

**SKRIPSI**

**MEMBANGUN APLIKASI UNTUK MEMPREDIKSI  
KEBERLANJUTAN STUDI MAHASISWA S1 UNIVERSITAS  
HASANUDDIN MENGGUNAKAN *NEURAL NETWORK* PADA  
*MICROFRAMEWORK FLASK***

Disusun dan diajukan oleh

**MUNAWIR DJAMALUDDIN**

**H13114314**



**PROGRAM STUDI SISTEM INFORMASI  
DEPARTEMEN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS HASANUDDIN  
MAKASSAR**

**2021**

**MEMBANGUN APLIKASI UNTUK MEMPREDIKSI  
KEBERLANJUTAN STUDI MAHASISWA S1 UNIVERSITAS  
HASANUDDIN MENGGUNAKAN *NEURAL NETWORK* PADA  
*MICROFRAMEWORK FLASK***

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana  
Komputer pada Program Studi Sistem Informasi Departemen Matematika  
Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin**

**MUNAWIR DJAMALUDDIN**

**H13114314**

**PROGRAM STUDI SISTEM INFORMASI  
DEPARTEMEN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS HASANUDDIN  
MAKASSAR**

**2021**

## LEMBAR PERNYATAAN KEOTENTIKAN

Yang bertanda tangan dibawah ini :

Nama : Munawir Djamaluddin

NIM : H13114314

Program Studi : Sistem Informasi

Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

**MEMBANGUN APLIKASI UNTUK MEMPREDIKSI KEBERLANJUTAN  
STUDI MAHASISWA S1 UNIVESITAS HASANUDDIN MENGGUNAKAN  
NEURAL NETWORK PADA MICROFRAMEWORK FLASK**

Adalah benar hasil karya saya sendiri bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini merupakan hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 02 Juni 2021

Yang Menyatakan



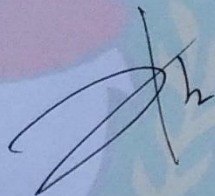
Munawir Djamaluddin

**MEMBANGUN APLIKASI UNTUK MEMPREDIKSI KEBERLANJUTAN  
STUDI MAHASISWA S1 UNIVESITAS HASANUDDIN MENGGUNAKAN  
NEURAL NETWORK PADA MICROFRAMEWORK FLASK**

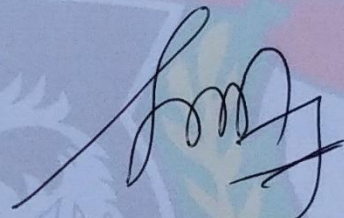
Disetujui oleh :

**Pembimbing Utama**

**Pembimbing Pertama**



**Dr. Hendra, S.Si., M.Kom**  
NIP. 19760102 200212 1 001



**Supri Bin Hj Amir, S.Si., M.Eng**  
NIP. 19880504 201903 1 012

**Ketua Program Studi**



**Dr. Muhammad Hasbi, M.Sc**  
NIP. 19630720 198903 1 003



**Pada Tanggal : 02 Juni 2021**

**LEMBAR PENGESAHAN SKRIPSI**

**MEMBANGUN APLIKASI UNTUK MEMPREDIKSI KEBERLANJUTAN  
STUDI MAHASISWA S1 UNIVESITAS HASANUDDIN MENGGUNAKAN  
NEURAL NETWORK PADA MICROFRAMEWORK FLASK**

Disusun dan diajukan oleh :

**MUNAWIR DJAMALUDDIN**

**H13114314**

Telah dipertahankan dihadapan Panitia Ujian yang dibentuk dalam rangka  
Penyelesaian Studi Program Sarjana pada Program Studi Sistem Informasi  
Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin  
Pada tanggal 02 Juni 2021  
dan dinyatakan telah memenuhi syarat kelulusan.

Menyetujui

**Pembimbing Utama**

**Pembimbing Pertama**

**Dr. Hendra, S.Si., M.Kom**  
NIP. 19760102 200212 1 001

**Supri Bin Hj Amir, S.Si., M.Eng**  
NIP. 19880504 201903 1 012

**Ketua Program Studi**

**Dr. Muhammad Hasbi, M.Sc**  
NIP. 19630720 198903 1 003



## KATA PENGANTAR



Segala puji bagi Allah Azza Wa Jalla Rabb semesta alam yang ditangan-Nya terenggam nyawa seluruh makhluk semesta alam, yang Maha kekal sebelum sesuatunya ada, dan akan tetap kekal setelah segala sesuatunya tiada. Shalawat serta salam semoga selalu dilimpahkan kepada Nabi Muhammad ﷺ dan kepada para keluarga serta Sahabat beliau. Alhamdulillah Wasyukurillah, berkat pertolongan Allah akhirnya skripsi dengan judul “ **Membangun Aplikasi untuk Memprediksi Keberlanjutan Studi Mahasiswa S1 Universitas Hasanuddin Menggunakan Neural Network pada Micoframework Flask** ” yang disusun sebagai salah satu syarat akademik untuk meraih gelar sarjana komputer pada Program Studi Sistem Informasi Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin ini dapat dirampungkan. Penulis berharap skripsi ini dapat memberikan tambahan pengetahuan baru bagi para pembelajar dalam program studi Sistem Informasi.

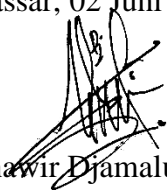
Penulis menyadari bahwa penyelesaian tugas akhir ini masih terdapat kekurangan dalam penyelesaian dan bantuan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada kedua orang tua penulis, Ayahanda Almarhum **Djamaluddin** dan Ibunda **St. Amirah** yang telah mendidik dengan penuh kesabaran, terimakasih telah mencurahkan kasih sayang yang tak pernah putus, kesungguhan dalam memberikan dukungan moril serta tak kenal lelah dalam memanjatkan doa serta memberikan nasihat dan motivasi kepada penulis selama menjalani proses pendidikan. Untuk kakak-kakak penulis **Mutmainnah Djamaluddin** dan **Magfirah Djamaludin** serta keluarga yang senantiasa memberikan dukungan dan doa, terima kasih atas segala perhatian yang telah kalian berikan kepada penulis. Tugas akhir ini hanya setitik kebahagiaan kecil yang bisa penulis persembahkan kepada kalian.

Penghargaan yang tulus dan ucapan terima kasih dengan penuh keikhlasan juga penulis ucapkan kepada :

1. Ibu **Prof. Dr. Dwia Aries Tina Palubuhu, MA**, selaku Rektor Universitas Hasanuddin berserta seluruh jajarannya.
2. Bapak **Dr. Eng. Amiruddin**, selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin beserta seluruh jajarannya.
3. Bapak **Prof. Dr. Nurdin, S.Si., M.Si**, selaku Ketua Departemen Matematika dan segenap dosen pengajar dan staf Departemen Matematika yang telah membekali ilmu dan kemudahan kepada penulis dalam berbagai hal selama menjadi mahasiswa di Departemen Matematika.
4. Bapak **Dr. Hendra, S.Si., M.Kom**, selaku selaku dosen pembimbing utama sekaligus penasehat akademik atas nasehat, dukungan, doa dan dengan setulus hati telah meluangkan waktunya ditengah berbagai kesibukan dan prioritasnya untuk membimbing penulis menyelesaikan tugas akhir ini.
5. Bapak **Supri bin Hj. Amir, S.Si., M.Eng**, selaku dosen pembimbing pertama sekaligus senior atas segala masukan bantuan, nasehat serta motivasi yang diberikan kepada penulis dalam membimbing penulis menyelesaikan tugas akhir ini.
6. Bapak **Prof. Dr. Moh. Ivan Azis, M.Sc** selaku ketua penguji dan Bapak **Dr. Eng. Armin Lawi, S.Si., M.Eng** selaku sekretaris penguji atas kesediaannya untuk memberikan saran dan arahan kepada penulis dalam penyusunan skripsi ini.
7. **Dosen pengajar Departemen Matematika** yang telah membekali ilmu kepada penulis selama menjadi mahasiswa di Departemen Matematika. Serta Staf Departemen Matematika yang telah membantu banyak dalam perkuliahan.
8. Saudara tak sedarah **TRANSPOSE 2014**, terkhusus ketua angkatan **Fandy Heribet**, saudara-saudara di grup **HARMOKO vs SOBAT NESTAPA** terkhusus **Arfyan Saputra, Miftah Farid, Setiawan Ahmad, Nivel Saputra Wahid, Muh. Alfajar, Hedi Kuswanto, Rachmat Darmawan, Ahmad Husain, Muhammad Mappanyompa, Muh. Sarwan, Muh. Ij'lal, Andi Muh. Fawzy, Syahrul** serta seluruh saudara – saudari tanpa terkecuali Transpose 2014 terkhusus **Nurfahmi Afdhaly, A. Navira Indiyani Tamar, Amalia, Asti Febrianti Asis, Muslimah, Sukmawati**, terima kasih telah menjadi rumah ke dua bagi penulis.

9. Seluruh **Anggota Himatika FMIPA Unhas**, kanda POLINOM 2011, REKURENSI 2012, BINOMIAL 2013, adinda SIMETRIS 2015, ALGORITMA 2016, DISKRIT 2017, INTEGRAL 2018, POLIGON 2019 dan HORIZONTAL 2020. Terima kasih telah menjadi keluarga dan memberikan kenangan yang tidak akan penulis lupakan.
10. Saudara – saudara **MIPA 2014** tanpa terkecuali terkhusus ketua angkatan **Muh. Sidiq Tolleng** semoga selalu tetap pada slogan “Kita Semua Sama”.
11. Sahabat – sahabat “Projek Terencana” **Anny Nur Sriansyah San, Nurhakimah Mujahid, Aulia Azizah** dan **Novika Amelia** atas doa, semangat serta kebersamaannya selama ini yang banyak membantu penulis.
12. Rekan – rekan guru dan staf di **SMK Laniang Makassar**, terkhusus **Utty, Kak Lia, Kak Eki, Kak Ikhwan, Ilham** dan **Ibu Seni** yang selama ini banyak membantu penulis dalam hal pekerjaan maupun memberi motivasi dan semangat dalam menyelesaikan penyusunan skripsi ini.
13. Seluruh teman-teman KKN Takalar Unhas Gel.96, terkhusus kepada teman **Posko Desa Towata**, terima kasih atas waktu singkat dan pengalaman yang bermakna.
14. Semua pihak yang tidak dapat disebutkan satu persatu, atas segala bentuk kontribusi, partisipasi, serta motivasi yang diberikan kepada penulis selama ini. Semoga apa yang telah diberikan akan dilipatgandakan oleh Allah Subhanahu Wa Ta’ala. Aamiin.

Makassar, 02 Juni 2021



Munawir Djamaluddin



## **PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Hasanuddin, saya yang bertanda tangan dibawah ini :

Nama : Munawir Djameluddin  
NIM : H13114314  
Program Studi : Sistem Informasi  
Departemen : Matematika  
Fakultas : Matematika dan Ilmu Pengetahuan Alam  
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Prediktor Royalti Noneklusif (*Non-exclusive Royalty-Free Right*)** atas tugas akhir saya yang berjudul :


### **Membangun Aplikasi untuk Memprediksi Keberlanjutan Studi Mahasiswa S1 Universitas Hasanuddin Menggunakan Neural Network pada Microframework Flask**

Beserta perangkat yang ada (jika diperlukan). Terkait dengan hal diatas, maka pihak universitas berhak menyimpan, mengalih-media/format-kan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar pada 02 Juni 2021

Yang menyatakan

  
(Munawir Djameluddin)

## ABSTRAK

Prediksi keberlanjutan masa studi dibutuhkan oleh manajemen perguruan tinggi dalam menentukan kebijakan preventif terkait pencegahan dini kasus *Drop Out* (DO). Penelitian ini bertujuan untuk melakukan prediksi masa studi mahasiswa S1 Universitas Hasanuddin yang selanjutnya akan dibangun model untuk diaplikasikan dalam sebuah aplikasi web. Aplikasi ini akan memberikan kemudahan dalam melakukan dasar pertimbangan dalam pengambilan keputusan bagi pihak manajemen universitas dalam melakukan prediksi keberlanjutan studi mahasiswa. Model yang digunakan untuk memprediksi keberlanjutan masa studi adalah algoritma *Neural Network* (NN) dengan arsitektur *Backpropagation* yang kemudian dilanjutkan pembuatan aplikasi pada *Microframework Flask*. Dari penelitian ini ditemukan bahwa dari data masa studi mahasiswa S1 Universitas Hasanuddin dan diolah dengan menggunakan Algoritma *Neural Network* diperoleh akurasi model yaitu 96%. Setelah model didapatkan akan disimpan dan digunakan pada pembuatan aplikasi web *Microframework Flask*.

**Kata kunci :** Prediksi masa studi, *Neural Network*, *Backpropagation*, *Flask*

## ABSTRACT

College management needs to predict the sustainability of the study period in determining preventive policies related to the early prevention of Drop Out (DO) cases. This research aims to predict the study period of undergraduate students of Hasanuddin University which will be built a model to be applied in a web application. This application will provide convenience in conducting basic considerations in decision making for the university management in predicting the sustainability of student studies. The model used to predict the sustainability of the study period is Neural Network (NN) algorithm with Backpropagation architecture which then continues to build applications on Microframework Flask. From this research, it was found that from the study period data of undergraduate students of Hasanuddin University and processed using Neural Network Algorithm obtained the accuracy of the model is 96%. After the model is available it will be saved and used in the creation of the Microframework Flask web application.

**Keywords** : Study period prediction, *Neural Network*, *Backpropagation*, *Flask*

## DAFTAR ISI

HALAMAN SAMBUNG .....	i
HALAMAN JUDUL .....	ii
LEMBAR PERNYATAAN KEOTENTIKAN .....	<b>Error! Bookmark not defined.</b>
LEMBAR PERSETUJUAN PEMBIMBING .....	<b>Error! Bookmark not defined.</b>
LEMBAR PENGESAHAN SKRIPSI .....	<b>Error! Bookmark not defined.</b>
KATA PENGANTAR .....	vi
PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS .....	ix
ABSTRAK .....	x
ABSTRACT .....	xi
DAFTAR ISI .....	xii
DAFTAR GAMBAR .....	xv
DAFTAR TABEL .....	xvi
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan Penelitian .....	2
1.4 Batasan Masalah .....	2
1.5 Manfaat Penelitian .....	3
BAB II TINJAUAN PUSTAKA .....	4
2.1 Evaluasi Keberlanjutan Studi .....	4
2.2 <i>Neural Network</i> .....	6
2.2.1 Struktur Neural Network .....	7
2.3 Algoritma Pembelajaran .....	10
2.4 Propagasi Balik ( <i>Backpropagation</i> ) .....	11
2.4.1 Algoritma Pelatihan pada Propagasi Balik .....	11
2.4.2 Akurasi dan Generalisasi .....	15

2.5 <i>Confusion Matrix</i> .....	16
2.6 <i>Synthetic Minority Oversampling Technique</i> .....	17
2.7 Python.....	18
2.7.1 Flask.....	18
2.7.2 <i>Instalasi Flask</i> .....	19
2.7.3 Fitur – fitur Flask .....	19
<b>BAB III METODE PENELITIAN</b> .....	<b>21</b>
3.1 Waktu dan Lokasi Penelitian.....	21
3.2 Sumber Data.....	21
3.3 Metode Analisis.....	21
3.4 Instrumen Penelitian.....	22
3.5 Diagram Alir Penelitian ( <i>Flow Chart</i> ) .....	23
<b>BAB IV HASIL DAN PEMBAHASAN</b> .....	<b>24</b>
4.1. Preprocessing .....	24
4.2. Deskripsi Data.....	24
4.3. Resampling Data.....	25
4.4. Data Latih dan Data Uji .....	26
4.5. <i>Confusion Matrix</i> .....	28
4.6. Desain Sistem .....	29
4.7. Pembuatan Aplikasi Web dengan Flask .....	29
4.7.1 Memuat Data.....	30
4.7.2 Mendefinisikan Keras Model.....	30
4.7.3 Mengkompilasi Keras Model.....	31
4.7.4 Mengeksekusi Keras Model.....	31
4.7.5 Mengevaluasi Keras Model .....	31
4.7.6 Pembuatan Template.....	32
4.7.7 Pembuatan Objek Aplikasi .....	33
4.7.8 Menjalankan Aplikasi .....	35
<b>BAB 5 KESIMPULAN DAN SARAN</b> .....	<b>38</b>
5.1 Kesimpulan.....	38
5.2 Saran.....	38

DAFTAR PUSTAKA .....	39
LAMPIRAN.....	41

## DAFTAR GAMBAR

Gambar 2.1 Saraf Biologis dan Struktur Saraf Tiruan.....	7
Gambar 2.2 Jaringan dengan lapisan tunggal ( <i>single layer net</i> ).....	7
Gambar 2.3 Jaringan dengan banyak lapisan ( <i>multilayer net</i> ).....	8
Gambar 2.4 Jaringan dengan lapisan kompetitif ( <i>competitive layer net</i> ) .....	9
Gambar 2.5 Arsitektur Jaringan Saraf.....	10
Gambar 2.6 Arsitektur <i>Backpropagation</i> .....	14
Gambar 2.7 Akurasi dan Generalisasi.....	15
Gambar 3.1 Diagram Alir Penelitian .....	23
Gambar 4. 1 Grafik <i>Model Accuracy</i> .....	27
Gambar 4.2 Grafik <i>Loss Function</i> .....	28
Gambar 4.3 <i>Activity Diagram</i> .....	29
Gambar 4. 4 Halaman Dashboard.....	32
Gambar 4.5 Halaman Form.....	33
Gambar 4.6 Halaman About .....	33
Gambar 4.7 Tampilan <i>running app.py</i> .....	36
Gambar 4.8 Tampilan hasil prediksi lanjut .....	36
Gambar 4.9 Tampilan hasil prediksi tidak lanjut.....	37

## DAFTAR TABEL

Tabel 2.1 Kesetaraan Nilai Angka, Nilai Huruf, dan Nilai Konversi .....	4
Tabel 2.2 <i>Confusion Matrix</i> .....	16
Tabel 2.3 Sumber Data.....	21
Tabel 4.1 Data akademik yang didapat dari PTIK Universitas Hasanuddin .....	24
Tabel 4.2 Variabel data .....	25
Tabel 4.3 Gambaran umum tentang data .....	25
Tabel 4.4 Gambaran umum tentang data yang telah diresampling.....	26
Tabel 4.5 <i>Confusion Matrix</i> .....	28
Tabel 4.6 Hasil kinerja klasifikasi model.....	28



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Semakin ketatnya persaingan dalam mendapatkan lapangan pekerjaan menuntut perguruan tinggi menghasilkan sarjana yang berkualitas dan memiliki daya saing. Untuk itu, setiap perguruan tinggi selalu melakukan evaluasi performansi mahasiswa. Hasil evaluasi tersebut disimpan dalam basis data akademik. Data tersebut dapat digunakan untuk sebagai pendukung keputusan oleh manajemen perguruan tinggi. Salah satu variabel indikator efisiensi proses pendidikan adalah informasi mengenai lama masa studi mahasiswa.

*Artificial Neural Network* (ANN) sejak diperkenalkan pada sekitar tahun 1940 telah banyak diimplementasikan pada berbagai bidang keilmuan. ANN banyak digunakan untuk melakukan prediksi atau peramalan. Williams dan Li (2008) telah meneliti penggunaan ANN dengan algoritma training back-propagation untuk melakukan prediksi pacuan kuda di Jamaika. ANN dengan jenis *feed forward network* atau *backpropagation* yang digunakan dalam penelitian ini telah terbukti memberikan hasil yang baik untuk keperluan prediksi (Meinanda dkk, 2009).

Dengan perkembangan teknologi saat ini, terdapat banyak alat untuk membantu kegiatan *web development*, salah satunya adalah *web framework* yang dibuat menggunakan bahasa Python, yaitu Flask. Flask adalah sebuah alat yang membantu membuat kerangka untuk sebuah *web* dan dengan menggunakan Flask, pengembang pemula pun dapat menciptakan sebuah *web* yang bagus. (Irsyad, 2018)

Dengan acuan kesempatan penelitian yang tersedia berdasarkan penelitian sebelumnya maka pada penelitian ini akan diteliti *variable predictor* dari data akademis yang berpengaruh terhadap keberlanjutan masa studi dan pembuatan model *Neural Network* untuk prediksi keberlanjutan masa studi, kemudian akan dibuatkan aplikasi berbasis web dengan menggunakan Flask. Model prediksi tersebut digunakan untuk menentukan kebijakan terhadap mahasiswa yang diprediksi memiliki masa studi melebihi batas.

Demikian, maka akan disusun sebuah tugas akhir dengan judul **“Membangun Aplikasi Untuk Memprediksi Keberlanjutan Studi Mahasiswa S1 Universitas Hasanuddin Menggunakan *Neural Network* Pada *Microframework Flask*”**

### **1.2 Rumusan Masalah**

Berdasarkan latar belakang tersebut maka rumusan masalah yang akan di bahas pada penelitian ini adalah:

1. Bagaimana melakukan prediksi masa studi mahasiswa S1 dengan menggunakan algoritma *Neural Network*?
2. Bagaimana membangun aplikasi prediksi keberlanjutan studi mahasiswa S1 Universitas Hasanuddin dengan menggunakan *Microframework Flask*?

### **1.3 Tujuan Penelitian**

Tujuan penelitian untuk penulisan tugas akhir ini adalah :

1. Melakukan prediksi masa studi mahasiswa S1 dengan menggunakan algoritma *Neural Network*.
2. Membangun aplikasi prediksi keberlanjutan studi mahasiswa S1 Universitas Hasanuddin dengan menggunakan *Microframework Flask*.

### **1.4 Batasan Masalah**

Agar pembahasan pada penelitian ini lebih terarah, maka masalah yang akan dibahas dibatasi pada :

1. Data yang digunakan dalam melakukan prediksi studi mahasiswa S1 dengan menggunakan algoritma *Neural Network* adalah data mahasiswa S1 Universitas Hasanuddin angkatan 2011 - 2018
2. Penelitian ini menggunakan data yang akan diolah dengan menggunakan *Neural Network*.
3. *Output* yang dihasilkan dari program aplikasi merupakan nilai numerik dan akan ditampilkan dengan *Microframework Flask*.

### **1.5 Manfaat Penelitian**

Adapun manfaat dari penelitian yaitu memberikan kemudahan dalam melakukan dasar pertimbangan dalam pengambilan keputusan bagi pihak manajemen universitas dalam melakukan prediksi keberlanjutan studi mahasiswa S1 Universitas Hasanuddin.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Evaluasi Keberlanjutan Studi**

Berdasarkan peraturan rektor Universitas Hasanuddin Nomor : 2781/UN4.1/KEP/2018 tentang penyelenggaraan program sarjana Universitas Hasanuddin, pasal 1 ayat 26 putus studi adalah mahasiswa yang tidak dapat memenuhi persyaratan akademik ataupun non akademik sesuai ketentuan yang berlaku. Mahasiswa dinyatakan putus studi apabila :

- a. Habis masa studinya sesuai peraturan yang berlaku
- b. Mengundurkan diri atas permintaan sendiri
- c. Evaluasi pada akhir semester IV tidak mencapai IPK sekurang-kurangnya 2.00 yang diperhitungkan oleh pemerintah dan atau Unhas
- d. Dikeluarkan karena melakukan tindak pidana dan/atau melanggar ketentuan yang ditetapkan oleh pemerintah dan atau Unhas

Berdasarkan pasal 7, masa studi program sarjana dijadwalkan untuk 8 (delapan) semester dan dapat ditempuh dalam waktu sekurang-kurangnya 6 (enam) semester dan selama-lamanya 14 (empat belas) semester. Mahasiswa program sarjana yang tidak dapat menyelesaikan studinya pada akhir semester ke-14 dinyatakan putus studi. Adapun di pasal 6 ayat 5, dijelaskan bahwa mahasiswa yang berstatus tidak aktif selama 2 (dua) semester berturut-turut dinyatakan putus studi

Nilai hasil belajar dinyatakan dengan huruf dengan konversi bentuk bilangannya, yaitu:

Tabel 2.1 Kesetaraan Nilai Angka, Nilai Huruf, dan Nilai Konversi

Rentang Nilai Angka	Nilai Huruf	Nilai Konversi
85 - 100	A	4.00
80 - < 85	A <sup>-</sup>	3.75
75 - < 80	B <sup>+</sup>	3.50

70 - < 75	B	3.00
65 - < 70	B <sup>-</sup>	2.75
60 - < 65	C <sup>+</sup>	2.50
50 - < 60	C	2.00
40 - < 50	D	1.00
< 40	E	0.00

Nilai A sampai D adalah nilai lulus, sedangkan nilai E adalah nilai tidak lulus. Matakuliah dengan nilai E harus diulang dan diprogramkan pada semester berikutnya sampai mendapatkan nilai lulus. Nilai lulus tidak dapat diulangi pada semester selanjutnya, kecuali nilai D dan nilai C dengan syarat telah melulusi minimal 114 sks dan sementara  $IPK < 3.00$  dan hanya boleh diulangi satu kali dan nilai terakhir yang diakui. Sedangkan nilai K (kosong) diberikan kepada mahasiswa yang mengundurkan diri dari matakuliah secara sah dan tertulis atas persetujuan Penasihat Akademik dan diketahui oleh KPS dan Dekan, serta ditetapkan dengan surat izin dari Rektor.

Keberhasilan studi mahasiswa program sarjana dinyatakan dengan indeks prestasi (IP). Indeks Prestasi Semester (IPS) dihitung dari nilai konversi dan nilai bobot kredit setiap matakuliah yang tercantum dalam KRS dengan rumus sebagai berikut: (Universitas Hasanuddin, 2018)

$$IPS = \frac{\sum(N_i \times K_i)}{\sum K_i} \quad (2.1)$$

dimana

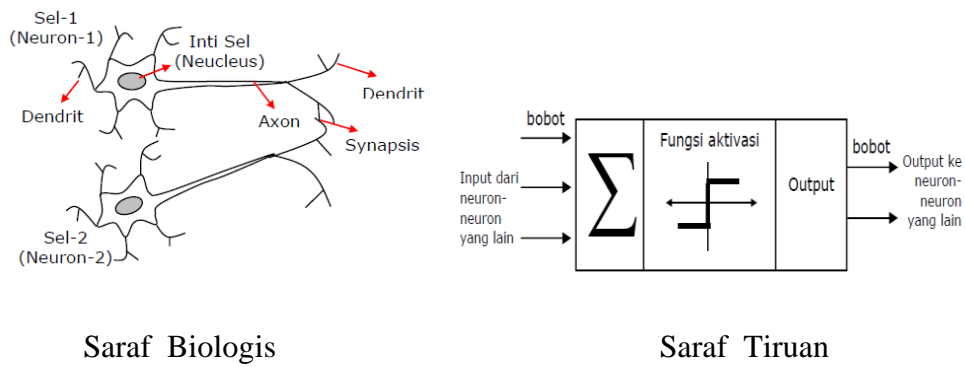
$K_i$  = bobot sks matakuliah ke-i dalam satu semester.

$N_i$  = nilai mutu setelah disetarakan ke nilai konversi matakuliah ke-i.

Indeks Prestasi Kumulatif (IPK) dihitung dari semua nilai matakuliah yang sudah dilulusi oleh mahasiswa dengan menggunakan rumus diatas.

## 2.2 Neural Network

*Neural Network* (NN) atau sering disebut dengan Jaringan Saraf Tiruan (JST), merupakan salah satu teknik klasifikasi yang cukup handal. *Neural Network* merupakan jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan jaringan saraf manusia. *Neural Network* adalah salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah buatan digunakan karena jaringan saraf ini diimplementasikan dengan menggunakan program computer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran (Fausett 1994). Dalam syaraf biologis, setiap sel saraf (neuron) akan memiliki satu inti sel yang bertugas untuk melakukan pemrosesan informasi yang akan diterima oleh dendrit. Selain menerima informasi, dendrit juga menyertai axon sebagai keluaran dari suatu pemrosesan informasi. Informasi hasil olahan ini menjadi masukan bagi neuron lain. jika memenuhi batas tertentu, yang sering dikenal dengan nama nilai ambang (*threshold*) yang dikatakan teraktivasi. Seperti halnya otak manusia, JST juga terdiri dari beberapa neuron yang berhubungan untuk mentransformasikan informasi yang terima melalui sambungan keluarnya. Hubungan ini dikenal dengan nama bobot. Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tersebut. Informasi (*input*) kan dikirim ke neuron dengan bobot kedatangan. Input ini akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang datang. Hasil penjumlahan ini kemudian akan dibandingkan dengan nilai suatu ambang (*threshold*) tertentu melalui fungsi aktivasi setiap neuron (Gambar 2.1). Apabila input tersebut melewati suatu nilai ambang tertentu, maka neuron tersebut akan diaktifkan, tapi kalau tidak, maka neuron tersebut tidak akan diaktifkan. Apabila neuron tersebut diaktifkan maka neuron tersebut akan mengirimkan *output* melalui bobot-bobot outputnya ke semua neuron yang berhubungan dengannya. Neuron-neuron akan dikumpulkan dalam lapisan-lapisan (*layer*) yang disebut dengan lapisan neuron (*neuron layer*) yang saling berhubungan. Informasi akan dirambatkan mulai dari lapisan input sampai ke lapisan output melalui lapisan lainnya yang sering dikenal dengan lapisan tersembunyi (*hidden layer*), dan perambatannya tergantung algoritma pembelajarannya (Kusumadewi & Hartati, 2010).



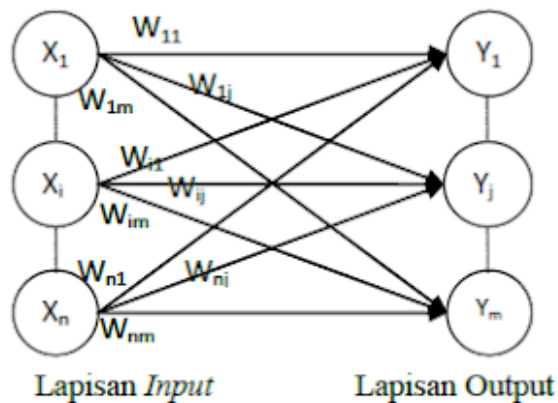
Gambar 2.1 Saraf Biologis dan Struktur Saraf Tiruan (Kusumadewi & Hartati, 2010).

### 2.2.1 Struktur Neural Network

*Artificial neural network (ANN)* terinspirasi dari kesadaran atas *complex learning system* pada otak yang terdiri dari set-set neuron yang saling berhubungan secara dekat. Jaringan neuron mampu melakukan tugas yang sangat kompleks seperti klasifikasi dan pemahaman pola. ANN dapat memperkirakan rentang yang cukup luas suatu model statistika dan fleksibel dalam menggambarkan model (linier maupun non linier). ANN dapat digunakan untuk permasalahan yang sama dengan permasalahan statistika multivariat seperti *multiple regression*, *discriminant analysis*, dan *cluster analysis*. Dalam banyak kasus, hasil yang didapat dengan ANN dapat dibandingkan dengan model statistika multivariat. (Meinanda dkk, 2009)

Menurut (Haykin, 2009) secara umum ada tiga jenis arsitektur dari *Neural Network* yaitu:

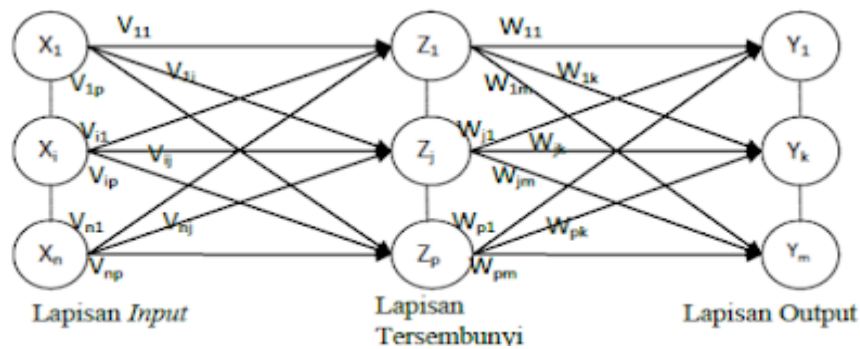
- a. Jaringan dengan lapisan tunggal (*single layer net*)



Gambar 2.2 Jaringan dengan lapisan tunggal (*single layer net*) (Riadi, 2016)

Di dalam *Neural Network* dengan satu layer, neuron-neuron diorganisasi dalam bentuk layer-layer. Dalam bentuk paling sederhana dari *Neural Network* dengan satu layer, kita mempunyai sebuah input layer dari node sumber di mana informasi diproyeksikan ke output layer dari neuron tapi tidak bisa sebaliknya. Dengan kata lain, jaringan ini adalah tipe feed forward. Input layer dari node sumber tidak dihitung karena tidak ada perhitungan yang dilakukan.

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

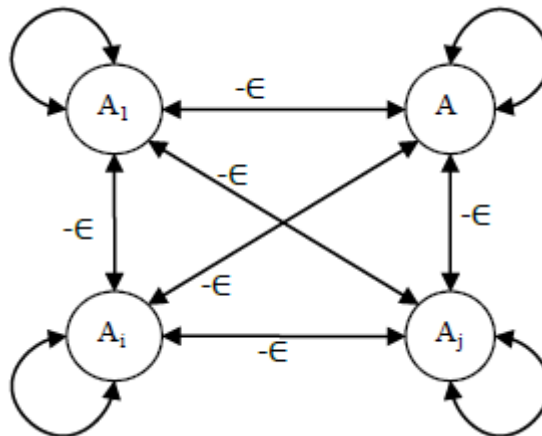


Gambar 2.3 Jaringan dengan banyak lapisan (*multilayer net*) (Riadi, 2016)

*Multilayer net* merupakan jaringan dengan satu atau lebih lapisan tersembunyi (hidden layer). Jaringan multi lapis ini memiliki kemampuan lebih dalam memecahkan masalah bila dibandingkan dengan jaringan lapis tunggal, namun pelatihannya mungkin lebih rumit. Pada beberapa kasus, pelatihan pada jaringan ini lebih baik karena memungkinkan bagi jaringan untuk memecahkan masalah yang tidak dapat diselesaikan jaringan berlapis tunggal karena jaringan tidak bisa dilatih untuk menampilkan secara benar.



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



Gambar 2.4 Jaringan dengan lapisan kompetitif (*competitive layer net*) (Riadi, 2016)

Bentuk lapisan kompetitif merupakan *Neural Network* yang sangat besar. Interkoneksi antar neuron pada lapisan ini tidak ditunjukkan pada arsitektur seperti jaringan yang lain. Pada jaringan ini sekumpulan neuron bersaing untuk mendapatkan hak menjadi aktif atau sering pula disebut dengan prinsip winner takes all atau yang menanglah yang mengambil semua bagiannya.

Hubungan antar neuron dalam jaringan saraf mengikuti pola tertentu tergantung pada arsitektur jaringan sarafnya. (Kusumadewi & Hartati, 2010)

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

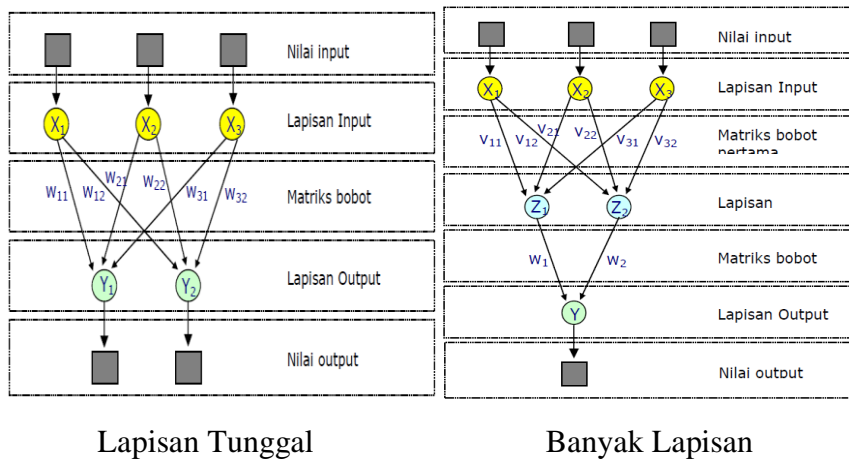
Jaringan dengan lapisan tunggal hanya memiliki satu lapisan dengan bobot-bobot terhubung. Jaringan ini hanya menerima input kemudian secara langsung akan mengolahnya menjadi output tanpa harus melalui lapisan tersembunyi.

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

Jaringan dengan banyak lapisan memiliki satu atau lebih lapisan yang terletak diantara lapisan input dan lapisan output (memiliki satu atau lebih lapisan tersembunyi). Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih sulit daripada lapisan tunggal.

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

Pada jaringan ini sekumpulan neuron bersaing untuk mendapatkan hak menjadi aktif.



Gambar 2.5 Arsitektur Jaringan Saraf (Kusumadewi & Hartati, 2010)

### 2.3 Algoritma Pembelajaran

Salah satu bagian terpenting dari konsep *Neural Network* adalah terjadinya proses pembelajaran. Tujuan utama dari proses pembelajaran adalah melakukan pengaturan terhadap bobot-bobot yang ada pada jaringan saraf, sehingga diperoleh bobot akhir yang tepat yang sesuai dengan pola data yang dilatih. Cara berlangsungnya pembelajaran atau pelatihan JST dikelompokkan menjadi 3 (tiga), yaitu (Puspitaningrum, 2006) :

1. Pembelajaran terawasi (*supervised learning*)

Pada metode ini setiap pola yang diberikan ke dalam JST telah diketahui outputnya. Selisih antara pola output yang dihasilkan dengan output yang dikehendaki (output target) yang disebut error digunakan untuk mengoreksi bobot JST sehingga JST mampu menghasilkan output sedekat mungkin dengan pola target yang telah diketahui oleh JST.

2. Pembelajaran yang tak terawasi (*unsupervised learning*)

Pada metode pembelajaran yang tak terawasi tidak memerlukan target output. Pada metode ini tidak ditentukan hasil yang seperti apakah yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam suatu range tertentu tergantung pada nilai input yang diberikan. Tujuan pembelajaran ini adalah mengelompokkan unit-unit yang hampir sama dalam suatu area tertentu. Pembelajaran ini biasanya sangat cocok untuk pengelompokan (klasifikasi) pola.

### 3. Pembelajaran Hibrida (*Hybrid Learning*)

Pembelajaran Hibrida merupakan kombinasi dari metode pembelajaran supervised learning dan unsupervised learning. Sebagian dari bobot-bobotnya ditentukan melalui pembelajaran terawasi dan sebagian lainnya melalui pembelajaran tak terawasi.

## 2.4 Propagasi Balik (*Backpropagation*)

Propagasi balik (*backpropagation*) adalah salah satu algoritma pembelajaran dalam teknik JST yang sering digunakan untuk pencocokan pola. Propagasi balik merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh *perceptron* dengan banyak lapisan (*multilayer perceptron*) untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya. Algoritma propagasi balik menggunakan error output untuk mengubah nilai-nilai bobotnya dalam perambatan error output untuk mengubah nilai bobot-bobotnya dalam perambatan mundur (*backward*). Untuk mendapatkan error ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu (Puspitaningrum, 2006).

### 2.4.1 Algoritma Pelatihan pada Propagasi Balik

Algoritma pelatihan pada propagasi balik dapat dibedakan menjadi 2 (dua) jenis berdasarkan kecepatan proses latihannya, yaitu : algoritma pelatihan sederhana dan algoritma pelatihan yang lebih cepat. Prinsip dasar dari algoritma propagasi balik sederhana adalah memperbaiki bobot-bobot jaringan dengan arah yang membuat fungsi kinerja menjadi turun dengan cepat, namun memiliki kelemahan yakni proses pelatihannya biasanya akan berjalan cukup lambat, sehingga diperbaiki dengan algoritma pelatihan yang lebih cepat dengan 2 (dua) alternatif, yaitu : dengan menggunakan teknik heuristik dan dengan menggunakan teknis optimasi numeris (Kusumadewi & Purnomo, 2004)

#### 1. Perbaikan dengan Teknik Heuristik

Teknik ini merupakan pengembangan dari suatu analisis kinerja pada algoritma *steepest (gradient) descent standard*. Ada 3 (tiga) algoritma dengan teknik ini, yakni :

a. *Gradient Descent* dengan *Adaptive Learning Rate*

Pada fungsi ini, selama proses pembelajaran, *learning rate* akan terus bernilai konstan karena apabila *learning rate* terlalu tinggi maka algoritma menjadi tidak stabil dan jika terlalu rendah algoritma akan sangat lama dalam mencapai kekonvergenan.

b. *Gradient Descent* dengan Momentum dan *Adaptive Learning Rate*

Fungsi ini akan memperbaiki bobot-bobot berdasarkan *gradient descent* dengan *learning rate* yang bersifat adaptif seperti training data tapi juga dengan menggunakan momentum

c. *Resilient Backpropagation*

Algoritma pelatihan ini menggunakan fungsi aktivasi sigmoid yang membawa *input* dengan *range* yang tak terbatas ke nilai *output* dengan *range* yang terbatas, yaitu antara 0 sampai 1. Algoritma ini berusaha mengeliminasi besarnya efek dari turunan parsial dengan cara hanya menggunakan turunannya saja dan mengabaikan besarnya nilai turunan.

2. Perbaikan dengan Teknik Optimasi Numeris

Teknik ini terbagi menjadi 2 macam, yaitu :

a. Algoritma *Conjugate Gradient*

Pada algoritma ini pengaturan bobot tidak selalu dilakukan dalam arah turun seperti pada metode *gradient descent*, tapi menggunakan *conjugate gradient* dimana pengaturan bobot tidak selalu dengan arah menurun tapi disesuaikan dengan arah konjugasinya. Algoritma ini memanfaatkan fungsi *line search* untuk menempatkan sebuah titik minimum. Dari 4 (empat) algoritma *Conjugate Gradient*, tiga diantaranya melakukan proses *line search* secara terus menerus selama iterasi, yaitu : *Fletcher-Reeves Update*, *Polak-Ribiere*, dan *Powell-Beale Restarts*. Proses ini memakan waktu yang cukup lama untuk jumlah data yang besar dan iterasi yang besar pula, sehingga algoritma keempat, yaitu algoritma *scaled conjugate gradient* mencoba memperbaiki hal kekurangan tersebut.

## b. Algoritma Quasi Newton

Metode Newton merupakan salah satu alternatif *conjugate gradient* yang bisa mendapatkan nilai optimum lebih cepat. Metode Newton ini memang berjalan lebih cepat, namun metode ini sangat kompleks, memerlukan waktu dan memori yang cukup besar karena pada setiap iterasinya harus menghitung turunan kedua, perbaikan dari metode ini dikenal dengan nama metode Quasi-Newton atau metode Secant. Terdapat 2 (dua) alternatif algoritma dalam metode ini, yaitu :

(1) Algoritma *one step secant* yang menjembatani antara metode Quasi-Newton dengan *Gradient Conjugate*, dimana algoritma ini tidak menyimpan matriks Hessian secara lengkap dengan asumsi bahwa pada setiap iterasi matriks Hessian sebelumnya merupakan matriks identitas sehingga pencarian arah baru dapat dihitung tanpa harus menghitung invers matriks.

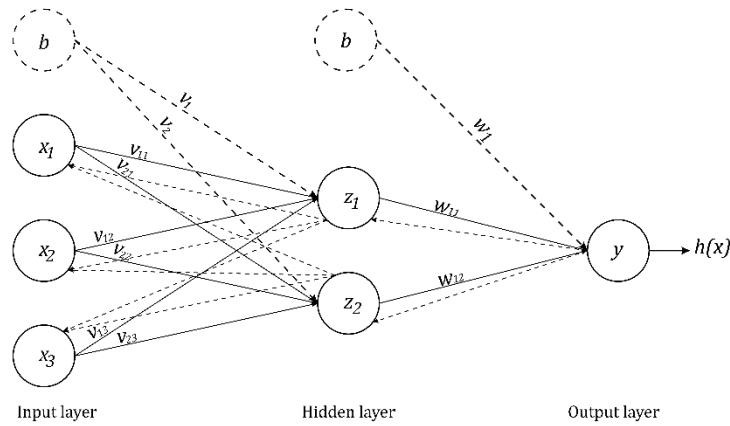
### (2) Algoritma Levenbarg-Marquardt

Metode ini dirancang dengan menggunakan turunan kedua tanpa harus menghitung matriks Hessian, melainkan matriks Jacobian yang dapat dihitung dengan teknik propagasi balik standar yang tentu saja lebih sederhana dibanding dengan menghitung matriks Hessian.

Algoritma *backpropagation* biasanya digunakan oleh *perceptron* dengan banyak lapisan untuk mengubah bobot – bobot yang terhubung dengan neuron – neuron yang ada pada lapisan tersembunyinya. Algoritma *backpropagation* menggunakan *error output* untuk mengubah nilai bobot – bobotnya dalam arah mundur (*backward*). Untuk mendapatkan error tersebut, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu. (Fausett, 1994)

Pada dasarnya pelatihan dengan metode *backpropagation* terdiri atas tiga langkah, yaitu sebagai berikut :

- a. Data dimasukkan ke input jaringan (*feedforward*)
- b. Perhitungan dan propagasi balik dari error yang bersangkutan
- c. Pembaharuan (*adjustment*) bobot.



Gambar 2.6 Arsitektur *Backpropagation*

Jika nilai error yang dihasilkan lebih besar dari batas error yang digunakan dalam sistem, maka akan dilakukan koreksi bobot. Koreksi bobot dapat dilakukan dengan menambah atau menurunkan nilai bobot.

Jika sinyal keluaran terlalu besar dari target yang ditentukan maka bobotnya diturunkan, sebaliknya jika sinyal keluaran terlalu kecil dari target yang ditentukan maka bobotnya dinaikkan. Koreksi bobot akan dilakukan sampai selisih target dan sinyal keluaran sekecil mungkin atau sama dengan batas error. Untuk melakukan koreksi bobot akan dilakukan penelusuran ke belakang seperti ditunjukkan dengan tanda panah mundur.

Adapun cara kerja dari *Backpropagation* :

1. Tiap – tiap output menerima target pola yang berhubungan dengan input pembelajaran, hitung informasi errornya.

$$\delta_k = \hat{y} - y \quad (2.2)$$

Kemudian hitung koreksi bobot (yang akan digunakan untuk memperbaiki nilai bobot  $w_{jk}$ ) dengan laju pembelajaran  $\alpha$ .

$$\Delta w_{jk} = \alpha \delta_k Z_j, (k = 1,2, \dots, m \text{ dan } j = 0,1,2, \dots, p) \quad (2.3)$$

2. Tiap – tiap unit tersembunyi ( $Z_j, j = 1,2,3, \dots, p$ ) menjumlahkan delta inputnya (dari unit – unit yang ada pada lapisan dibawahnya)

Hitung informasi errornya, dimana  $\theta^{(l)T}$  adalah matriks bobot yang ada di hidden ke layer yang ada di belakangnya dan  $\delta^{(l+1)}$  adalah hidden layer ke  $i+1$  dari arah belakang kemudian dikalikan dengan fungsi aktivasi.

$$\delta_j = \theta^{(l)T} \delta^{(l+1)} * (z_{in_j}) \quad (2.4)$$

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai  $v_{ij}$  dengan laju pembelajaran  $\alpha$ ).

$$\Delta v_{ij} = \alpha \delta_j x_i, (j = 1, 2, \dots, p \text{ dan } i = 0, 1, 2, \dots, n) \quad (2.5)$$

3. Hitung semua perubahan bobot

a. Perubahan bobot garis yang menuju ke unit keluaran

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (2.6)$$

b. Perubahan bobot garis yang menuju ke unit tersembunyi

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (2.7)$$

c. Proses iterasi selesai

Keterangan simbol :

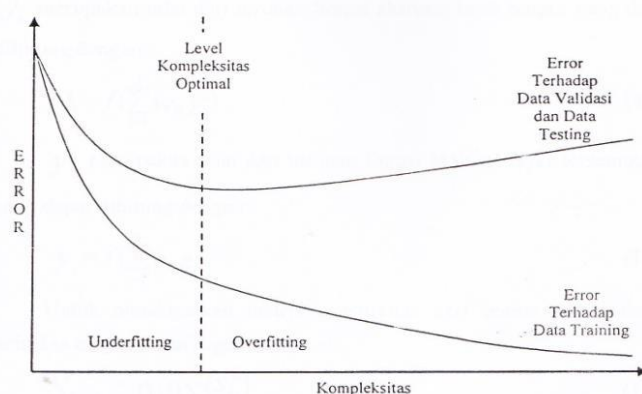
$\delta_k$  = Informasi tentang kesalahan pada unit  $Y_k$  yang disebarkan kembali ke unit tersembunyi

$\delta_j$  = Informasi tentang kesalahan dari lapisan output ke unit tersembunyi

$\alpha$  = Laju pembelajaran (*Lerning rate*)

#### 2.4.2 Akurasi dan Generalisasi

Gambar 2.7 menunjukkan akurasi dan generalisasi berkaitan dengan tingkat kompleksitas dari suatu *Neural Network* (NN). Peningkatan kompleksitas dari *Neural Network* meningkatkan akurasi dari *Neural Network* terhadap data penelitian, tetapi peningkatan akurasi dan kompleksitas ini dapat menurunkan tingkat generalisasi *Neural Network* pada data validasi dan data pengujian (Larose, 2005)



Gambar 2.7 Akurasi dan Generalisasi (Larose, 2005)

## 2.5 Confusion Matrix

Di dalam *machine learning*, mengukur kinerja atau performa dari suatu model adalah hal yang esensial. Model yang diperoleh dari pelatihan melalui data training perlu diuji melalui data testing. Kinerja diukur berdasarkan seberapa baik model tersebut memprediksi dengan benar data yang ada.

Pada klasifikasi biner, kelas positif yang berhasil diprediksi dengan benar disebut *true positive*, jika kelas positif tersebut diprediksi negatif (salah) disebut *false negative*. Kelas negatif yang berhasil diprediksi negatif (benar) disebut *true negative*, dan kelas negatif yang diprediksi positif disebut *false positive*. Jumlah dari kasus-kasus tersebut direpresentasikan dalam suatu tabel kontingensi yang disebut *confusion matrix* (Swets, 1988)

Tabel 2.2 *Confusion Matrix*

<i>Confusion Matrix</i>		Prediksi	
		Negatif	Positif
Aktual	Negatif	<i>True Negative</i>	<i>False Positive</i>
	Positif	<i>False Negative</i>	<i>True Positive</i>

Akurasi adalah ukuran kinerja yang menunjukkan seberapa baik suatu pengklasifikasi dalam mengklasifikasikan seluruh data. Akurasi adalah rasio antara observasi yang diklasifikasikan secara benar dengan total observasi:

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \quad (0.8)$$

Presisi adalah ukuran kinerja yang menunjukkan seberapa besar kebenaran suatu pengklasifikasi dari seluruh kelas positif yang diprediksi. Presisi adalah rasio antara jumlah kelas positif yang diklasifikasikan secara benar dengan jumlah observasi yang diklasifikasikan positif:

$$Precision = \frac{TP}{(TP + FP)} \quad (0.9)$$

*Recall* atau sensitivitas adalah ukuran kinerja yang menunjukkan seberapa baik suatu pengklasifikasi dalam mengklasifikasikan kelas positif. *Recall* adalah rasio antara jumlah observasi positif yang diklasifikasikan secara benar dengan jumlah observasi positif asli:



$$Recall = \frac{TP}{(TP + FN)} \quad (0.10)$$

F1-Score adalah *harmonic mean* antara *precision* dan *recall*:

$$F1 = 2 * \frac{Precision * Recall}{(Precision + Recall)} \quad (0.11)$$

## 2.6 Synthetic Minority Oversampling Technique

*Synthetic Minority Oversampling Technique* (SMOTE) diperkenalkan oleh (Chawla VN, 2002) sebagai salah satu cara dalam menangani pengaruh dari sedikitnya informasi mengenai kelas minoritas dalam suatu gugus data. SMOTE merupakan algoritma dengan pendekatan *oversampling* yaitu menambah jumlah data pengamatan pada kelas minoritas. Ilustrasi data tidak seimbang dengan membuat data buatan atau sintetis yang diyakini lebih baik dibandingkan dengan cara mereplikasi data yang ada sehingga setara dengan kelas mayoritas (Chawla VN, 2002)

Teknik SMOTE merupakan salah satu metode penanganan data tidak seimbang dengan membangkitkan data buatan untuk kelas data minoritas sehingga proporsi kelas data mayor dan minor menjadi lebih seimbang. Data buatan atau data sintetis tersebut dibuat berdasarkan *k*-tetangga terdekat (*k-nearest neighbor*). Seluruh peubah yang digunakan pada penelitian ini adalah variabel kategorik, sehingga perhitungan jarak antar contoh kelas minornya dilakukan dengan rumus *Value Difference Metric* (VDM) yaitu (Cost S, 1993)

1. Hitung jarak antara dua amatan yang berskala nominal menggunakan MVDM dengan formula sebagai berikut (Cost S, 1993) :

$$\Delta(x, y) = w_x w_y \sum_{i=1}^N \delta(v_{1i}, v_{2i}) \quad (2.12)$$

dimana ;

$\Delta(x, y)$  : jarak antara amatan  $x$  dan  $y$

$w_x$  : bobot amatan  $x$  (dapat diabaikan)

$w_y$  : bobot amatan  $y$  (dapat diabaikan)

$N$  : banyaknya variabel penjelas

$\delta(v_{1i}v_{2i})$  : jarak antara amatan  $x$  dan  $y$  pada variabel ke- $i$  dengan perhitungan jarak antar amatan  $x$  dan  $y$  pada variabel ke- $i$  dilakukan melalui formula VDM (Cost S, 1993):

$$\delta(v_{1i}, v_{2i}) = \sum_{j=1}^S \left| \frac{C_{1j}}{C_1} - \frac{C_{2j}}{C_2} \right|^k \quad (2.13)$$

dimana :

$S$  : banyaknya kelas pada variabel respon

$C_{xj}$  : banyaknya kategori  $x$  pada kelas ke- $j$

$C_{yj}$  : banyaknya kategori  $y$  pada kelas ke- $j$

$C_x$  : banyaknya kategori  $x$  terjadi

$C_y$  : banyaknya kategori  $y$  terjadi

$k$  : konstanta (biasanya bernilai 1)

2. Pilih kategori mayoritas antara amatan kelas minoritas dengan  $k$ -tetangga terdekatnya untuk nilai yang bersifat nominal. Jika terjadi nilai yang sama maka pilih secara acak.
3. Nilai yang terpilih tersebut merupakan amatan yang baru.

## 2.7 Python

### 2.7.1 Flask

Flask adalah sebuah *web framework* yang ditulis dengan bahasa Python dan tergolong sebagai jenis *microframework*. Flask berfungsi sebagai kerangka kerja aplikasi dan tampilan dari suatu *web*. Dengan menggunakan Flask dan bahasa Python, pengembang dapat membuat sebuah *web* yang terstruktur dan dapat mengatur *behaviour* suatu *web* dengan lebih mudah.

Flask termasuk pada jenis *microframework* karena tidak memerlukan suatu alat atau pustaka tertentu dalam penggunaannya. Sebagian besar fungsi dan komponen umum seperti validasi form, *database*, dan sebagainya tidak terpasang secara *default* di Flask. Dengan begitu, fleksibilitas serta skalabilitas dari Flask dapat dikatakan cukup tinggi dibandingkan dengan *framework* lainnya. (Irsyad, 2018)

### 2.7.2 *Instalasi Flask*

Dalam melakukan instalasi Flask pada sebuah perangkat, dibutuhkan PIP yang biasanya sudah terinstall pada Python versi 3.4 ke atas. PIP adalah sebuah *package management system* yang biasa digunakan untuk mengatur dan menginstall *package* yang berisi modul – modul Python. PIP digunakan untuk menginstall Flask karena Flask ditulis dan dikembangkan dengan bahasa dan modul – modul pemrograman Python. Dengan menggunakan PIP, semua hal yang dibutuhkan untuk instalasi Flask akan diunduh dan dipasang dalam satu perintah.

Sebelum melakukan instalasi, pengguna Flask sebaiknya menggunakan *virtual environment* dalam proses pengembangan web. *Virtual environment* adalah sebuah sistem untuk mengisolasi sebuah proyek Python agar tidak terjadi konflik dengan proyek Python lainnya. Dengan *virtual environment*, sebuah proyek Python atau Flask pada kasus ini, seakan ada pada satu komputer sendiri dengan dependensi sendiri dan tidak berhubungan dengan proyek Python lainnya. Untuk menginstall *virtual environment* perintah yang digunakan adalah sebagai berikut.

```
$ pip install virtualenv
```

Setelah mengaktifkan *virtual environment*, Flask dapat diinstall tanpa perlu mengkhawatirkan adanya konflik dependensi dengan proyek Python lainnya. Untuk menginstall Flask perintah yang digunakan adalah

```
$ pip install Flask
```

### 2.7.3 *Fitur – fitur Flask*

Fitur – fitur bawaan dari Flask diantaranya adalah

- *Built-in development server*
- *Debugger* cepat
- *Integrated support* untuk pengetesan unit
- Kompatibel dengan mesin aplikasi Google
- *RESTful request dispatching*
- *Jinja2 templating*

- Mendukung *secure cookies*
- Berbasis *unicode*
- Mengikuti WSGI 1.0

Selain itu, Flask juga didukung dengan dokumentasi yang sangat baik dan banyak forum yang ada di internet untuk diskusi terkait masalah Flask. (Irsyad, 2018)