data cleaning*), integrasi data (*data integration*), pemilihan data (*data selection*), transformasi data (*data transformation*), proses *mining*, evaluasi pola (*pattern evaluation*) dan penyajian pengetahuan (*knowledge presentation*).

2.3.1 Proses Data Mining

Proses data *mining* dapat dilakukan dengan beberapa tahapan yaitu sebagai berikut (Han dkk, 2012):

1. Pembersihan data (*data cleaning*)

Pembersihan data merupakan proses menghilangkan *noise* dan data yang tidak konsisten atau data tidak relevan. Pada umumnya data yang diperoleh, baik dari *database* suatu perusahaan maupun hasil eksperimen, memiliki isian-isian yang tidak sempurna seperti data hilang, data tidak valid atau hanya sekedar salah ketik. Selain itu, ada juga atribut-atribut data yang tidak relevan dengan hipotesis data *mining* yang dimiliki. Data-data yang tidak relevan itu juga lebih baik dibuang. Pembersihan data juga akan mempengaruhi performa teknik data *mining* karena data yang ditangani akan berkurang jumlah dan kompleksitasnya.

2. Integrasi data (*Data Integration*)

Integrasi data merupakan penggabungan data dari berbagai *database* ke dalam satu *database* baru. Tidak jarang data yang diperlukan untuk data *mining* tidak hanya berasal dari satu *database* tetapi juga berasal dari beberapa *database* atau *file* teks. Integrasi data dilakukan pada atribut-atribut yang mengidentifikasi entitas-entitas

yang unik seperti atribut nama, jenis produk, nomor pelanggan dan lainnya. Integrasi data perlu dilakukan secara cermat karena kesalahan pada integrasi data bisa menghasilkan hasil yang menyimpang dan bahkan menyesatkan pengambilan aksi nantinya. Sebagai contoh bila integrasi data berdasarkan jenis produk ternyata menggabungkan produk dari kategori yang berbeda maka akan didapatkan korelasi antar produk yang sebenarnya tidak ada.

3. Seleksi Data (*Data Selection*)

Seleksi data adalah proses seleksi atau pemilihan data yang dianggap relevan untuk dianalisis, karena sering kali data yang ada pada *database* tidak semuanya terpakai. Sebagai contoh, kasus penelitian faktor kecenderungan orang membeli dalam *market basket analyst*, tidak perlu mengambil nama pelanggan, cukup dengan *id* pelanggan saja.

4. Transformasi data (*Data Transformation*)

Data diubah atau digabung ke dalam format yang sesuai untuk diproses dalam data mining. Beberapa metode data mining membutuhkan format data yang khusus sebelum bisa diaplikasikan. Sebagai contoh beberapa metode standar seperti analisis asosiasi dan *clustering* hanya bisa menerima input data kategorik. Transformasi dapat dikatakan sebagai proses penormalan data dengan melakukan penskalaan nilai atribut dari data sehingga dapat berada pada *range* tertentu. Salah satu transformasi yang digunakan adalah transformasi *Min-Max*. Transformasi *Min-Max* adalah metode normalisasi yang melakukan transformasi linear terhadap data asli. Keuntungan dari transformasi *Min-Max* adalah keseimbangan nilai perbandingan antar data sebelum dan sesudah proses normalisasi. Metode transformasi *Min-Max* memberikan hasil tak bias. Pola data *input* diskalakan ke dalam *range* antara 0 sampai 1 menggunakan rumus berikut:

$$X'_i = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (2.1)$$

keterangan:

X'_i = data transformasi ke-i

x_i = data asli/aktual ke-i

x_{min} = nilai minimal dari data x

x_{max} = nilai maksimal dari data x

5. Proses *Mining*

Proses *mining* merupakan suatu proses utama saat metode diterapkan untuk menemukan pengetahuan berharga dan tersembunyi data, seperti menerapkan algoritma atau metode untuk klasifikasi, estimasi, asosiasi, prediksi, klastering dan lain sebagainya.

6. Evaluasi Pola (*Pattern Evaluation*)

Evaluasi pola dilakukan untuk mengidentifikasi pola-pola menarik ke dalam *knowledge-based* yang ditemukan. Pada tahap ini, hasil dari teknik data *mining* berupa pola-pola yang khas maupun model prediksi dievaluasi untuk dinilai apakah hipotesis yang ada memang tercapai atau tidak. Apabila ternyata hasil yang diperoleh tidak sesuai hipotesis yang ada, maka dapat dicoba metode data *mining* lain yang lebih sesuai, atau menerima sebagai suatu hasil yang di luar dugaan yang mungkin bermanfaat.

7. Presentasi Pengetahuan (*Knowledge Presentation*)

Presentasi pengetahuan merupakan visualisasi dan penyajian pengetahuan mengenai metode yang digunakan untuk memperoleh pengetahuan yang diperoleh pengguna. Tahap terakhir dari proses data *mining* adalah bagaimana memformulasikan keputusan atau aksi dari hasil analisis yang didapat. Presentasi hasil data mining dalam bentuk pengetahuan yang bisa dipahami semua orang adalah satu tahapan yang diperlukan dalam proses data *mining*. Salah satu presentasi hasil data *mining* yang dapat mengkomunikasikan hasil tersebut adalah visualisasi.

2.3 *Machine Learning*

Machine Learning atau pembelajaran mesin merupakan pendekatan dalam *Artificial Intelligence* (AI) yang banyak digunakan untuk menggantikan atau menirukan

perilaku manusia dalam menyelesaikan masalah atau melakukan otomatisasi (Tanaka dan Okutomi, 2014). Sesuai namanya, *machine learning* mencoba menirukan bagaimana proses manusia atau makhluk cerdas belajar dan menggeneralisasi. Ciri dari *machine learning* adalah adanya proses pelatihan dan pembelajaran. Pembelajaran mesin atau *machine learning* memungkinkan komputer mempelajari sejumlah data (*learn from data*) sehingga dapat menghasilkan suatu model untuk melakukan proses *input-output* tanpa menggunakan kode program yang dibuat secara eksplisit.

Oleh karena itu, *machine learning* membutuhkan data untuk dipelajari yang disebut sebagai data *training*. Terdapat dua macam pelatihan dalam metode ini, yaitu pelatihan terawasi (*supervised learning*) dan pelatihan tidak terawasi (*unsupervised learning*). Perbedaan antara kedua jenis pelatihan tersebut tergantung pada bagaimana algoritma pelatihan menggunakan jenis pola. Tipe pembelajaran terbagi menjadi dua (Fausset, 1994), yaitu:

a. *Supervised Learning* (Pelatihan Terawasi)

Metode ini digunakan ketika nilai keluaran yang diinginkan sudah diketahui sebelumnya. Nilai keluaran jaringan ini akan dibandingkan dengan nilai keluaran target yang dinamakan dengan galat. Ketika galat masih cukup besar artinya masih perlu dilakukan pembelajaran lagi hingga didapatkan nilai galat yang kecil atau nilai keluaran dari jaringan yang tidak berbeda jauh dengan nilai keluaran target. Adapun contoh algoritma yang menggunakan metode pembelajaran terawasi (*supervised learning*) adalah *hebbian (hebb rule)*, *perceptron*, *adaline*, *boltzman*, *hapfield*, dan *backpropagation*.

b. *Unsupervised Learning* (Pelatihan Tidak Terawasi)

Metode pembelajaran tak terawasi (*unsupervised learning*) merupakan metode pembelajaran ketika tidak memerlukan acuan nilai keluaran yang diharapkan. Pada metode ini tidak diketahui seperti apa nilai keluaran yang diharapkan, melainkan hanya ditentukan nilai bobot dalam suatu *range* sesuai nilai *input* yang diberikan.

Adapun penerapan dari metode ini adalah klasifikasi. Karena tujuan utama dari metode ini adalah mengelompokkan unit-unit yang memiliki kemiripan di suatu area

tertentu. Contoh algoritma yang menggunakan metode pembelajaran tak terawasi adalah *competitive*, *hebbian*, *kohonen*, LVQ (*Learning Vector Quantization*), dan *neocognitron*.

2.4 Artificial Neural Network

Artificial neural network merupakan jaringan saraf tiruan berbasis komputasi yang pertama kali dikemukakan oleh Warren MacCulloch bersama dengan Walter Pitts pada tahun 1943. MacCulloch merupakan seorang ahli saraf dan Pitts merupakan seorang ahli logika. Keduanya bekerjasama merancang model perhitungan sederhana yang menyatukan keilmuan biologi terkait jaringan saraf pada manusia dengan ilmu logika dalam menyelesaikan suatu permasalahan (Suzuki, 2013).

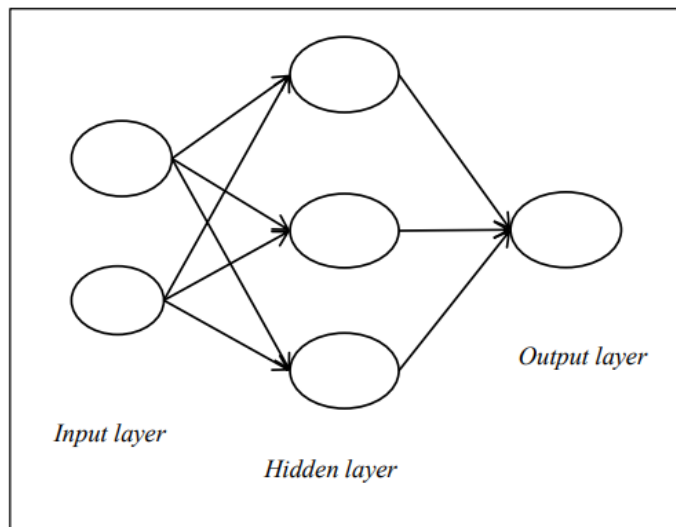
Artificial Neural Network biasa disebut jaringan syaraf tiruan adalah sistem komputasi arsitektur yang operasinya merupakan sistem pemrosesan informasi dalam menstimulasikan sistem kecerdasan buatan yang memiliki karakteristik sama dengan cara kerja sistem saraf biologis manusia (Fausett, 1994).

Artificial Neural Network tercipta sebagai suatu generalisasi model matematis dari pemahaman manusia atau suatu syarat biologis manusia yang dibentuk atas dasar asumsi sebagai berikut (Wati, 2011):

1. Elemen sederhana yang digunakan dalam memproses informasi disebut neuron.
2. Terdapat penghubung antar neuron yang digunakan sebagai pengantar sinyal.
3. Setiap sambungan penghubung memiliki bobot yang bersesuaian yang akan diproses dengan sinyal yang terhubung.
4. Setiap neuron memiliki fungsi aktivasi dengan cara menjumlahkan bobot-bobot yang masuk untuk menentukan sinyal keluarannya.

2.4.1 Arsitektur *Artificial Neural Network*

Secara umum terdapat tiga lapisan pada arsitektur ANN, yaitu *input layer*, *hidden layer*, dan *output layer*. Setiap lapisan tersebut dapat memiliki jumlah *node* dan neuron yang berbeda-beda (Desiana dan Arhami, 2006). Arsitektur ANN digambarkan pada Gambar 1 berikut.



Gambar 1. Arsitektur *Artificial Neural Network*

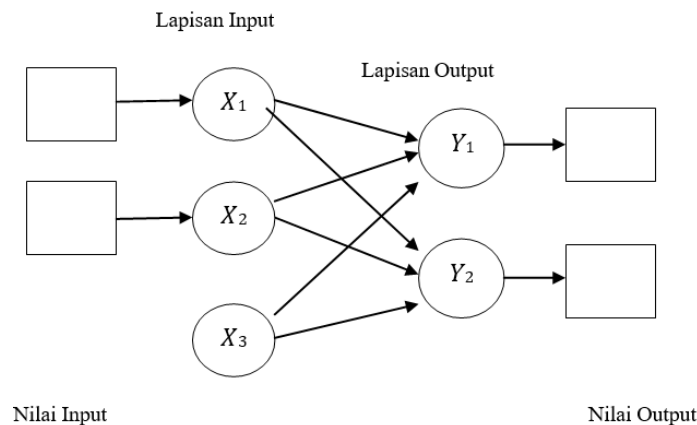
Berdasarkan Gambar 1, lapisan arsitektur ANN terdiri dari:

- a. *Input layer* yaitu lapisan dari beberapa neuron yang akan menerima sinyal dari luar kemudian diteruskan ke neuron lain yang ada pada jaringan, sesuai dengan cara kerja sel saraf sensorik.
- b. *Hidden layer* yaitu lapisan tiruan dari sel-sel saraf konektor pada jaringan saraf biologis yang berfungsi untuk meningkatkan kemampuan jaringan.
- c. *Output layer* yaitu lapisan yang berfungsi menyalurkan sinyal-sinyal keluaran hasil proses jaringan yang merupakan tiruan dari sel-sel saraf motorik pada jaringan saraf biologis.

Fausset (1994) mengatakan bahwa ANN memiliki beberapa arsitektur yang sering digunakan, yaitu:

1. Jaringan lapisan tunggal (*single layer network*)

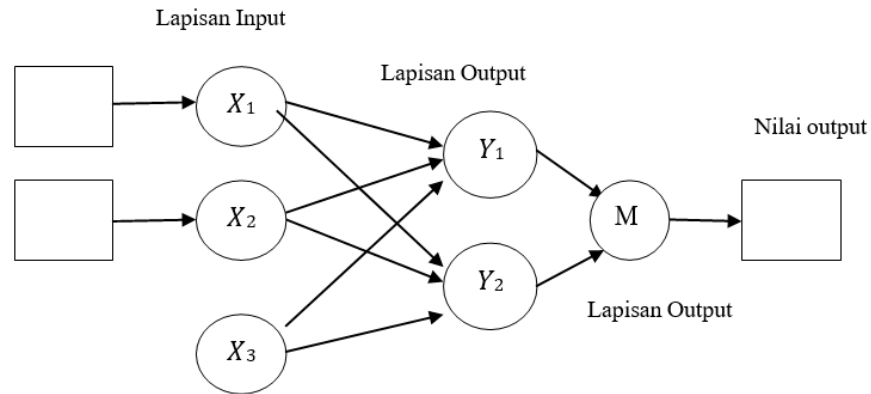
Jaringan lapisan tunggal terdiri dari satu *layer input* dan satu *layer output*. Setiap *neuron/unit* yang terdapat di dalam lapisan/*layer input* selalu terhubung dengan setiap *neuron* yang terdapat pada *layer output*. Jaringan ini hanya menerima input kemudian secara langsung akan mengolahnya menjadi *output* tanpa harus melalui lapisan tersembunyi. Contoh algoritma jaringan syaraf tiruan yang menggunakan metode ini yaitu: *ADALINE*, *Hopfield*, *Perceptron*.



Gambar 2. Jaringan Lapisan Tunggal

2. Jaringan *multilayer* (*multilayer network*).

Jaringan *multilayer* memiliki memiliki 3 jenis layer, yaitu: *layer input*, *layer output*, dan *layer hidden*. Jaringan *multilayer* ini dapat menyelesaikan permasalahan yang kompleks dibandingkan jaringan dengan lapisan tunggal. Namun proses pelatihan sering membutuhkan waktu yang cenderung lama. Contoh algoritma ANN yang menggunakan metode ini yaitu: *MADALINE*, *backpropagation*, *neocognitron*.



Gambar 3. Jaringan *Multilayer*

2.4.2 Komponen dan Parameter Model *Artificial Neural Network*

Proses membangun *model* ANN dengan algoritma *backpropagation* menggunakan komponen dan parameter sebagai berikut.

2.4.2.1 Lapisan Tersembunyi (*Hidden Layer*)

Algoritma *backpropagation* merupakan bagian dari *multilayer perceptron*, sehingga dalam struktur jaringannya terdapat *hidden layer* yang harus diperhatikan. Jumlah *hidden layer* juga memberikan dampak yang cukup signifikan bagi nilai galat yang dihasilkan dari proses *training* dan *testing*. Jumlah *hidden layer* 1 saja terkadang cukup memadai untuk menghasilkan output yang sesuai target. Penggunaan 1 *hidden layer* menyebabkan akurasi yang berbeda dengan penggunaan 2 *hidden layer*. Karena penggunaan 2 *hidden layer* akan menambah iterasi saat proses *looping* atau proses *backward*. Hal ini tidak menutup kemungkinan terjadinya peningkatan akurasi dan membuat pelatihan menjadi lebih mudah dan cepat apabila dilakukan penambahan *hidden layer* atau sebaliknya (Fausett, 1994).

Ada beberapa aturan yang dapat digunakan dalam menentukan jumlah *node* pada hidden layer (Heaton, 2008).

1. Jumlah *node* harus berada di antara jumlah lapisan *input* dan jumlah lapisan *output*.
2. Jumlah *node* dapat ditentukan menggunakan rumus:

$$[(\text{jumlah } input \times \frac{2}{3}) + \text{jumlah } output].$$
3. Jumlah *node* pada *hidden layer* harus kurang dari dua kali lipat jumlah *input layer*.

2.4.2.2 Fungsi Aktivasi

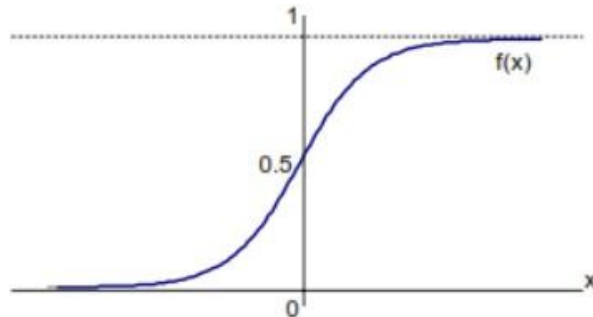
Fungsi aktivasi dalam algoritma *backpropagation* berperan sebagai sinyal untuk menentukan keluaran ke beberapa neuron lainnya. Fungsi aktivasi memiliki peranan sangat penting dalam algoritma *backpropagation*, karena penggunaannya tergantung sesuai kebutuhan dan target yang diinginkan. Fungsi aktivasi ini akan menentukan besarnya bobot. Fungsi aktivasi adalah fungsi yang menggambarkan hubungan antara tingkat aktivasi internal (*summation function*) yang mungkin berbentuk *linear* atau *non-linear* (Puspitaningrum, 2006).

Fungsi aktivasi akan menentukan apakah sinyal dari *input* neuron akan diteruskan atau tidak, sehingga fungsi aktivasi berfungsi untuk menentukan apakah neuron tersebut harus “aktif” atau tidak berdasarkan dari *weighted sum* dari *input*. Inputan ini akan diproses melalui suatu fungsi perambatan. Fungsi aktivasi pada algoritma *backpropagation* harus mempunyai beberapa karakteristik penting, yaitu kontinu, dapat dibedakan, dan tidak meningkat secara monoton.

Keefisienan komputasi, diperoleh dari turunan fungsi tersebut agar mudah dikomputasikan (Siang, 2005). Ada beberapa fungsi aktivasi yang sering digunakan

dalam algoritma *backpropagation* (Sena, 2017), yaitu:

1. Fungsi Aktivasi *Sigmoid*



Gambar 4. Fungsi Aktivasi *Sigmoid*

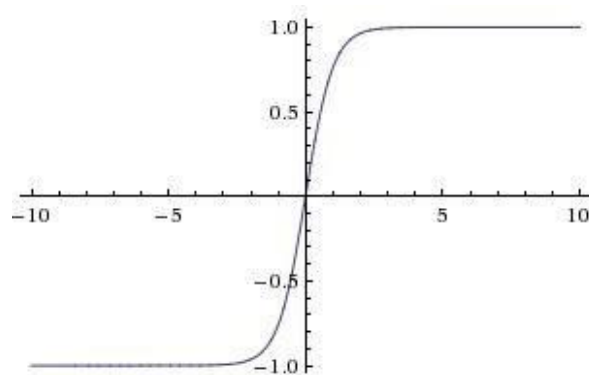
Fungsi *Sigmoid* merupakan fungsi *non-linear* yang memiliki *range* antara 0 sampai 1, sehingga fungsi ini sering digunakan untuk algoritma *backpropagation* yang membutuhkan nilai *input* berupa bilangan real dan *output* yang terletak pada interval 0 sampai 1. Fungsi *Sigmoid* memiliki kekurangan, yaitu dapat mematikan *gradient* dan *output* dari fungsi ini tidak *zero centered*. Berikut ini adalah fungsi *Sigmoid* (Siang, 2005):

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

dengan turunan

$$F'(x) = (F(x)(1 - F(x))) \quad (2.3)$$

2. Fungsi Aktivasi *TanH*

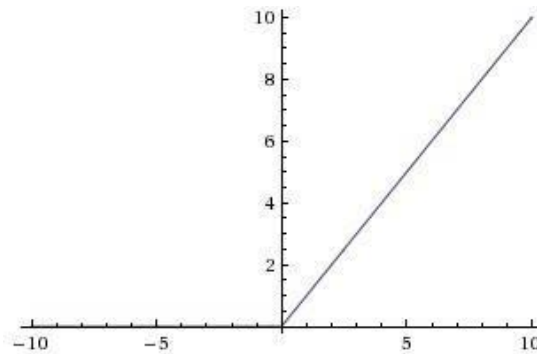


Gambar 5. Fungsi Aktivasi *TanH*

Fungsi aktivasi *Tanh* merupakan fungsi non-linear yang akan mengubah nilai *input*-nya menjadi sebuah nilai yang memiliki *range* antara dari -1 sampai 1 seperti yang ditampilkan pada Gambar 5. Fungsi aktivasi *Tanh* memiliki kekurangan yaitu bisa mematikan *gradient*, tetapi kelebihan dari fungsi ini adalah *output* yang dimiliki merupakan *zero-centered*. Fungsi *Tanh* merupakan pengembangan dari fungsi *Sigmoid* (Kusumadewi, 2003).

$$F(x) = \tanh(x) = \frac{2}{1+e^{-x}} - 1 \quad (2.4)$$

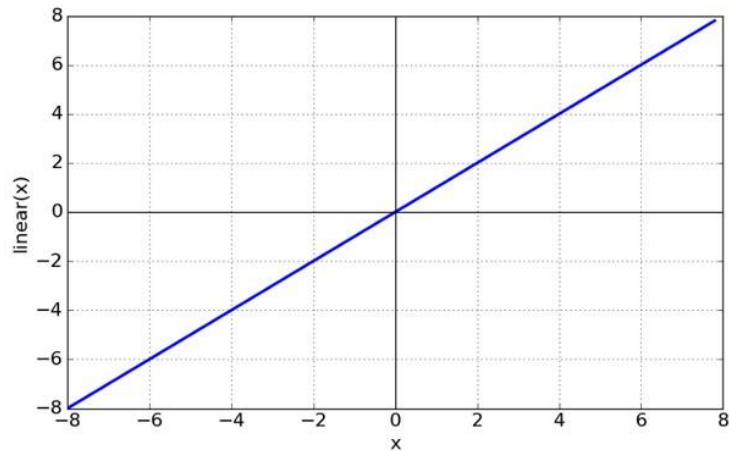
3. Fungsi Aktivasi *Rectified Linear Unit* (ReLU)



Gambar 6. Fungsi Aktivasi *ReLU*

Fungsi Aktivasi *ReLU* adalah lapisan aktivasi pada model yang mengaplikasikan fungsi $f(x) = \max(0, x)$, dapat dikatakan bahwa ReLU pada intinya hanya membuat pembatas pada bilangan nol, artinya apabila $x \leq 0$ maka $x = 0$ dan apabila $x > 0$ maka $x = x$. Kelebihan dari fungsi ini sangat mempercepat proses konvergensi yang dilakukan *stochastic gradient descent* (SGD) apabila dibandingkan dengan fungsi *Sigmoid* dan *Tanh* (Miranda dan Castano, 2017).

4. Fungsi Aktivasi *Linear*



Gambar 7. Fungsi Aktivasi *Linear*

Fungsi Aktivasi *Linear* merupakan fungsi yang memiliki nilai *output* yang sama dengan nilai *input*, dan dirumuskan sebagai berikut:

$$\text{Linear Function; } f(x) = x \quad (2.5)$$

Hal ini berkaitan dengan, jika sebuah neuron menggunakan fungsi aktivasi *linear*, maka keluaran dari neuron tersebut adalah *weighted sum* dari *input* dijumlahkan dengan bias (Fausett, 1994).

2.4.2.3 *Learning Rate*

Learning rate merupakan salah satu parameter *training* untuk menghitung nilai koreksi bobot pada waktu proses *training*. Nilai *learning rate* ini berada pada *range* nol 0 sampai 1. Semakin besar nilai *learning rate*, maka proses *training* akan berjalan semakin cepat. Semakin besar *learning rate*, maka ketelitian jaringan akan semakin berkurang, dan berlaku sebaliknya. Apabila *learning rate*-nya semakin kecil, maka

ketelitian jaringan akan semakin besar atau bertambah dengan konsekuensi proses *training* akan memakan waktu yang semakin lama (Yifei Lu, 2017).

2.4.2.4 Batch Size dan Optimizer

Batch size atau ukuran *batch* merupakan istilah yang digunakan dalam pembelajaran mesin dan mengacu pada jumlah contoh pelatihan yang digunakan dalam satu iterasi. Ukuran *batch* adalah *hyperparameter* yang menentukan jumlah sampel untuk dikerjakan sebelum memperbarui parameter model internal. Pada akhir *batch*, prediksi dibandingkan dengan variabel *output* yang diharapkan dan galat dihitung. Berdasarkan galat tersebut, algoritma pembaruan digunakan untuk meningkatkan model.

Dataset *training* dapat dibagi menjadi satu atau beberapa *batch*. Umumnya ukuran *batch* yang digunakan sebesar 32, 64, dan 128. *Optimizer* atau optimisasi adalah fungsi pembantu dalam menentukan bobot yang lebih tepat sehingga dapat mengurangi jumlah *trial and error* secara drastis saat melakukan *backpropagation*. Adam adalah salah satu *optimizer* yang merupakan metode *learning rate* adaptif, dimana Adam menghitung *learning rate* individu untuk parameter yang berbeda. Nama “Adam” berasal dari “*adaptive moment estimation*” karena Adam menggunakan estimasi gradien momen pertama dan kedua untuk mengadaptasi *learning rate* untuk setiap bobot *neural network* (Kingma dan Ba, 2015).

2.4.2.5 Epoch dan Iterasi

Jumlah *epoch* adalah *hyperparameter* yang menentukan berapa kali algoritma pembelajaran akan bekerja mengolah seluruh *dataset training*. Satu *epoch* berarti

bahwa setiap sampel dalam dataset training memiliki kesempatan untuk memperbarui parameter model internal. Satu *epoch* terdiri dari satu atau lebih *batch*. Satu *epoch* berarti bahwa seluruh dataset sudah melalui proses *training* pada *neural network* sampai dikembalikan ke awal untuk sekali putaran. Karena satu *epoch* terlalu besar untuk dimasukkan (*feeding*) kedalam komputer, maka perlu membaginya kedalam satuan kecil (*batches*). Sedangkan Iterasi adalah jumlah batch yang diperlukan untuk menyelesaikan satu *epoch*. Satu Iterasi telah dilakukan jika satu kali proses *forward* dan *backward pass* telah dilakukan.

2.5 Algoritma *Backpropagation*

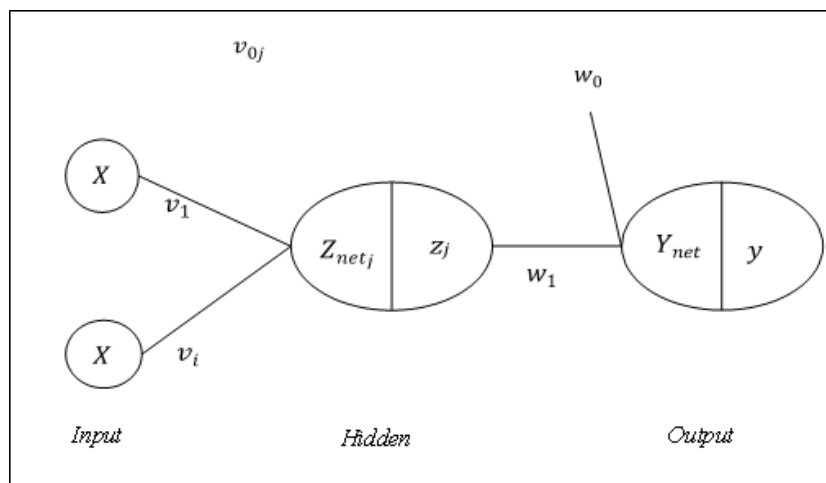
Backpropagation adalah sebuah metode sistematis jaringan saraf tiruan yang menggunakan algoritma pembelajaran terawasi (*supervised learning*) dan biasanya digunakan oleh *perceptron* dengan banyak layer lapisan untuk mengubah bobot-bobot yang ada pada lapisan tersembunyi. Metode *backpropagation* pada awalnya dirancang untuk *neural network feedforward*, tetapi pada perkembangannya, metode ini diadaptasi untuk pembelajaran pada model *neural network* lainnya (Astuti, 2009). Ciri dari metode ini adalah meminimalkan galat pada *output* yang dihasilkan oleh jaringan.

Algoritma *backpropagation* mempunyai pengaturan hubungan yang sangat sederhana yaitu: jika keluaran memberikan hasil yang salah, maka penimbang (*weight*) dikoreksi supaya galat dapat diperkecil dan respon jaringan selanjutnya diharapkan akan mendekati nilai yang benar. Algoritma ini juga berkemampuan untuk memperbaiki penimbang pada lapisan tersembunyi (*hidden layer*) (Purnomo dan Kurniawan, 2006).

Fausett (1994) menyatakan bahwa algoritma *backpropagation* dalam melakukan *training* terhadap suatu jaringan terdapat 3 tahapan yaitu fase *feedforward* (tahap

maju), fase *backpropagation* dan fase perubahan atau penyesuaian bobot dengan bobot yang diatur berdasarkan bilangan acak. Ketiga fase tersebut dilakukan berulang hingga kondisi penghentian terpenuhi. Kondisi tersebut akan berhenti ketika jumlah iterasi melebihi jumlah maksimum iterasi yang telah ditetapkan. Secara rinci langkah dalam algoritma *backpropagation* diuraikan sebagai berikut:

Algoritma 1: fase *feedforward* yaitu fase dimana setiap *input* dihitung maju menuju *hidden layer* hingga *output layer* dengan menggunakan fungsi aktivasi yang ditentukan.



Gambar 8. Algoritma 1

1. Setiap unit *input* ($x_i, i = 1, 2, \dots, p$) menerima sinyal *input* x_i dan diteruskan ke *hidden layer*.
2. Jumlahkan tiap *hidden layer* ($Z_j, j = 1, 2, \dots, n$) dengan bobot sinyal *input*.

$$Z_{netj} = v_{0j} + \sum_{i=1}^p x_i v_{ji} \quad (2.6)$$

keterangan:

$Z_{net,j}$ = nilai *output* untuk unit Z_j

x_i = nilai *input* pada unit i

v_{ij} = bobot antara *input* unit i dan *layer* j

v_{0j} = bobot bias *input* unit i dan *layer* j

Selanjutnya fungsi aktivasi diaplikasikan untuk menghitung sinyal *output* dan kemudian mengirim sinyal ke semua unit di lapisan *output*:

$$z_j = f(Znet_j) = \frac{1}{1 + \exp^{-znet_j}} \quad (2.7)$$

keterangan:

z_j = nilai aktivasi dari unit Z_j

3. Setiap unit *output* ($Y_l, l = 1, 2, \dots, k$) menjumlahkan bobot sinyal *input*.

$$Y_{netl} = w_{0k} + \sum_{j=1}^n Z_j v_{kj} \quad (2.8)$$

keterangan:

Y_{netl} = *input* untuk unit Y_l

w_{0k} = nilai bobot sambungan pada bias untuk unit Y_l

w_{kj} = nilai bobot sambungan dari Z_{ij} ke unit Y_l

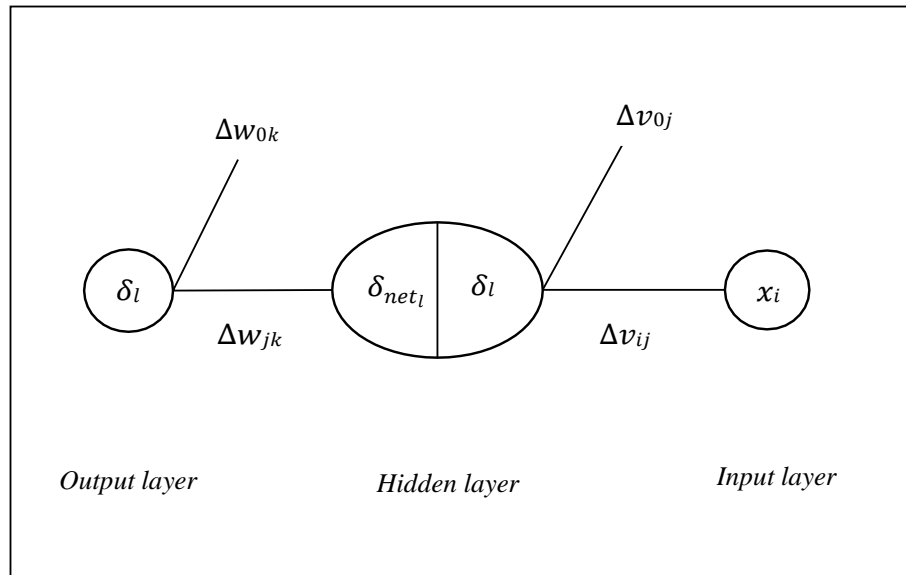
Kemudian mengaplikasikan fungsi aktivasinya untuk menghitung sinyal *output*.

$$y_l = f(Ynet_j) = \frac{1}{1 + \exp^{-y_{netl}}} \quad (2.9)$$

keterangan:

y_l = *input* untuk unit Y_l

Algoritma II: fase *backpropagation* yaitu fase dimana kesalahan (selisih antara keluaran dengan target yang diinginkan) yang terjadi dihitung mundur dari garis yang berhubungan langsung dengan unit-unit di *output layer*.



Gambar 9. Algoritma 2

1. Setiap unit *output* (Y_l , $l = 1, 2, \dots, k$) menerima pola target sesuai dengan pola *training input*, menghitung informasi galat:

$$\delta_l = (t_l - y_l) f'(y_{net_l}) \quad (2.10)$$

keterangan:

δ_l = nilai aktivasi kesalahan pada *output layer*

t_l = nilai target data

Kemudian menghitung bobot barunya (digunakan untuk memperbaharui w_{jk}).

$$\Delta w_{jk} = \alpha \delta_l z_j \quad (2.11)$$

keterangan:

Δw_{jk} = selisih antara $w_{kj}(t)$ dengan $w_{kj}(t+1)$

α = nilai *learning rate*

z_j = nilai aktivasi dari unit Z_j

Selanjutnya menghitung koreksi bias yang akan digunakan untuk memperbaharui bobot bias w_{0k}

$$\Delta w_{0k} = \alpha \delta_l \quad (2.12)$$

keterangan:

Δw_{0k} = koreksi nilai bias

Setelah koreksi bias dihitung, langkah berikutnya adalah mengirim faktor δ_l ke unit *layer* sebelumnya.

2. Setiap unit *hidden* ($Z_j, j = 1, 2, \dots, n$) menerima *input* delta dari langkah sebelumnya. Kemudian menghitung faktor δ di setiap unit *hidden* berdasarkan kesalahan tiap unit *hidden*.

$$\delta_{netl} = \sum_{l=1}^k \delta_l w_{kj} \quad (2.13)$$

keterangan:

δ_{netl} = nilai untuk menghitung kesalahan *hidden layer*

w_{kj} = bobot dari Z_{ij} ke Y_l

Kemudian menghitung faktor δ di unit *hidden* dengan fungsi aktivasi yang telah diturunkan.

$$\delta_l = \delta_{netl} f'(z_{netj}) = \delta_{netl} z_j (1 - z_j) \quad (2.14)$$

keterangan:

δ_l = nilai aktivasi kesalahan pada *hidden layer*

Selanjutnya menghitung koreksi bobot yang akan digunakan untuk memperbaharui v_{ij} .

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (2.15)$$

keterangan:

Δv_{ij} = selisih antara $v_{ij}(t)$ dengan $v_{ij}(t + 1)$

x_i = unit ke-i pada *input layer*

Kemudian menghitung koreksi bias yang akan digunakan untuk memperbaharui v_{0j} .

$$\Delta v_{0j} = \alpha \delta_j \quad (2.16)$$

keterangan:

Δv_{0j} = koreksi nilai bias

Algoritma III: Perubahan Bobot dan bias yaitu fase dilakukannya modifikasi bobot untuk menurunkan kesalahan yang terjadi.

1. Setiap unit *output* Y_l memperbaharui bias dan bobot ($j = 0, \dots, n$)

$$w_{jl}(\text{baru}) = w_{jl}(\text{lama}) + \Delta w_{jl} \quad (2.17)$$

Setiap *hidden* unit Z_j memperbaharui bobot dan bias ($i = 0, \dots, p$)

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (2.18)$$

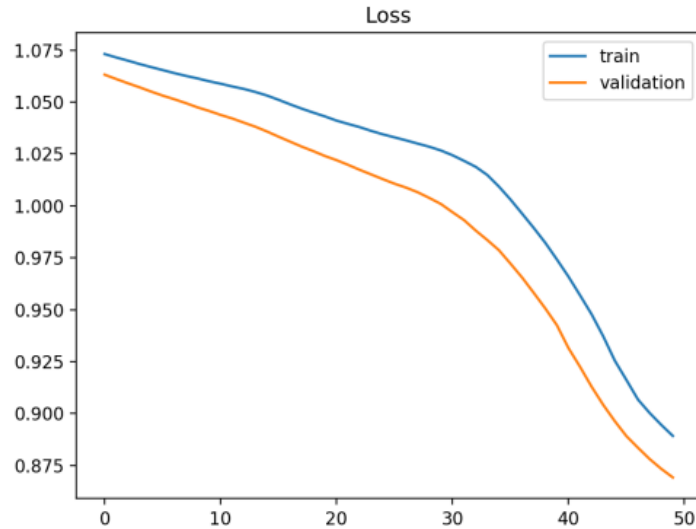
2. Proses pelatihan akan berhenti ketika kondisi telah terpenuhi

2.6 *Learning Curves*

Learning Curves atau kurva pembelajaran merupakan plot kinerja pembelajaran model. Kurva pembelajaran adalah alat diagnostik yang banyak digunakan dalam *machine learning* untuk algoritma yang belajar dari data *training* secara bertahap. Model dievaluasi dengan data *training* dan data *validation* setelah setiap pembaruan selama proses *training*, dan plot kinerja model yang diukur dapat menunjukkan kurva pembelajaran. Kurva pembelajaran dapat digunakan untuk mendiagnosis model yang *underfit*, *overfit*, atau *good fit* serta apakah data *training* dan data *validation* cukup representatif (Brownlee, 2018).

a. Kurva Pembelajaran *Underfit*

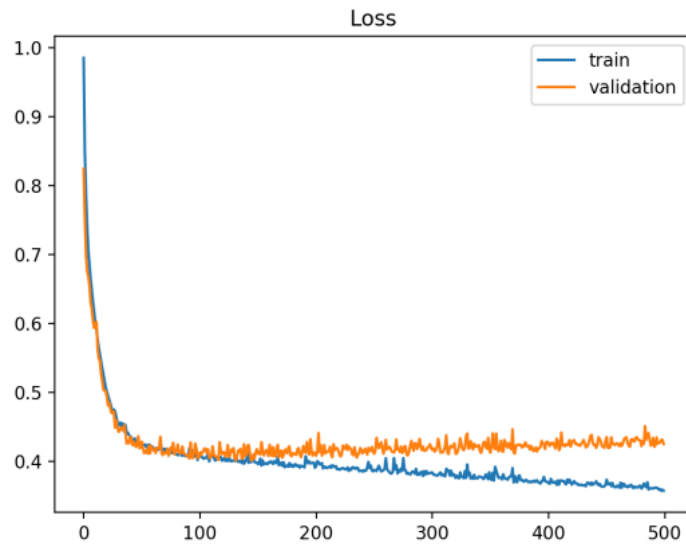
Underfitting mengacu pada model yang tidak dapat mempelajari data *training*. *Underfitting* terjadi ketika model tidak dapat memperoleh nilai galat yang cukup rendah pada proses *training*. Model *underfitting* dapat diidentifikasi dari kurva *training loss* saja yang menunjukkan garis datar atau nilai *noise* dari *loss* yang relatif tinggi, sehingga dapat dikatakan bahwa model tidak dapat mempelajari data *training*.



Gambar 10. Model *Underfit*

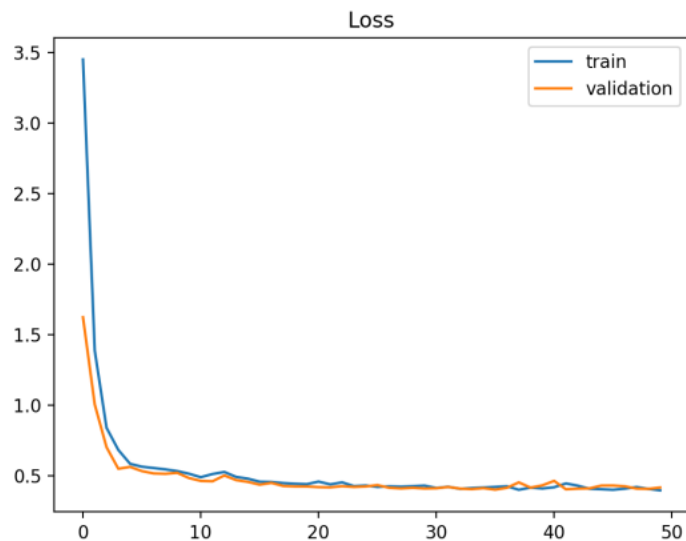
b. Kurva Pembelajaran *Overfit*

Overfitting mengacu pada model yang telah mempelajari data *training* terlalu baik, termasuk gangguan statistik atau fluktuasi acak dalam data *training*. *Overfitting* mengakibatkan peningkatan kesalahan generalisasi. Peningkatan kesalahan generalisasi ini dapat diukur dengan kinerja model pada data *validation*.

Gambar 11. Model *Overfit*

c. Kurva Pembelajaran *Good Fit*

Good Fit adalah tujuan dari algoritma pembelajaran dan ada di antara model overfit dan underfit. *Good Fit* diidentifikasi oleh *training* dan *validation loss* yang menurun ke titik stabilitas dengan kesenjangan minimal antara dua nilai *loss* terakhir. Plot kurva pembelajaran menunjukkan *good fit* jika plot *training loss* menurun ke titik stabilitas dan plot *loss validation* menurun ke titik stabilitas juga dan memiliki celah yang kecil satu sama lain.

Gambar 12. Model *Good Fit*

2.7 Validasi Model

Validasi model adalah suatu hal yang penting untuk prediksi, yaitu bagaimana mengukur kesesuaian antara data yang sudah ada dengan data yang telah diprediksi. Menghitung nilai galat sangat penting untuk melihat hasil pelatihan pada ANN. Hal ini dikarenakan pada tahap pelatihan nilai galat yang diharapkan adalah nilai yang paling kecil. Ada beberapa cara yang dapat digunakan untuk mengevaluasi model (fungsi) yang sudah dihasilkan (Pakaja, 2012):

1. Nilai *Mean Squared Error* (MSE)

Mean Squared Error adalah rata-rata kesalahan kuadrat antara nilai aktual dan nilai prediksi. *Mean Squared Error* secara umum digunakan untuk mengevaluasi rata-rata kesalahan pada pencarian nilai prediksi. Nilai MSE yang rendah atau nilai MSE mendekati nol menunjukkan bahwa hasil prediksi sesuai dengan data aktual dan bisa dijadikan perhitungan prediksi untuk periode mendatang. Rumus perhitungan MSE adalah sebagai berikut:

$$MSE = \frac{\sum_{i=1}^n (y_i - t_i)^2}{n} \quad (2.19)$$

keterangan:

y_i = nilai aktual

t_i = nilai prediksi

n = banyaknya data

Karena MSE diukur dalam satuan kuadrat, maka *root mean square error* (RMSE) dapat ditafsirkan sebagai jarak rata-rata, antara nilai yang diprediksi dan diamati, diukur dalam satuan variabel target dengan rumus sebagai berikut:

$$RMSE = \sqrt{MSE} \quad (2.20)$$

2. Nilai *Mean Square Percentage Error* (MAPE)

Mean Square Percentage Error adalah persentase kesalahan rata-rata secara mutlak. Perhitungan MAPE bertujuan untuk mengukur statistik tentang akurasi prediksi menggunakan kesalahan absolut pada tiap periode dibagi dengan nilai aktual kemudian merata-rata kesalahan persentase tersebut. *Mean square percentage error* memberikan informasi seberapa besar kesalahan prediksi dibandingkan dengan nilai sebenarnya. Terdapat beberapa kriteria tentang nilai MAPE (Lewis, 1982) yaitu:

Tabel 1. Kriteria Nilai MAPE

Range MAPE	Arti Nilai
< 10%	Kemampuan model prediksi sangat baik
10%-20%	Kemampuan model prediksi baik
20%-50%	Kemampuan model prediksi layak
> 50%	Kemampuan model prediksi buruk

Rumus perhitungan MAPE adalah sebagai berikut:

$$MAPE = \frac{\sum_{t=1}^n \frac{|y_i - t_i|}{y_i}}{n} \times 100\% \quad (2.21)$$

keterangan:

y_i = nilai aktual

t_i = nilai prediksi

n = banyaknya data

Terdapat simbol absolut pada rumus MAPE menunjukkan bahwa nilai negatif perhitungannya akan tetap secara nyata bernilai positif.

III. METODE PENELITIAN

3.1 Tempat dan Waktu Penelitian

Penelitian ini dilakukan pada semester genap tahun akademik 2021/2022, bertempat di Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung.

3.2 Data Penelitian

Data yang digunakan pada penelitian ini merupakan data sekunder berupa data historis indeks harga emas selama 6 tahun yang diunduh langsung dari *website Yahoo Finance*: <https://finance.yahoo.com/quote/GC%3DF/history?p=GC%3DF>. Data tersebut merupakan data harga emas pada periode 25 Januari 2016 sampai 23 Maret 2022 dengan jumlah data sebanyak 1554. Struktur data penelitian yang digunakan dapat dilihat pada tabel berikut:

Tabel 2. Data Harga Emas

<i>Date</i>	<i>Open</i>	<i>High</i>	<i>Low</i>	<i>Close</i>	<i>Volume</i>
25/01/2016	1099,3	1109,3	1099,3	1106,2	8
26/01/2016	1115,1	1121,7	1114,3	1121,7	181
27/01/2016	1117,8	1117,8	1115,8	1116,1	168706
28/01/2016	1125,4	1125,7	1109,8	1115,6	52861
29/01/2016	1114,4	1118,6	1108,8	1116,4	6955
....

keterangan:

<i>Open</i>	: harga pembukaan (US\$ per <i>troy once</i>)
<i>High</i>	: harga tertinggi (US\$ per <i>troy once</i>)
<i>Low</i>	: harga terendah (US\$ per <i>troy once</i>)
<i>Close</i>	: harga penutupan (US\$ per <i>troy once</i>)
<i>Volume</i>	: jumlah transaksi emas dalam sehari (US\$)

3.3 Metode Penelitian

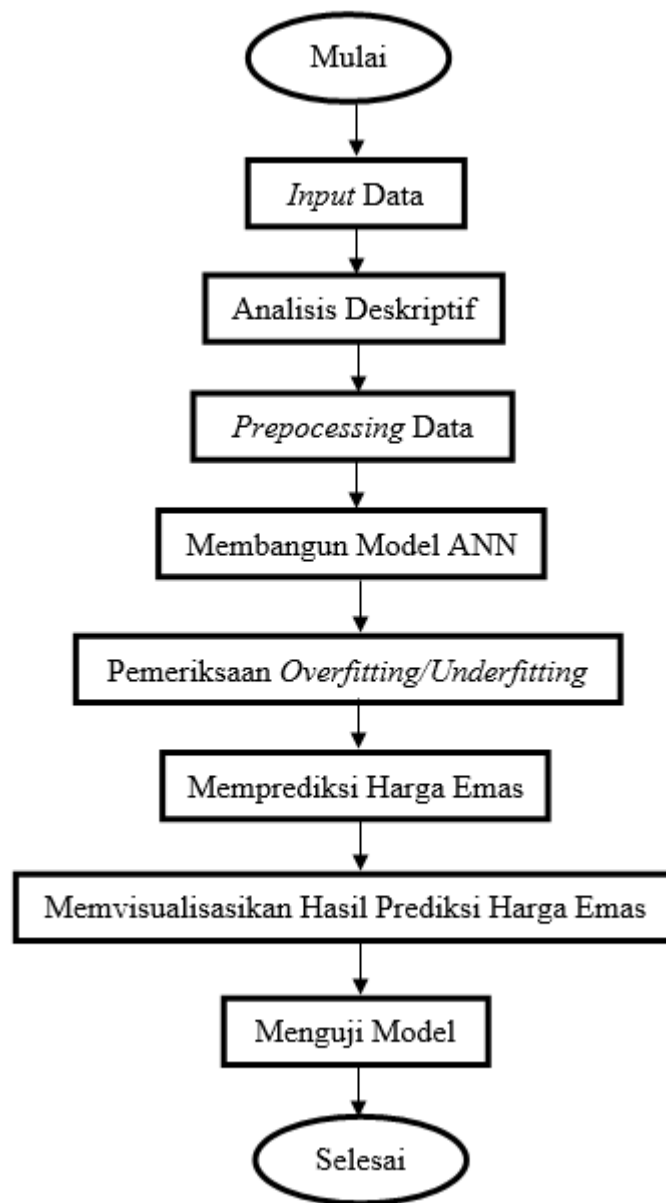
Penelitian ini menggunakan model ANN dengan algoritma *backpropagation* untuk memprediksi harga penutupan harga emas dengan bantuan bahasa pemrograman *Python*. Selanjutnya akan dilakukan evaluasi berdasarkan nilai kesalahan terkecil dan tingkat akurasi dari empat fungsi aktivasi yang digunakan. Adapun langkah-langkah penelitian ini, yaitu:

1. Melakukan *input* data harga emas yang didapatkan dari *website yahoo finance* ke dalam bahasa pemrograman *Python* menggunakan *Google Colab*.
2. Melakukan analisis deskriptif dengan memvisualisasikan data harga emas ke dalam plot *time series*.
3. Melakukan preprocessing data, yaitu:
 - a. Memeriksa apakah terdapat data hilang.
 - b. Melakukan transformasi data menggunakan *Min-Max Normalization*.
 - c. Melakukan Pembagian sebanyak 80% data *training*, 13, 59% data *validation* 6,41% data *testing*.
4. Membangun model ANN dengan algoritma *backpropagation*:
 - a. Menentukan pemilihan variabel *input* dan variabel *output*.
 - b. Menentukan komponen dan parameter model jaringan yaitu *hidden layer* serta *nodenya*, *batch size*, *learning rate*, *epoch*, dan berdasarkan empat fungsi aktivasi.
 - c. Membangun model dengan menggunakan komponen yang telah ditentukan

berdasarkan 4 fungsi aktivasi.

5. Melakukan pemeriksaan *overfitting* dan *underfitting* dengan menggunakan kurva pembelajaran dari nilai *loss* dan *validation loss* yang telah dibagi pada langkah 3c.
6. Melakukan prediksi menggunakan data *testing*.
7. Melakukan validasi model berdasarkan nilai MSE, nilai MAPE, dan akurasi yang sudah diperoleh.
8. Melakukan visualisasi data pada hasil prediksi harga emas dengan model terbaik berdasarkan empat fungsi aktivasi.

Secara singkat proses memprediksi harga emas dengan model ANN algoritma *backpropagation* menggunakan bahasa pemrograman *python* pada gambar berikut:



Gambar 13. Diagram Alir Proses Prediksi dengan *Artificial Neural Network*

V. KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan mengenai model ANN dengan algoritma *backpropagation* dengan membandingkan empat fungsi aktivasi dalam memprediksi harga emas diperoleh beberapa kesimpulan sebagai berikut:

1. Struktur terbaik model ANN dengan algoritma *backpropagation* berdasarkan empat fungsi aktivasi dalam memprediksi harga emas adalah menggunakan empat *node* pada *input layer*, satu *hidden layer* sebanyak empat *node*, *output layer*, *learning rate* sebesar 0.01, 64 *batch size*, 274 epoch, 5206 iterasi dan fungsi aktivasi *Linear*.
2. Prediksi harga emas berdasarkan model ANN menggunakan struktur jaringan terbaik memperoleh hasil prediksi yang cukup baik, karena hasil prediksi dengan data aktual menunjukkan perbedaan selisih harga yang kecil.
3. Pengujian Model ANN menggunakan struktur jaringan terbaik menghasilkan nilai MSE sebesar 0.00051 dan nilai MAPE sebesar 1.97989% dengan akurasi sebesar 98,02011%. Model yang dibangun dengan struktur terbaik juga tidak mengalami *overfitting* maupun *underfitting* dan dapat dikatakan model *good fit*. Oleh karena itu, performa model ANN sudah baik dan optimal.

5.2 Saran

1. Model ANN dan hasil prediksi yang telah diperoleh dapat digunakan untuk pembuatan sistem aplikasi untuk membantu memprediksi harga emas dengan mudah.
2. Bagi penelitian selanjutnya, perlu dilakukan peramalan harga emas dari model ANN yang telah diperoleh.
3. Bagi penelitian selanjutnya, untuk memudahkan dalam menentukan struktur atau parameter terbaik, perlu dilakukan *Hyperparameter Tuning* agar memudahkan proses pembangunan model ANN dan mendapatkan model yang lebih optimal.

DAFTAR PUSTAKA

- Astuti, E.D. 2009. *Pengantar Jaringan Syaraf Tiruan*. Star Publishing, Wonosobo.
- Ayub, M. 2007. Proses Data Mining dalam Pembelajaran Berbantuan Komputer. *Jurnal Sistem Informasi*. 2(1): 21-30.
- Brownlee, J. 2020. *Better Deep Learning: Train Faster, Reduce Overfitting, and Make Better Predictions*. Machine Learning Mistery, San Fransisco.
- Fausett, L.V. 1994. *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. Prentice Hall, New Jersey.
- Han, J., Kamber, M., dan Pei, J. 2012. *Data Mining: Concept and Techniques*. Third Edition. Morgan Kaufmann Publishers, Waltham.
- Heaton, J. 2008. *Introduction to Neural Networks with Java*. Heaton Research, Inc.
- Herjanto, E. 2006. *Manajemen Operasi Edisi Ketiga*. Grasindo, Jakarta.
- Kingma, D.P. dan Ba, J.L. 2015. Adam: A Method for Stochastic Optimization. *Proceeding a conference paper at the 3rd International Conference for Learning Representations*, San Diego.
- Kusumadewi, S. 2003. *Artificial Intelligence Teknik dan Aplikasi*. Graha Ilmu, Yogyakarta.
- Larose, D.T. 2005. *Discovering knowledge in data: an introduction to data mining*. Jhon Wiley & Sons Inc, USA.
- Lesnussa, Y.A., Mustamu, C.G., Lembang, F.K., dan Talakua, M.W. 2018. Application of Backpropagation Neural Networks in Predicting Rainfall Data in Ambon City. *International Journal of Artificial Intelligence Research*. 2(2): 41-50.

- Lestari, K.T.N., Albar, M.A., dan Afwani, R. 2019. Penerapan Metode Backpropagation Dalam Memprediksi Jumlah Kunjungan Wisatawan Ke Provinsi Nusa Tenggara Barat (NTB). *J-Cosine*. **3**(1): 39-48.
- Lewis, C.D. 1982. *Internatinal and Business Forecasting*. Butterworths, London.
- Lu, Y. 2017. *Deep Neural Networks and Fraud Detection*. U.U.D.M. Project Report 2017:38.
- Makridakis, S., Wheelwright, S.C., dan McGee, V.E. 1999. *Metode dan Aplikasi Prediksi*. Ed ke-2. Terjemah Untung Sus Andriyanto. Erlangga, Jakarta
- Miranda, A.C. dan Castano, V.M. 2017. Smart Frost Control in Greenhouses by Neural Networks Models. *Computers and Electronics in Agriculture*. **137**: 102–114. <https://doi.org/10.1016/j.compag.2017.03.024>. Diakses pada 19 Juni 2022
- Montgomery, D. 1990. *Pengantar Pengendalian Kualitas Statistik*. Gadjah Mada University Press, Yogyakarta.
- Nafi'iyah, N. 2016. Perbandingan Regresi Linear, Backpropagation, dan Fuzzy Mamdani dalam Prediksi Harga Emas, hlm. 291-296. *Prosiding Seminar Nasional Inovasi dan Aplikasi Teknologi Di Industri (SENIATI)*, Institut Teknologi Nasional Malang.
- Pakaja, F., Naba, A. dan Purwanto. 2012. Peramalan Penjualan Mobil Menggunakan Jaringan Syaraf Tiruan dan Certainty Factor. *Jurnal Electrics, Electronics, Communications, Controls, Informatics, Systems (EECCIS)*. **6**(1): 23-28.
- Patalay, S. dan Bandlamudi, M.R. 2021. Gold Price Prediction Using Machine Learning Model Trees. *Proceeding International Conference on Changing Business Paradigm (ICCBP)*, Murshidabad. <https://www.researchgate.net/publication/355357499>. Diakses pada 17 Januari 2022.
- Purnomo, M.H. dan Kurniawan A. 2006. *Supervised Neural Network*. Graha Ilmu, Surabaya.
- Puspitaningrum, D. 2006. *Pengantar Jaringan Saraf Tiruan*. Andi Offset, Yogyakarta.
- Rahmahwati, F.S. dan Arifin, F. 2017. Developing Artificial Neural Network Based on Visual Studio for Dance Assessment. *Jurnal Pendidikan Teknologi dan Kejuruan*. **23**(4): 402–411.

- Santoso, B. 2007. *Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Graha Ilmu, Yogyakarta.
- Sena, S. 2017. *Pengenalan Deep Learning Part 1: Neural Network*. <https://medium.com/@samuelsena/pengenalan-deep-learning-8fbb7d8028ac>. Diakses pada 09 Maret 2022 pukul 19.00
- Siang, J.J. 2005. *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*. Penerbit Andi, Yogyakarta.
- Siregar, S.P. dan Wanto, A. 2017. Analysis of Artificial Neural Network Accuracy Using Backpropagation Algorithm in Predicting Process (Forecasting). *International Journal of Information System & Technology*. **1**(1): 34-42.
- Suzuki, K. 2013. *Artificial Neural Network-Architectures and Applications*. InTech, Croatia.
- Stern, H.S. 1996. Neural Network in Applied Statistics. *Technometrics*. **3**(38): 205-2014.
- Tanaka, M. dan Okutomi, M.A. 2014. Novel Inference of a Restricted Boltzmann Machine. *Proceeding of 22nd International Conference on Pattern Recognition (ICPR)*, Tokyo. <https://ieeexplore.ieee.org/document/6976981>. Diakses pada 13 februari 2022.
- Turban, E. 2005. *Decision Support System and Intellegent System (Sistem Pendukung Keputusan dan Sistem Cerdas)*. Andi, Yogyakarta.
- Untoro, A.B. 2020. Prediksi Harga Saham Dengan Menggunakan Jaringan Syaraf Tiruan. *Jurnal Teknologi Informatika dan Komputer MH*. **6**(2): 103-111.