

**TUGAS AKHIR - CM224835**

**PENERAPAN ALGORITMA *DENSE NEURAL NETWORK*  
UNTUK KLASIFIKASI JENIS SEDIMEN DASAR LAUT  
MENGUNAKAN DATA MOSAIK DAN POLA FITUR  
*BACKSCATTER***

**HESSI CANDRA HARISA**

**NRP 03311940000014**

Dosen Pembimbing

**Khomsin, S.T., M.T.**

**NIP 197507052000121001**

**Dr. Muhammad Aldila Syariz, S.T., M.S., Ph.D.**

**NIP 2022199211023**

**Program Studi S1 Teknik Geomatika**

Departemen Teknik Geomatika

Fakultas Teknik Sipil, Perencanaan, dan Kebumihan

Institut Teknologi Sepuluh Nopember

Surabaya

2023





**TUGAS AKHIR - CM224835**

**PENERAPAN ALGORITMA *DENSE NEURAL NETWORK* UNTUK KLASIFIKASI JENIS SEDIMEN DASAR LAUT MENGGUNAKAN DATA MOSAIK DAN POLA FITUR *BACKSCATTER***

**HESSI CANDRA HARISA**

**NRP 03311940000014**

Dosen Pembimbing

**Khomsin, S.T., M.T.**

**NIP 197507052000121001**

**Dr. Muhammad Aldila Syariz, S.T., M.S., Ph.D.**

**NIP 2022199211023**

**Program Studi S1 Teknik Geomatika**

Departemen Teknik Geomatika

Fakultas Teknik Sipil, Perencanaan, dan Kebumihan

Institut Teknologi Sepuluh Nopember

Surabaya

2023





**FINAL ASSIGNMENT - CM224835**

**APPLICATION OF DENSE NEURAL NETWORK  
ALGORITHM FOR CLASSIFICATION OF SEAFLOOR  
SEDIMENT TYPES USING MOSAIC AND  
BACKSCATTER TEXTURE FEATURE**

**HESSI CANDRA HARISA**

**NRP 03311940000014**

Advisor

**Khomsin, S.T., M.T.**

**NIP 197507052000121001**

**Dr. Muhammad Aldila Syariz, S.T., M.S., Ph.D**

**NIP 2022199211023**

**Study Program Bachelor of Geomatics Engineering**

Geomatics Engineering Department

Faculty of Civil, Planning, and Geo Engineering

Institut Teknologi Sepuluh Nopember

Surabaya

2023



## LEMBAR PENGESAHAN

### PENERAPAN ALGORITMA *DENSE NEURAL NETWORK* UNTUK KLASIFIKASI JENIS SEDIMEN DASAR LAUT MENGGUNAKAN DATA MOSAIK DAN POLA FITUR *BACKSCATTER*

#### TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat  
memperoleh gelar Sarjana Teknik pada  
Program Studi S-1 Sarjana  
Departemen Teknik Geomatika  
Fakultas Teknik Sipil, Perencanaan, dan Kebumihan  
Institut Teknologi Sepuluh Nopember

Oleh : **HESSI CANDRA HARISA**  
NRP. 0331194000014

Disetujui oleh Tim Penguji Tugas Akhir

1. Khomsin, S.T., M.T.  
NIP. 197507052000121001

Pembimbing

2. Dr. Muhammad Aldila Syariz, S.T., M.S., Ph.D.  
NIP. 2022199211023

Ko-pembimbing

3. Danar Guruh Pratomo, S.T, M.T., Ph.D.  
NIP. 198005072003121001

Penguji

4. Irena Hana Hariyanto, S.T., M.T.  
NIP. 2022199812024

Penguji

5. Dr-Ing. Noorlaila Hayati, S.T., M.T.  
NIP. 199006032014042001

Penguji

**SURABAYA**

**JULI 2023**

***“Halaman ini sengaja dikosongkan”***

## APPROVAL SHEET

### APPLICATION OF DENSE NEURAL NETWORK ALGORITHM FOR CLASSIFICATION OF SEAFLOOR SEDIMENT TYPES USING MOSAIC AND BACKSCATTER TEXTURE FEATURE

#### FINAL PROJECT REPORT

Submitted to fulfill one of the requirements  
for obtaining a degree bachelor of engineering at  
Undergraduate Study Program  
Department of Geomatics Engineering  
Faculty of Civil, Planning, and Geo Engineering  
Institut Teknologi Sepuluh Nopember

By : HESSI CANDRA HARISA  
NRP. 03311940000014

Approved by Final Project Examiner Team:

1. Khomsin, S.T., M.T.  
NIP. 197507052000121001

Advisor



2. Dr. Muhammad Aldila Syariz, S.T., M.S., Ph.D.  
NIP. 2022199211023

Co-advisor



3. Danar Guruh Pratomo, S.T, M.T., Ph.D.  
NIP. 198005072003121001

Examiner



4. Irena Hana Hariyanto, S.T., M.T.  
NIP. 2022199812024

Examiner



5. Dr-Ing. Noorlaila Hayati, S.T., M.T.  
NIP. 199006032014042001

Examiner



SURABAYA  
JULY 2023

***“Halaman ini sengaja dikosongkan”***

## PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama mahasiswa / NRP : Hessi Candra Harisa / 0331194000014

Departemen : Teknik Geomatika

Dosen Pembimbing / NIP :

1. Khomsin, S.T., M.T. / 19750705200021001
2. Dr. Muhammad Aldila Syariz, S.T., M.S., Ph.D. / 2022199211023

dengan ini menyatakan bahwa Tugas Akhir dengan judul “Penerapan Algoritma *Dense Neural Network* untuk Klasifikasi Jenis Sedimen Dasar Laut Menggunakan Data Mosaik dan Pola Fitur *Backscatter*” adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, 31 Juli 2023

Dosen Pembimbing 1

Mengetahui

Dosen Pembimbing 2



Khomsin, S.T., M.T.  
NIP. 197507052000121001



Dr. M Aldila Syariz, S.T., M.S., Ph.D.  
NIP. 2022199211023

Mahasiswa



Hessi Candra Harisa  
NRP. 0331194000014

***“Halaman ini sengaja dikosongkan”***



***“Halaman ini sengaja dikosongkan”***

# **PENERAPAN ALGORITMA DENSE NEURAL NETWORK UNTUK KLASIFIKASI JENIS SEDIMEN DASAR LAUT MENGGUNAKAN DATA MOSAIK DAN POLA FITUR BACKSCATTER**

**Nama Mahasiswa / NRP** : Hessi Candra Harisa / 0331194000014  
**Departemen** : Teknik Geomatika  
**Dosen Pembimbing I** : Khomsin, S.T., M.T.  
**Dosen Pembimbing II** : Dr. Muhammad Aldila Syariz, S.T., M.S., Ph.D.

## **Abstrak**

Pintu gerbang untuk pergerakan impor dan ekspor barang saat ini adalah pelabuhan, di mana transportasi laut mendominasi 80% pangsa pasar dunia. Pelabuhan Benoa, yang terletak di Desa Pedungan, Kecamatan Denpasar Selatan, Kota Denpasar, Bali, merupakan salah satu pelabuhan yang sibuk. Untuk memastikan keselamatan dan kenyamanan dalam penggunaan transportasi laut, penting untuk memperoleh informasi tentang kondisi dasar perairan. Salah satu cara untuk memperoleh informasi tersebut adalah dengan melakukan survei hidrografi menggunakan *Multibeam Echosounder* yang menghasilkan data *backscatter* sehingga dapat diolah menjadi mosaik *backscatter*, yang kemudian dapat digunakan untuk mengklasifikasikan jenis sedimen dasar laut. Dalam penelitian ini, digunakan empat pola fitur atau tekstur dasar perairan, yaitu *Angular Second Moment (ASM)*, *Energi (Energy)*, *Kontras (Contrast)*, dan *Korelasi (Correlation)*. Penelitian ini bertujuan untuk menentukan desain yang tepat untuk metode *Dense Neural Network* sehingga menghasilkan akurasi yang baik dan juga untuk mengetahui klasifikasi jenis sedimen dan persebarannya dari hasil pengolahan data mosaik dan pola fitur *backscatter* menggunakan metode *Dense Neural Network* di Pelabuhan Benoa. Terdapat 12 titik sampel sedimen yang dibagi menjadi 9 titik untuk data *training* dan 3 titik untuk data *testing*. Pemodelan DNN dengan nilai akurasi paling tinggi dimiliki oleh Model 5 dimana nilai akurasi untuk *training* sebesar 88% dan akurasi untuk *testing* sebesar 100%. Hasil klasifikasi menggunakan DNN menghasilkan 6 jenis sedimen yaitu pasir berlanau berkerikil karang, lanau berpasir, pasir berlanau, kerikil karang berpasir berlanau, lanau kelepungan lunak, dan pasir berkerikil. Persentase paling banyak dimiliki oleh jenis sedimen lanau berpasir sebesar 49,30% dan persentase paling sedikit dimiliki oleh lanau berkelepungan lunak sebesar 0,53%.

**Kata kunci:** DNN, Pola Fitur, Jenis Sedimen, Klasifikasi, Mosaik *Backscatter*

***“Halaman ini sengaja dikosongkan”***

# **APPLICATION OF DENSE NEURAL NETWORK ALGORITHM FOR CLASSIFICATION OF SEAFLOOR SEDIMENT TYPES USING MOSAIC AND BACKSCATTER TEXTURE FEATURE**

**Student Name / NRR** : Hessi Candra Harisa / 03311940000014  
**Department** : Geomatics Engineering  
**Advisor I** : Khomsin, S.T., M.T.  
**Advisor II** : Dr. Muhammad Aldila Syariz, S.T., M.S., Ph.D.

## **Abstract**

The gateway for import and export movements of goods today is the port, where maritime transportation dominates 80% of the world market share. Benoa Port, located in Pedungan Village, South Denpasar District, Denpasar City, Bali, is one of the busy ports. To ensure safety and comfort in maritime transportation, it is crucial to obtain information about the seabed conditions. One way to acquire such information is by conducting hydrographic surveys using Multibeam Echosounder, which generates backscatter data that can be processed into backscatter mosaics, subsequently used to classify the types of seabed sediments. In this study, four basic water feature or texture patterns are utilized: Angular Second Moment (ASM), Energy, Contrast, and Correlation. This research aims to determine the appropriate design for the Dense Neural Network method to achieve good accuracy and to classify the types and distribution of sediments from the processed backscatter data mosaic and feature patterns using the Dense Neural Network method at Benoa Port. There are 12 sediment sample points, divided into 9 points for training data and 3 points for testing data. The DNN modeling with the highest accuracy is Model 5, with a training accuracy of 88% and testing accuracy of 100%. The classification results using DNN yield 6 types of sediments: sandy silt with gravel corals, sandy silt, silty sand, sandy gravel with sandy silt, soft clayey silt, and gravelly sand. The most abundant sediment type is sandy silt with a percentage of 49.30%, while the least abundant is soft clayey silt with a percentage of 0.53%.

**Key word:** *Backscatter Mosaic, Classification, DNN, Sediment Type, Texture Feature*

***“Halaman ini sengaja dikosongkan”***

## KATA PENGANTAR

Puji syukur atas kehadiran Allah SWT, karena atas rahmat dan karunia-Nya penulis dapat menyelesaikan Tugas Akhir dengan judul “Penerapan Algoritma *Dense Neural Network* untuk Klasifikasi Jenis Sedimen Dasar Laut Menggunakan Data Mosaik dan Pola Fitur *Backscatter*”.

Dalam penyusunan laporan ini banyak pihak yang telah memberikan bantuan dan dukungan kepada penulis. Oleh karena itu, tidak lupa penulis menyampaikan ucapan terima kasih kepada:

1. Orang tua Bapak Eko Purnomo, Ibu Hariyati, Kakak Agnes Candra Dewi, Kakak Kuncara Nata Waskita dan Keponakan Kainayya Candra Waskita serta keluarga besar penulis yang tidak dapat disebutkan satu persatu yang selalu memberi dukungan dan masukan selama masa perkuliahan serta saat pelaksanaan Tugas Akhir.
2. Bapak Dinar Guruh Pratomo, S.T., M.T., Ph.D. selaku Ketua Departemen Teknik Geomatika yang sudah memberikan izin untuk melaksanakan tugas akhir ini.
3. Bapak Khomsin, S.T., M.T. dan Bapak Dr. Muhammad Aldila Syariz, S.T., M.S., Ph.D. selaku dosen pembimbing yang telah memberikan bimbingan hingga Tugas Akhir ini dapat terselesaikan dengan baik.
4. Seluruh dosen pengajar dan staf akademik Departemen Teknik Geomatika yang telah membantu dan memberi pelajaran akademik maupun non akademik selama masa perkuliahan.
5. PT. Pelabuhan Indonesia (Persero) Regional 3 yang telah menyediakan data *Multibeam Echosounder* dan data sampel sedimen.
6. Teman-teman penulis yang telah membantu dalam proses pengerjaan Tugas Akhir ini serta memberikan dukungan secara moril kepada penulis;
7. Teman-teman angkatan 2019 terkhusus teman-teman Laboratorium Geomarin yang telah menjadi keluarga serta bagian dari hidup penulis selama berkuliah di Teknik Geomatika yang selalu memberikan warna baru, dukungan, bantuan serta masukkan kepada penulis hingga saat ini.
8. Semua pihak yang tidak bisa disebutkan satu per satu, yang telah membantu dan memberi dukungan kepada penulis dalam menyelesaikan laporan Tugas Akhir.

Penulis menyadari bahwa dalam penulisan laporan ini tidak terlepas dari kekurangan. Akhir kata, penulis berharap semoga laporan ini dapat bermanfaat bagi pembaca.

Surabaya, Juli 2023

Penulis

*“Halaman ini sengaja dikosongkan”*

## DAFTAR ISI

LEMBAR PENGESAHAN .....	i
APPROVAL SHEET .....	iii
PERNYATAAN ORISINALITAS .....	v
STATEMENT OF ORIGINALITY .....	vii
KATA PENGANTAR .....	xiii
DAFTAR ISI .....	xv
DAFTAR GAMBAR .....	xvii
DAFTAR TABEL .....	xix
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Permasalahan .....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan .....	3
1.5 Manfaat .....	3
BAB II TINJAUAN PUSTAKA .....	5
2.1 Penelitian Terdahulu .....	5
2.2 Survei Hidrografi .....	6
2.3 Sedimen Dasar Laut .....	8
2.4 <i>Multibeam Echosounder</i> .....	12
2.4.1 Definisi .....	12
2.4.2 Kalibrasi <i>Multibeam Echosounder</i> .....	13
2.5 Mosaik <i>Backscatter</i> .....	18
2.6 Pola Fitur Mosaik <i>Backscatter</i> .....	19
2.6.1 <i>Angular Second Moment (ASM)</i> .....	20
2.6.2 Kontras ( <i>Contrast</i> ) .....	20
2.6.3 Korelasi ( <i>Correlation</i> ) .....	21
2.6.4 Energi ( <i>Energy</i> ) .....	21
2.7 <i>Dense Neural Network</i> .....	22
2.7.1 <i>Data Training</i> dan <i>Data Testing</i> .....	26
2.7.2 <i>Activation Function</i> .....	27
2.7.3 <i>Hyperparameter</i> .....	29
2.8 Pasang Surut Air Laut .....	29
2.9 <i>Sound Velocity Profiler</i> .....	32
2.10 <i>Global Positioning System</i> .....	32
2.11 EIVA .....	33
2.12 FMGT .....	34
2.13 SNAP .....	35
2.14 ArcGIS .....	35
2.15 Google Colab .....	36
BAB III METODOLOGI PENELITIAN .....	39
3.1 Lokasi Penelitian .....	39
3.2 Data dan Peralatan .....	39
3.2.1 Data .....	39
3.2.2 Peralatan .....	39
3.3 Metodologi Penelitian .....	40
3.3.1 Tahapan Pelaksanaan Penelitian .....	40

3.3.2 Tahapan Pengolahan <i>Raw Data</i> .....	42
3.3.3 Tahapan Ekstraksi Fitur Mosaik <i>Backscatter</i> .....	46
3.3.4 Tahapan Desain Model dan Klasifikasi dengan DNN.....	54
BAB IV HASIL DAN PEMBAHASAN.....	59
4.1 Model <i>Dense Neural Network</i> .....	59
4.2 Hasil Evaluasi Model .....	62
4.3 Klasifikasi dan Persebaran Sedimen Dasar Laut.....	74
BAB V KESIMPULAN DAN SARAN .....	81
5.1 Kesimpulan.....	81
5.2 Saran.....	81
DAFTAR PUSTAKA.....	83
LAMPIRAN .....	89
BIODATA PENULIS.....	115

## DAFTAR GAMBAR

Gambar 2.1 Survei Hidrografi .....	7
Gambar 2.2 Akuisisi Data Survei Hidrografi .....	8
Gambar 2.3 Contoh Sedimen Dasar Laut .....	9
Gambar 2.4 Van Veen Grab .....	10
Gambar 2.5 <i>Gravity Corer</i> .....	11
Gambar 2.6 Pengukuran MBES .....	13
Gambar 2.7 Sistem Koordinat Kapal dan Perputaran.....	14
Gambar 2.8 Latensi atau <i>Time Delay</i> .....	15
Gambar 2.9 Kalibrasi <i>Roll</i> .....	16
Gambar 2.10 Hasil Kalibrasi <i>Roll</i> Sebelum dan Sesudah .....	16
Gambar 2.11 Kalibrasi <i>Pitch</i> .....	16
Gambar 2.12 Hasil Kalibrasi <i>Pitch</i> Sebelum dan Sesudah.....	17
Gambar 2.13 Kalibrasi <i>Yaw</i> .....	17
Gambar 2.14 Hasil Kalibrasi <i>Yaw</i> Sebelum dan Sesudah .....	17
Gambar 2.15 Mosaik <i>Backscatter</i> .....	18
Gambar 2.16 Urutan Hierarki DNN .....	22
Gambar 2.17 Penerapan <i>Artificial Intelligence</i> .....	23
Gambar 2.18 Jenis <i>Machine Learning</i> .....	23
Gambar 2.19 Aplikasi <i>Machine Learning</i> .....	24
Gambar 2.20 Bentuk <i>Dense Neural Network</i> .....	26
Gambar 2.21 Pembagian Data <i>Training</i> dan Data <i>Testing</i> .....	27
Gambar 2.22 Fungsi Aktivasi Sigmoid .....	28
Gambar 2.23 Fungsi Aktivasi ReLU .....	29
Gambar 2.24 Pola Gerak Pasut Harian Ganda .....	30
Gambar 2.25 Pola Gerak Pasut Harian Tunggal.....	30
Gambar 2.26 Pola Gerak Pasut Campuran Condong Harian Ganda .....	31
Gambar 2.27 Pola Gerak Pasut Campuran Condong Harian Tunggal .....	31
Gambar 2.28 <i>Sound Velocity Profiler</i> .....	32
Gambar 2.29 Penentuan Posisi dengan GNSS .....	33
Gambar 3.1 Peta Lokasi Penelitian.....	39
Gambar 3.2 Diagram Alir Tahapan Pelaksanaan Penelitian .....	40
Gambar 3.3 Diagram Alir Tahapan Pengolahan <i>Raw Data</i> .....	42
Gambar 3.4 Mosaik <i>Backscatter</i> .....	45
Gambar 3.5 Diagram Alir Tahapan Ekstraksi Fitur Mosaik <i>Backscatter</i> .....	46
Gambar 3.6 Hasil Ekstraksi Fitur Frekuensi 170 kHz.....	47
Gambar 3.7 Hasil Ekstraksi Fitur Frekuensi 220 kHz.....	48
Gambar 3.8 Hasil Ekstraksi Fitur Frekuensi 270 kHz.....	49
Gambar 3.9 Hasil Ekstraksi Fitur Frekuensi 350 kHz.....	50
Gambar 3.10 Hasil Ekstraksi Fitur Frekuensi 400 kHz.....	51
Gambar 3.11 Diagram Alir Desain Model dan Klasifikasi dengan DNN .....	54
Gambar 3.12 Peta Lokasi Titik Sedimen.....	55
Gambar 4.1 Model 1.....	59
Gambar 4.2 Model 2.....	59
Gambar 4.3 Model 3.....	60
Gambar 4.4 Model 4.....	60
Gambar 4.5 Model 5.....	60
Gambar 4.6 Model 6.....	60

Gambar 4.7 Grafik Hasil Akurasi Model 1 .....	63
Gambar 4.8 Grafik Hasil Akurasi Model 2.....	64
Gambar 4.9 Grafik Hasil Akurasi Model 3.....	65
Gambar 4.10 Grafik Hasil Akurasi Model 4.....	65
Gambar 4.11 Grafik Hasil Akurasi Model 5.....	66
Gambar 4.12 Grafik Hasil Akurasi Model 6.....	67
Gambar 4.13 Grafik Hasil <i>Loss</i> Model 1 .....	68
Gambar 4.14 Grafik Hasil <i>Loss</i> Model 2 .....	69
Gambar 4.15 Grafik Hasil <i>Loss</i> Model 3 .....	70
Gambar 4.16 Grafik Hasil <i>Loss</i> Model 4 .....	71
Gambar 4.17 Grafik Hasil <i>Loss</i> Model 5 .....	72
Gambar 4.18 Grafik Hasil <i>Loss</i> Model 6 .....	73
Gambar 4.19 Peta Sebaran Sedimen Dasar Laut Pelabuhan Benoa .....	75
Gambar 4.20 Titik Sampel Sedimen PT1 .....	76
Gambar 4.21 Titik Sampel Sedimen PT2 .....	76
Gambar 4.22 Titik Sampel Sedimen PT3 .....	76
Gambar 4.23 Titik Sampel Sedimen PT4 .....	76
Gambar 4.24 Titik Sampel Sedimen PT5 .....	77
Gambar 4.25 Titik Sampel Sedimen PT6 .....	77
Gambar 4.26 Titik Sampel Sedimen PT7 .....	77
Gambar 4.27 Titik Sampel Sedimen PT8 .....	77
Gambar 4.28 Titik Sampel Sedimen PT9 .....	77
Gambar 4.29 Titik Sampel Sedimen PT10 .....	77
Gambar 4.30 Titik Sampel Sedimen PT11 .....	78
Gambar 4.31 Titik Sampel Sedimen PT12 .....	78

## DAFTAR TABEL

Tabel 2.1 Skala Wentworth .....	10
Tabel 3.1 Nilai Intensitas Mosaik <i>Backscatter</i> .....	52
Tabel 3.2 Nilai Intensitas ASM .....	52
Tabel 3.3 Nilai Intensitas <i>Contrast</i> .....	53
Tabel 3.4 Nilai Intensitas <i>Correlation</i> .....	53
Tabel 3.5 Nilai Intensitas <i>Energy</i> .....	53
Tabel 3.5 Nilai Intensitas <i>Energy</i> (lanjutan) .....	54
Tabel 3.7 Data Sampel Sedimen .....	55
Tabel 3.8 Data <i>Training</i> .....	56
Tabel 3.9 Data <i>Testing</i> .....	56
Tabel 4.1 Hasil Akurasi Model DNN .....	62
Tabel 4.2 Hasil <i>Loss</i> Model DNN .....	68
Tabel 4.3 Perbandingan Jenis Sedimen .....	79

***“Halaman ini sengaja dikosongkan”***

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Sebagian besar wilayah di permukaan bumi merupakan perairan laut yang luasnya dua pertiga bagian bumi dan selebihnya merupakan daratan (Nugroho dkk., 2022). Luas laut di Indonesia yang juga mencakup dua pertiga bagian wilayah Indonesia memiliki banyak potensi di bidang kelautan, salah satunya dalam transportasi laut (Nikawanti & Aca, 2021). Laut menjadi lajur transportasi penting untuk lalu lintas penumpang dan pengiriman barang Indonesia yang menghubungkan antar pulau, negara bahkan benua. (Khomsin dkk., 2021). Posisi Indonesia yang sangat strategis yaitu diantara dua benua yaitu Benua Asia dan Benua Australia dan dua samudera yaitu Samudera Hindia dan Samudera Pasifik memberikan keuntungan terlebih dalam hal transportasi laut (Rustam, 2016). Pelabuhan menjadi pintu gerbang untuk pergerakan impor dan ekspor barang, dimana angkutan barang melalui transportasi laut menguasai 80% pangsa pasar dunia (Josep, 2019).

Dalam menjaga keselamatan serta memberikan kenyamanan saat menggunakan transportasi laut, diperlukan adanya informasi mengenai kondisi dasar perairan. Salah satu sarana untuk mengetahui informasi tersebut adalah dengan melakukan survei hidrografi (Nugroho dkk., 2022). Survei hidrografi adalah suatu ilmu yang mempelajari pengukuran serta penggambaran parameter-parameter untuk menjelaskan sifat-sifat dan konfigurasi dasar laut secara tepat, hubungan geografisnya dengan daratan, serta karakteristik-karakteristik dan dinamika-dinamika lautan (IHO, 2008). Tujuan dari kegiatan survei hidrografi adalah untuk pengukuran kedalaman suatu perairan dari permukaan air menggunakan gelombang akustik (Hovem, 2007). Pada survei hidrografi juga dikenal istilah survei batimetri. Survei batimetri adalah survei yang dilakukan untuk mengukur atau memetakan kedalaman di suatu perairan (IHO, 2008). Hasil dari kegiatan survei hidrografi dapat dimanfaatkan untuk beberapa hal seperti untuk pembuatan peta navigasi, untuk konstruksi laut, untuk eksplorasi dan pengeboran minyak lepas pantai, untuk mendeteksi kapal karam atau pipa bawah laut atau kabel bawah laut, untuk persebaran terumbu karang, dan juga untuk keperluan lainnya yang berhubungan dengan kondisi dasar perairan (Nugroho dkk., 2022).

Perkembangan teknologi survei hidrografi sekarang ini terus meningkat. Dapat dilihat dengan semakin beragamnya wahana atau instrumen yang digunakan untuk mengetahui kedalaman serta kondisi di dasar perairan. Salah satu wahana atau instrumen yang dapat digunakan untuk mengukur kedalaman adalah *Multibeam Echosounder* (MBES). Teknologi MBES merupakan instrumen yang paling efektif dan paling akurat saat ini jika dibandingkan dengan *Singlebeam Echosounder* (SBES) (Nugroho dkk., 2022). Resolusi yang didapatkan dari penggunaan MBES lebih tinggi jika dibandingkan dengan SBES (Khomsin dkk., 2021). MBES memanfaatkan gelombang akustik yang dipancarkan ke dasar laut melalui pemancar atau *transducer* dan merekam pantulan gelombang tersebut menggunakan penerima atau *receiver* (Nitriansyah & Cahyono, 2022). MBES memancarkan sinyal dalam satu pancaran, yang kemudian menghasilkan titik-titik kedalaman saat dipantulkan dan diterima oleh transduser, ketika titik-titik kedalaman tersebut dihubungkan, akan membentuk profil permukaan dasar laut. Dalam penentuan posisinya, penggunaan metode pengukuran gelombang hidrografi akustik menggunakan MBES ini digabungkan dengan sensor pengukuran posisi GNSS (*Global Navigation Satellite System*) sehingga diperoleh nilai dari kedalaman perairan dan posisi yang nantinya dapat dimanfaatkan menjadi beberapa analisis hidrografi yang lebih baik (Saroni & Basith, 2022). Pada saat pengukuran hidrografi menggunakan instrumen MBES selain mendapatkan data kedalaman juga didapatkan data *backscatter* dan data kolom air (*watercolumn*).

Data *backscatter* yang didapatkan saat pengukuran menggunakan MBES dapat digunakan untuk menentukan jenis sedimen perairan tersebut. *Backscatter* atau nilai hambur balik merupakan nilai intensitas akustik yang dipantulkan oleh dasar perairan menggunakan fungsi respon sudut pancaran (*angular response*) (Indramawan dkk., 2018). Data *backscatter* yang telah diolah akan menghasilkan mosaik *backscatter*. Mosaik *backscatter* adalah representasi dari intensitas akustik yang dipantulkan oleh dasar perairan yang digambarkan dengan derajat keabuan (*gray level*) (Jayanto dkk., 2021). Dalam mosaik *backscatter* terdapat informasi lain seperti informasi mengenai tekstur (Zhu dkk., 2021). Informasi tekstur yang dimaksud tidak hanya mencerminkan informasi permukaan dasar perairan dan hubungannya dengan lingkungan sekitarnya tetapi juga informasi mengenai struktur makro dan mikro dalam dasar perairan (Haralick dkk., 1973; Sun dkk., 2005). *Gray Level Cooccurrence Matrix* (GLCM) adalah metode paling efektif yang digunakan untuk menjelaskan dan menganalisis mengenai tekstur dasar perairan (Zhu dkk., 2021). Dalam penelitian ini akan digunakan 4 pola fitur atau *texture feature* yaitu *Angular Second Moment* (ASM), Energi (*Energy*), Kontras (*Contrast*), dan Korelasi (*Correlation*) (Zhu dkk., 2021). Mosaik *backscatter* dan pola fitur dari mosaik *backscatter* ini yang nanti akan digunakan untuk mendapatkan klasifikasi jenis sedimen dasar laut.

Sedimen di dasar perairan tersusun oleh pecahan batuan, mineral atau bahan organik yang terbawa dari berbagai sumber secara vertikal maupun secara horizontal. Karakteristik atau ciri dari sedimen dapat dilihat dari sifat-sifat alami yang dimilikinya seperti ukuran butir (*grain size*), densitas, kecepatan jatuh, komposisi, porositas, bentuk, dan sebagainya (Rifardi, 2012). Untuk mengetahui klasifikasi jenis sedimen dasar perairan menggunakan *backscatter* dapat ditentukan berdasarkan amplitudo dari gelombang suara yang kembali. Jika sinyal yang diterima merupakan sinyal yang kuat maka menunjukkan permukaan dasar perairan yang keras (*rock, gravel*) dan jika sinyal yang diterima merupakan sinyal yang lemah maka menunjukkan dasar perairan yang lebih halus (*silt, clay*) (Adi dkk., 2016).

Untuk dapat menentukan klasifikasi sedimen dasar perairan dapat digunakan model klasifikasi *supervised* dan *unsupervised*. Klasifikasi *supervised* memiliki akurasi yang lebih tinggi dibandingkan klasifikasi *unsupervised* karena klasifikasi *supervised* menggunakan contoh sedimen laut. Salah satu contoh model klasifikasi *supervised* yang umum digunakan dalam klasifikasi sedimen dasar perairan adalah *Neural Network* (Zhu dkk., 2021). Pada penelitian ini akan digunakan salah satu metode *Neural Network* yaitu *Dense Neural Network*. Metode *Dense Neural Network* atau dikenal juga dengan istilah *Fully-connected Neural Network* merupakan metode dengan bentuk yang paling sederhana (Amherd & Rodriguez, 2021). Metode ini nantinya akan menggabungkan mosaik *backscatter* dan pola fitur dari mosaik *backscatter* yang meliputi *Angular Second Moment* (ASM), Energi (*Energy*), Kontras (*Contrast*), dan Korelasi (*Correlation*) sebagai *input* sehingga dihasilkan klasifikasi jenis sedimen yang akan diolah kembali hingga menjadi peta persebaran sedimen dasar perairan.

Penelitian ini dilakukan untuk mengetahui klasifikasi jenis sedimen dan sebarannya di perairan sekitar Pelabuhan Benoa, Bali. Dari penelitian ini diharapkan dapat menjadi acuan dalam keselamatan navigasi pelayaran yang ada di Pelabuhan Benoa, Bali. Dalam penelitian Tugas Akhir ini akan dibahas mengenai model yang digunakan untuk klasifikasi jenis sedimen dasar laut dan hasil akurasi yang diperoleh dari model. Penelitian ini juga akan membahas pengolahan data mosaik dan pola fitur *backscatter* dengan menggunakan metode *Dense Neural Network* sehingga akan didapatkan hasil klasifikasi jenis sedimen dan persebaran sedimen di wilayah Pelabuhan Benoa, Bali.

## 1.2 Rumusan Permasalahan

Berdasarkan latar belakang penelitian yang telah dijelaskan sebelumnya, maka didapatkan rumusan masalah sebagai berikut:

1. Bagaimana penentuan desain yang tepat untuk metode *Dense Neural Network*?
2. Bagaimana hasil akurasi desain untuk klasifikasi jenis sedimen dasar laut di area penelitian?
3. Bagaimana klasifikasi jenis sedimen dan persebarannya dari hasil pengolahan data mosaik dan pola fitur *backscatter* menggunakan metode *Dense Neural Network*?

## 1.3 Batasan Masalah

Agar pembahasan dalam penelitian ini memiliki arah yang jelas dan berfokus pada pokok permasalahan, maka dibuatlah batasan masalah penelitian sebagai berikut:

1. Daerah yang menjadi objek penelitian adalah perairan di sekitar Pelabuhan Benoa yang berada di Desa Pedungan, Kecamatan Denpasar Selatan, Kota Denpasar, Bali.
2. Data yang digunakan yaitu data mosaik *backscatter* dan data pola fitur dari mosaik *backscatter* dari hasil survei *Multibeam Echosounder*.
3. Pengolahan data menggunakan metode *Dense Neural Network*.
4. *Software* yang digunakan untuk pengolahan data *Multibeam Echosounder* yaitu EIVA.
5. Sampel sedimen yang didapat dari pengambilan langsung di lapangan berdasarkan lokasi yang sudah ditentukan dijadikan sebagai data validasi.
6. Fokus penelitian ini yaitu mengetahui klasifikasi jenis sedimen dasar laut beserta persebarannya menggunakan data mosaik dan pola fitur *backscatter* dengan metode *Dense Neural Network*.

## 1.4 Tujuan

Tujuan dari adanya penelitian ini adalah sebagai berikut:

1. Penentuan desain yang tepat untuk metode *Dense Neural Network*.
2. Penentuan hasil akurasi desain untuk klasifikasi jenis sedimen dasar laut di area penelitian.
3. Pengolahan klasifikasi jenis sedimen dan persebarannya dari hasil pengolahan data mosaik dan pola fitur *backscatter* menggunakan metode *Dense Neural Network*.

## 1.5 Manfaat

Beberapa manfaat yang dapat dihasilkan dari penelitian ini adalah sebagai berikut:

- a. Memberikan dan meningkatkan pemahaman serta pengetahuan mengenai pengolahan klasifikasi jenis sedimen dasar laut beserta persebarannya menggunakan metode *Dense Neural Network* dengan data mosaik dan pola fitur *backscatter*.
- b. Memberikan informasi mengenai klasifikasi jenis sedimen dasar laut beserta persebarannya di sekitar Pelabuhan Benoa yang dapat meningkatkan keselamatan dalam transportasi laut.
- c. Membantu pihak swasta maupun pemerintah dalam penentuan klasifikasi jenis sedimen dasar laut menggunakan metode *Dense Neural Network* dengan data mosaik dan pola fitur *backscatter*.

*“Halaman ini sengaja dikosongkan”*

## BAB II TINJAUAN PUSTAKA

### 2.1 Penelitian Terdahulu

Dalam penelitian ini menggunakan beberapa referensi penelitian terdahulu. Penelitian yang dilakukan oleh Ji dkk (2020) mengenai klasifikasi sedimen dasar laut menggunakan data *backscatter Multibeam Echosounder* berdasarkan model *Random Forest* yang optimal. Penelitian ini berfokus pada koreksi intensitas *backscatter*, peningkatan kualitas gambar sonar, dan pembuatan klasifikator yang bertujuan untuk meningkatkan akurasi dari klasifikasi jenis sedimen dasar laut di sisi selatan Semenanjung Jiaodong, China. Dalam penelitian ini dibandingkan 3 model klasifikasi yaitu *Support Vector Machine* (SVM), *Random Forest* (RF), dan *Selecting Optimal Random Forest* (SORF). Dari ketiga klasifikasi tersebut didapatkan bahwa SORF memiliki akurasi yang paling tinggi yaitu 85%, sedangkan SVM 75.91% dan RF 76.94%. Selain itu, algoritma SORF mempunyai kinerja baik dalam hal efisiensi komputasi terutama ketika fitur berdimensi tinggi yang dimasukkan. Sehingga dapat disimpulkan bahwa model klasifikasi SORF lebih efektif penggunaannya dibandingkan dengan SVM dan RF. Perbedaan penelitian ini dengan penelitian tugas akhir penulis berada pada metode yang digunakan. Selain itu, tujuan dari penelitian ini lebih pada peningkatan akurasi dari klasifikasi jenis sedimen yang mana tujuan ini berbeda dengan tujuan penelitian tugas akhir penulis yang bertujuan untuk mengetahui klasifikasi jenis sedimen dan persebarannya. Lokasi penelitian dari keduanya juga berbeda. Lokasi penelitian ini di sisi selatan Semenanjung Jiaodong, China sedangkan lokasi penelitian tugas akhir penulis berada di perairan sekitar Pelabuhan Benoa.

Referensi penelitian lain adalah penelitian yang dilakukan oleh Fariyah dkk (2020) mengenai pengukuran dan analisis hambur balik akustik menggunakan teknologi *Multibeam Echosounder* untuk klasifikasi sedimen dasar laut teluk palu. Penelitian ini bertujuan untuk mendeteksi, mengklasifikasi dan memprediksi tipe sedimen dasar laut menggunakan data *backscatter* dengan metode *Angular Response Analysis* (ARA) dan *Support Vector Machine* (SVM) di Teluk Palu. *Software* yang digunakan pada penelitian ini yaitu *Caris Hips and Sips* 8.1 untuk pengolahan data batimetri dan *Python* 3.8.2 serta *Visual Studio Code* untuk pengolahan dan visualisasi SVM. Hasil dari penelitian ini dengan metode ARA sedimen didominasi oleh pasir (*sand*) dan lanau (*silt*) sedangkan dengan metode SVM didominasi oleh pasir berlanau (*silty sand*), lanau (*silt*) dan pasir (*sand*). Untuk hasil dari uji akurasi metode ARA sebesar 50% sedangkan hasil uji akurasi untuk metode SVM menghasilkan *overall accuracy* dengan nilai 60%. Kesimpulan dari penelitian ini bahwa hasil dengan menggunakan metode SVM lebih mendekati dengan keadaan sebenarnya dengan jenis sedimen yaitu pasir berlanau, lanau, dan pasir. Penelitian ini memiliki perbedaan dengan penelitian tugas akhir penulis pada bagian metode yang digunakan untuk klasifikasi jenis sedimen. Selain itu, lokasi penelitian ini berada di Teluk Palu, sedangkan lokasi penelitian tugas akhir penulis berada di perairan sekitar Pelabuhan Benoa.

Selain itu terdapat penelitian yang dilakukan oleh Cui dkk (2021) mengenai penggunaan model *deep learning* untuk klasifikasi sedimen dasar laut berdasarkan optimasi fitur *fuzzy ranking*. Dalam penelitian ini akan dibahas penentuan klasifikasi jenis sedimen menggunakan data batimetri dan data *backscatter* dengan metode optimasi fitur *fuzzy ranking* (FR) yang akan dibandingkan dengan penggunaan *Deep Belief Network* (DBN) untuk klasifikasi sedimen. Kedua metode tersebut menggunakan data *ground-truth* untuk validasi. Hasil dari penelitian ini menunjukkan bahwa metode FR meningkatkan kualitas fitur yang dimasukkan ke dalam klasifikator input, sedangkan metode DBN meningkatkan signifikan akurasi dan stabilitas klasifikasi. Perbedaan penelitian ini dengan penelitian tugas akhir penulis berada pada bagian metode.

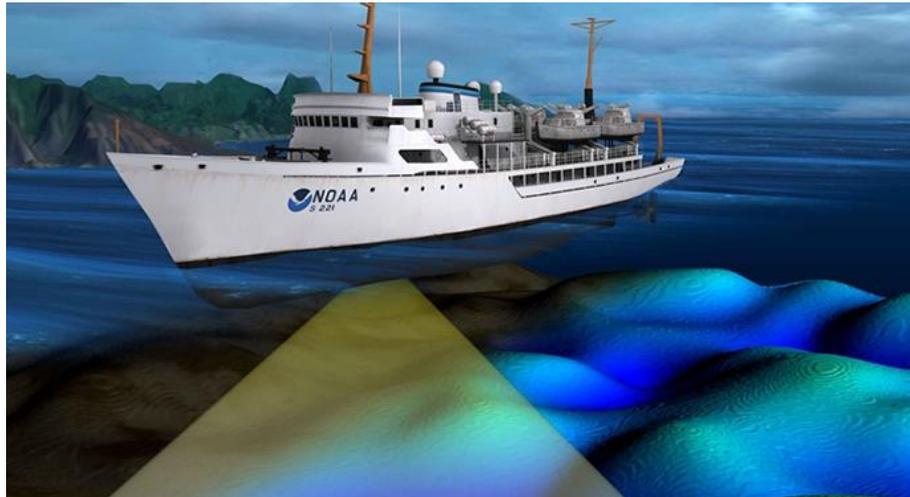
Penelitian terakhir yang dapat dijadikan referensi adalah penelitian yang dilakukan oleh Zhu dkk (2021) mengenai penggunaan *Deep Neural Network* untuk klasifikasi sedimen dasar laut menggunakan fitur *Multibeam Echosounder* yang berbeda bobotnya. Dalam penelitian ini akan dibahas penentuan klasifikasi jenis sedimen dasar laut menggunakan mosaik *backscatter* yang diintegrasikan dengan tiga jenis fitur lainnya yaitu fitur batimetri *Multibeam Echosounder*, *texture feature* atau pola fitur mosaik *backscatter*, dan fitur respon sudut *backscatter* atau *Angular Response (AR)* yang diberikan bobot yang berbeda dalam proses klasifikasi. Ketiga perpaduan tersebut diolah menggunakan *Deep Neural Network (DNN)* sehingga dapat memprediksi jenis sedimen dasar laut. Hasil dari penelitian ini menunjukkan bahwa akurasi klasifikasi keseluruhan dari sedimen dasar laut dan koefisien kappa dapat ditingkatkan secara signifikan menggunakan model klasifikasi DNN dengan bobot yang berbeda. Penelitian ini menggunakan metode *Deep Neural Network* sedangkan untuk penelitian tugas akhir ini menggunakan metode *Dense Neural Network*.

Berdasarkan keempat penelitian terdahulu yang telah dilakukan, yang membedakan dengan tugas akhir ini adalah lokasi penelitian, tujuan penelitian, serta metode yang digunakan. Pada penelitian ini lokasi yang digunakan yaitu di perairan di sekitar Pelabuhan Benoa, Kota Denpasar, Bali. Data yang digunakan pada penelitian ini berasal dari instrumen *Multibeam Echosounder*. Penelitian ini bertujuan untuk menentukan model DNN yang baik dengan nilai akurasi yang tinggi sehingga akan menghasilkan klasifikasi yang akurat, Penelitian tugas akhir ini juga bertujuan untuk mengetahui klasifikasi jenis sedimen dasar laut dan persebarannya menggunakan data mosaik *backscatter* dan pola fitur atau *texture feature* dari mosaik *backscatter* dengan metode *Dense Neural Network (DNN)*.

## 2.2 Survei Hidrografi

Secara etimologi, kata hidrografi bahasa Yunani yang terdiri dari kata “hidro” yang berarti air dan “grafi” yang berarti menulis. Menurut *International Hydrographic Organization (IHO)*, hidrografi adalah cabang keilmuan yang mempelajari pengukuran dan penggambaran ciri-ciri fisik yang ada pada laut dan daerah pantai yang bertujuan untuk navigasi dan semua aktivitas yang berhubungan dengan laut, termasuk aktivitas lepas pantai, penelitian, dan perlindungan lingkungan. Sedangkan survei hidrografi menurut IHO (2008) adalah ilmu yang mempelajari pengukuran serta penggambaran parameter-parameter untuk menjelaskan sifat-sifat dan konfigurasi dasar laut secara tepat, hubungan geografisnya dengan daratan, serta karakteristik-karakteristik dan dinamika-dinamika lautan. Survei hidrografi merujuk pada penelitian dan pemetaan berbagai elemen yang memengaruhi navigasi di laut, pembangunan infrastruktur kelautan, pengerukan perairan, eksplorasi dan pengeboran minyak di lepas pantai, serta kegiatan terkait lainnya (Akbar dkk., 2022). Menurut Hovem (2007) menyebutkan bahwa kegiatan hidrografi memiliki tujuan untuk melakukan pengukuran kedalaman perairan, baik melalui teknik survei akustik maupun sonar, guna menghasilkan peta navigasi serta informasi dasar laut yang digunakan dalam konstruksi kelautan, eksplorasi dan pengeboran minyak di lepas pantai, serta deteksi pipa bawah laut.

Dalam kegiatan survei hidrografi terdapat beberapa aktivitas yang meliputi penentuan posisi dan penggunaan sistem referensi, pengukuran kedalaman, pengukuran arus, pengukuran sedimen termasuk pengambilan contoh dan analisis, pengamatan pasang surut, serta pengukuran detail situasi dari garis pantai untuk pemetaan pesisir. Data-data yang diperoleh dari survei hidrografi dapat disajikan dalam bentuk peta maupun non-peta serta dapat juga disusun dalam basis data kelautan.



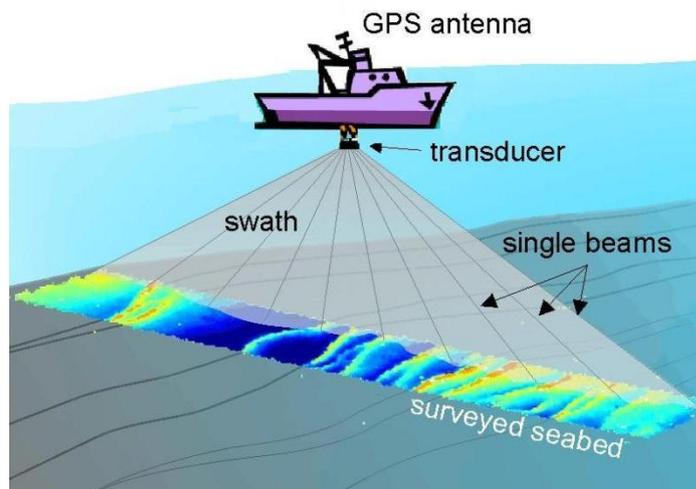
Gambar 2.1 Survei Hidrografi (NOAA, 2023)

Pentingnya melaksanakan survei hidrografi terletak pada penyediaan informasi tentang dasar perairan yang berfungsi sebagai landasan pertimbangan dalam mengembangkan pembangunan sektor perikanan dan kelautan. Dalam praktiknya, peta batimetri digunakan untuk menentukan rute pelayaran, merencanakan pembangunan pesisir, mendeteksi bencana bawah air seperti tsunami dan letusan gunung laut, memahami karakteristik morfologi suatu perairan, relief dasar bawah air, serta berbagai hal lainnya. (Sayyid dkk., 2020).

Batimetri merupakan ilmu yang mempelajari kondisi di bawah permukaan air yang berupa kedalaman. Batimetri juga merupakan salah satu metode atau teknik yang digunakan dalam penentuan kedalaman perairan dari analisa data kedalaman yang didapatkan melalui pengukuran seperti penyelaman ilmiah, pemeruman, maupun pengelolaan pesisir secara terpadu (Lesmana & Haykal, 2021). Menurut IHO (2008) survei batimetri dapat diartikan sebagai survei yang dilakukan untuk mengukur atau memetakan kedalaman di suatu perairan. Survei batimetri dilakukan untuk mendapatkan data kedalaman dan kondisi topografi dasar perairan termasuk dalam penemuan objek-objek yang terdapat di dasar perairan (Faridatunnisa dkk., 2018). Perkembangan ilmu pengetahuan dan teknologi mendorong pengembangan pengukuran kedalaman perairan, dimana awalnya menggunakan metode konvensional menggunakan batu duga (bandul pemberat), namun saat ini telah berkembang menjadi metode pengukuran hidrografi akustik (Sarono & Basith, 2022).

Survei batimetri biasanya dilakukan dengan menggunakan wahana kapal atau perahu dengan instrumen seperti *Singlebeam Echosounder* maupun *Multibeam Echosounder*. Penggunaan wahana kapal atau perahu sering mengalami kendala yaitu pada saat manuver kapal dan fleksibilitas pengambilan jalur perum. Hal ini dipengaruhi oleh dimensi dan spesifikasi wahana yang digunakan. Semakin besar wahana yang digunakan, maka akan semakin sulit wahana tersebut melakukan manuver dan begitu sebaliknya (Cahyono & Atunggal, 2012).

Pemeruman adalah kegiatan yang bertujuan untuk mendapatkan representasi visual tentang bentuk permukaan dasar perairan (*topography seabed*) melalui proses dan aktivitas tertentu. Pengukuran kedalaman dilakukan dengan menempatkan titik-titik pengukuran pada jalur-jalur yang disebut lajur perum. Jarak antara setiap titik pengukuran pada suatu lajur perum minimal sama atau lebih rapat dari interval lajur perum yang digunakan. Lajur perum dirancang dengan berbagai variasi bentuk, seperti garis lurus, lingkaran konsentrik, atau bentuk lainnya, tergantung pada metode yang digunakan. Desain lajur perum bertujuan untuk memungkinkan pendeteksian perubahan kedalaman yang lebih ekstrim dengan lebih baik (Huda dkk., 2022).



Gambar 2.2 Akuisisi Data Survei Hidrografi (Marine Matcher, 2023)

Dalam survei batimetri, hasil pengukuran dapat disajikan dalam bentuk peta batimetri yang umumnya menampilkan topografi dasar laut beserta garis-garis kontur dan juga menampilkan informasi tambahan berupa informasi navigasi permukaan. Peta batimetri dapat digunakan untuk berbagai keperluan seperti untuk peta perencanaan, peta konstruksi serta digunakan untuk kegiatan-kegiatan perencanaan pelabuhan, pengerukan, pemasangan pipa atau kabel bawah laut, dan berbagai konstruksi di laut (Prananda dkk., 2017). Peta batimetri merupakan representasi visual dari survei batimetri yang menyajikan informasi tentang kedalaman dan lokasi perairan, di mana data pengukuran survei dapat diwujudkan dalam bentuk tampilan 2 dimensi (2D) atau 3 dimensi (3D).

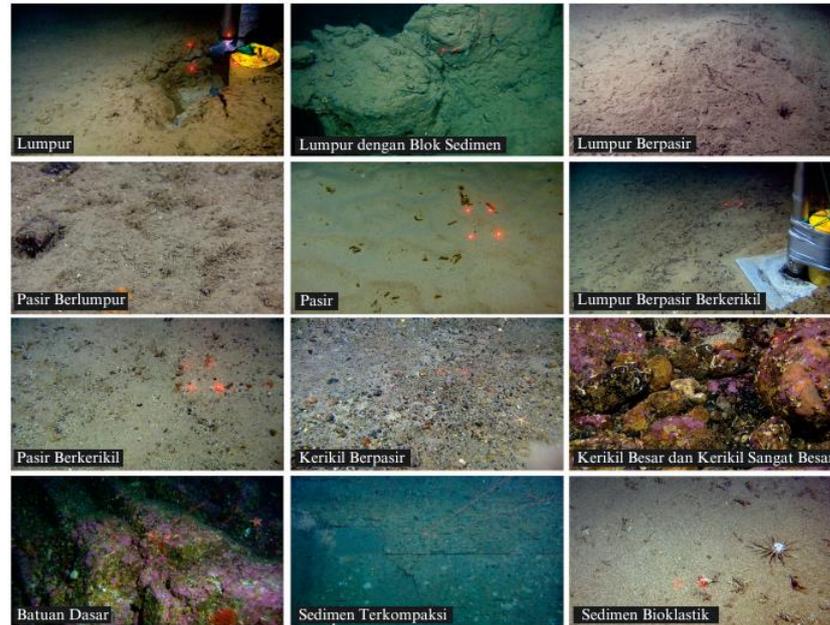
Pemetaan batimetri merupakan kegiatan yang bertujuan untuk menyediakan informasi spasial yang diperlukan dalam perencanaan, pengambilan keputusan, dan kegiatan di bidang kelautan. Tujuan dari pemetaan batimetri adalah mengumpulkan data dengan referensi geografis yang terkait dengan garis pantai, kedalaman perairan, eksplorasi dan eksploitasi sumber daya di wilayah laut dan pesisir. Pemetaan batimetri juga memiliki peran penting dalam navigasi dan keselamatan pelayaran, perencanaan pembangunan pesisir, serta kebutuhan pertahanan maritim (Ramadhan dkk., 2021).

Dari penjelasan di atas dapat disimpulkan bahwa survei hidrografi merujuk pada keseluruhan proses yang meliputi pengukuran, pengolahan data, dan visualisasi untuk menggambarkan secara komprehensif dasar perairan tersebut. Sedangkan survei batimetri merupakan survei yang dilakukan dengan tujuan untuk memperoleh informasi tentang kedalaman dan mengukur nilai kedalaman dasar laut.

### 2.3 Sedimen Dasar Laut

Pada umumnya sedimen merupakan partikel yang berasal dari cangkang, sisa-sisa kerangka organisme maupun pembongkaran batuan (Bayhaqi & Dungga, 2015). Partikel-partikel sedimen mempunyai ukuran yang besar (*boulders*) sampai ukuran yang sangat halus (*koloid*) dan mempunyai bentuk beragam mulai dari bulat, lonjong, sampai persegi (Usman, 2014). Proses mekanik arus yang berasal dari sungai dan atau arus laut dapat mengendapkan material sedimen. Arus dan gelombang juga merupakan faktor yang menentukan arah serta sebaran sedimen (Sasmito, 2020). Sebaran dan perbedaan ukuran dari partikel sedimen dapat dijadikan indikator perilaku pada aliran sedimen di suatu wilayah perairan (Nugroho & Basit, 2014; Purnawan dkk., 2015). Ponce (1989) menyatakan bahwa batu besar (*boulders*) dan krakal (*cobbles*) dapat diukur tersendiri, kerikil (*gravel*) dapat diukur tersendiri atau dengan

menggunakan ayakan, sedangkan pasir diukur dengan ayakan. Ayakan nomor 200 dapat digunakan untuk memisahkan partikel pasir dari partikel yang lain yang lebih halus seperti lumpur dan lempung, sedangkan lumpur dan lempung dapat dipisahkan dengan mengukur perbedaan kecepatan jatuhnya pada air diam.



Gambar 2.3 Contoh Sedimen Dasar Laut (Bellec dkk., 2017)

Sedimen dasar laut dapat memberikan informasi tentang kondisi geologi dari masa lampau hingga masa kini. Faktor-faktor hidro-oseanografi mempengaruhi transportasi dan pengendapan material sedimen di dasar laut. Seiring berjalannya waktu, sedimen yang terendapkan mengalami proses sedimentasi dan membentuk lapisan sedimen di dasar laut. Dengan memahami kondisi ini, dapat diketahui bagaimana lapisan sedimen terbentuk serta faktor oseanografi yang memengaruhinya (Stephanie dkk., 2014).

Sedimen laut berasal dari berbagai sumber, termasuk daratan dan aktivitas biologis, fisik, dan kimia yang terjadi baik di daratan maupun di laut itu sendiri, meskipun ada kontribusi kecil dari sumber vulkanik dan kosmik. Sedimen laut terdiri dari materi yang berasal dari berbagai sumber. Faktor-faktor seperti topografi bawah laut dan pola iklim mempengaruhi jenis sedimen yang terakumulasi. Distribusi laut saat ini mencerminkan iklim dan pola arus. Tipe sedimen di dasar laut berubah seiring waktu karena perubahan cekungan laut, arus, dan iklim. Urutan dan karakteristik sedimen, termasuk struktur dan tekstur yang terlihat dalam lapisan sedimen, mengindikasikan sebagian perubahan yang terjadi di atasnya (Rifardi, 2012).

Sedimen yang berada di dasar laut mempunyai peran penting dalam rekayasa teknik bawah laut dan arkeologi bawah laut (Wu dkk, 2021; Li dkk, 2021). Menurut Boggs (1986) sedimen dasar laut tersusun oleh material biogenik yang berasal dari organisme, material autigenik hasil dari proses kimiawi laut (seperti glaukonit, garam, dan fosfor), material residual, material sisa endapan sebelumnya, dan material detritus sebagai hasil erosi asal daratan (seperti pasir, kerkil, lanau, dan lempung). Pada tahun 1922 Uden Wentworth telah menemukan pengelompokan sedimen menurut ukuran partikel atau butirnya. Pengelompokan tersebut dikenal dengan nama Skala Wentworth. Secara spesifik, menghitung parameter statistik sedimen dapat memperoleh gambaran lingkungan pengendapan. Penentuan ukuran partikel atau butir dengan menggunakan skala *phi* atau *phi scale* ( $\phi$ ) dapat mempermudah pengklasifikasian suatu sampel sedimen. Parameter statistik dihitung dengan menggunakan persamaan logaritmik (Folk & Ward, 1957):

$$\varphi = -\log_2 d \quad (2.1)$$

dimana  $\varphi$  = skala *phi*  
 $d$  = diameter partikel sedimen

Berikut merupakan tabel Skala Wentworth beserta skala *phi* yang tercantum dalam Tabel 2.2 sebagai berikut:

Tabel 2.1 Skala Wentworth (Hedrick dkk., 2013)

Nama Partikel	Diameter Partikel (mm)	Skala ( $\varphi$ )
Bongkah ( <i>Boulder</i> )	>256	-8
Kerakal ( <i>Cobble</i> )	64 – 256	-6
Kerikil ( <i>Pebble</i> )	4 – 64	-2
Butir ( <i>Granule</i> )	2 – 4	-1
Pasir sangat kasar ( <i>Very coarse sand</i> )	1 – 2	0
Pasir kasar ( <i>Coarse sand</i> )	0,5 – 1	1
Pasir sedang ( <i>Medium sand</i> )	0,25 – 0,5	2
Pasir halus ( <i>Fine sand</i> )	0,125 – 0,25	3
Pasir sangat halus ( <i>Very fine sand</i> )	0,0625 – 0,125	4
Lanau ( <i>Silt</i> )	0,004 – 0,0625	8
Lempung ( <i>Clay</i> )	<0,004	>8

Terdapat berbagai alat yang dapat digunakan untuk mengumpulkan sedimen dasar laut sesuai dengan kontaminan dan informasi yang dibutuhkan. Salah satu alat yang digunakan untuk mengambil sedimen di dasar laut adalah *grab sediment sampler*. *Grab sediment sampler* memiliki kemudahan dalam pengoperasian dan mampu memberikan sampel sedimen dasar laut yang cocok untuk analisis butiran di laboratorium. Alat ini juga mudah dikerahkan, bahkan dari kapal kecil dan dalam kondisi laut yang kasar. Tersedia beberapa jenis *grab sediment sampler* di pasaran, dan yang paling umum digunakan adalah jenis *Shipek*, *Van Veen*, *Petersen*, dan *Smith-McIntyre* (Khomsin dkk., 2021).



Gambar 2.4 Van Veen Grab (Aquatic BioTechnology, 2023)

Penggunaan grab sediment samplers lebih efektif untuk mengumpulkan sedimen yang bersifat kohesif dan berbutir halus, seperti lumpur dan tanah liat, daripada sedimen yang tidak bersifat kohesif seperti pasir, kerang kerucut, dan kerikil. Alat ini umumnya mudah digunakan (bahkan dari kapal kecil dan dalam kondisi laut yang kasar) dan dapat memberikan sampel yang berukuran cukup besar. Pengambilan sampel sedimen dasar laut menggunakan grab sampler memiliki beberapa kelemahan. Kelemahan pertama terletak pada keterbatasan alat ini dalam mengambil sedimen dasar laut hanya pada perairan dangkal. Semakin dalam perairan, semakin sulit untuk mengumpulkan sedimen dasar karena meningkatnya tekanan yang sulit untuk diatasi, terutama pada sedimen yang keras seperti pasir, batu, dan kerikil. Kelemahan kedua adalah kesulitan alat ini dalam mengambil sampel sedimen yang terlalu lunak, seperti lumpur dan tanah liat, yang cenderung mudah terlepas atau kehilangan sedimen halus. Selain itu, terdapat kelemahan dalam akurasi menentukan posisi sedimen dasar laut, karena posisi kapal dapat berpindah ke lokasi lain akibat arus, gelombang, atau faktor lainnya saat pengambilan sampel. Selain itu, alat ini hanya memberikan informasi tentang permukaan dasar laut dan tidak memberikan informasi tentang kondisi dasar laut secara menyeluruh (Khomsin dkk., 2021).

Tidak seperti grab sediment sampler, *sediment core* dapat mengambil sedimen dari permukaan dasar laut dan menangkap lapisan stratigrafi dengan kedalaman. Terdapat beberapa jenis *sediment core* yang tersedia, seperti *gravity corers*, *multiple gravity corers*, *hydraulically damped corers*, *box corers*, *piston corers*, *freeze corers*, *Vibro corers*, dan *drilling*. Pemilihan jenis *sediment core* ini bergantung pada kondisi lingkungan dan tujuan penelitian yang diinginkan. *Gravity corer* adalah jenis yang paling umum digunakan untuk mengumpulkan sedimen dasar laut. Alat ini mampu dengan cepat dan berkelanjutan mengambil sedimen dasar laut hingga beberapa ribu meter dari dasar laut. Alat ini juga dapat dioperasikan dengan berbagai jenis kendaraan. Penggunaan *gravity corer* meliputi pengerukan, rekayasa minyak dan gas lepas pantai, jalur pipa dan kabel, serta sangat berguna dalam mengendalikan jenis tanah dalam survei geofisika (Khomsin dkk., 2021).



Gambar 2.5 *Gravity Corer* (Carmacoring, 2023)

Seperti halnya alat pengambil *grab sediment sampler*, *gravity corer* juga memiliki kelemahan yang serupa ketika digunakan untuk pemetaan sedimen dasar laut. Alat ini tidak mampu menghasilkan data sedimen primer secara kontinu, sehingga semakin banyak sampel yang diperlukan, semakin lama waktu yang dibutuhkan, dan pada akhirnya biaya yang dikeluarkan juga menjadi lebih tinggi. *Gravity corer* hanya efektif untuk mengumpulkan lempung yang sangat lembut hingga keras, karena penetrasi pada lempung keras atau pasir terbatas, dan sampel yang dihasilkan memiliki kualitas biasa hingga sedang. Pengambilan *core sampel* tidaklah mudah seperti pengambilan *grab sampler*. Teknik ini mahal karena memerlukan kapal dan peralatan khusus, pengukuran yang memakan waktu, dan analisis yang membutuhkan banyak tenaga kerja setelahnya. Kelemahan penting adalah teknik ini hanya memberikan informasi tentang posisi titik saja. Peta distribusi sedimen dasar laut dari *gravity corer* memiliki resolusi spasial rendah. Semakin padat titik sampelnya, semakin tinggi resolusinya, dan sebaliknya. Namun, semakin dekat titik sampelnya, akan memakan waktu lama dan mahal (Khomsin dkk., 2021). Pada penelitian ini, untuk pengumpulan sampel sedimen digunakan metode *sediment core*.

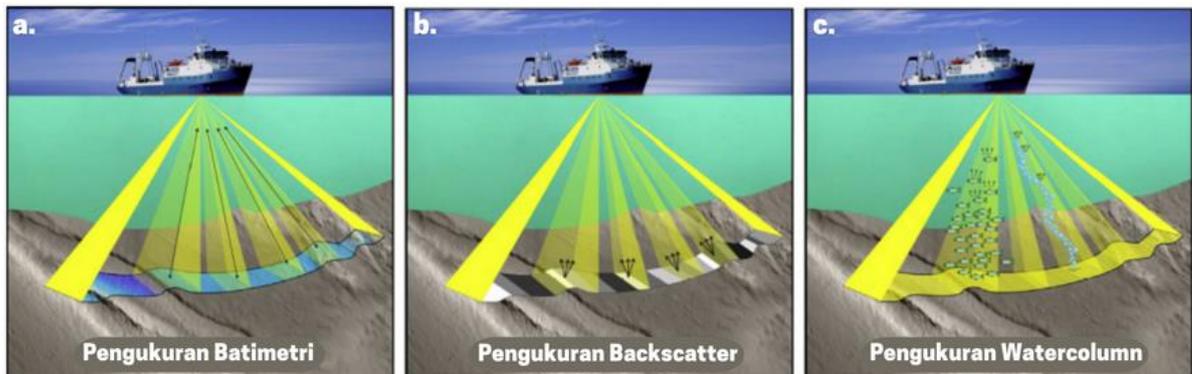
## 2.4 *Multibeam Echosounder*

### 2.4.1 Definisi

*Multibeam echosounder* merupakan salah satu instrumen yang digunakan untuk mengukur kedalaman perairan. *Multibeam echosounder* menghasilkan data pengukuran yang efektif dan akurat serta mencakup dasar laut yang lebih luas dibandingkan dengan penggunaan *Singlebeam Echosounder* (Manik dkk., 2014). *Multibeam Echosounder* memancarkan lebih dari satu *beam* dalam satu kali pancaran sinyal. Pada umumnya sistem pemancaran dari instrumen ini menggunakan sistem *swath* atau sapuan, dimana satu pancaran sinyal yang memiliki panjang dan lebar yang membentuk sebuah kolom (de Jong, 2002). *Multibeam Echosounder*, sebagai instrumen survei, tidak dapat bekerja secara independen. Sistem *Multibeam Echosounder* merujuk pada sekelompok sensor yang dipasang pada sebuah wahana survei dan digunakan bersama-sama untuk mendapatkan pengukuran kedalaman secara *real-time* (Akbar dkk., 2022).

Data yang bertampalan dari hasil survei yang diperoleh dari *Multibeam Echosounder* akan melalui proses validasi untuk memastikan tingkat kevalidan yang tinggi, sehingga semakin banyak data yang terkumpul, semakin akurat pula hasil survei yang dapat diperoleh. Data hasil akuisisi menggunakan *Multibeam Echosounder* terbagi menjadi 3 data, yaitu data batimetri (kedalaman), data *backscatter*, dan data *watercolumn*. Batimetri adalah pengukuran kedalaman laut dan topografi dasar laut. Data ini digunakan untuk memetakan kontur dan kedalaman laut, menghasilkan peta batimetri yang menunjukkan perubahan elevasi dan struktur dasar laut. Pengukuran *backscatter* adalah pengukuran intensitas pantulan suara dari dasar laut. Data *backscatter* memberikan informasi tentang komposisi, tekstur, dan reflektivitas permukaan dasar laut. Hal ini dapat membantu dalam identifikasi jenis sedimen, kandungan organik, dan keberadaan struktur geologis seperti batuan atau terumbu karang. *Multibeam Echosounder* juga dapat melakukan pengukuran kolom air, yang berarti mencatat pantulan suara di dalam kolom air di atas dasar laut. Data *water column* memberikan informasi tentang keberadaan dan distribusi organisme seperti ikan atau plankton dalam kolom air. Pengukuran ini dapat membantu dalam pemahaman tentang keanekaragaman biologis dan aktivitas organisme di perairan. Aplikasi dari *Multibeam Echosounder* meliputi pengukuran kedalaman perairan, pemetaan jalur pipa, deteksi *hydrothermal vent*, deteksi kebocoran minyak dan gas, navigasi laut, serta pencitraan objek kolom air (Muljawan dkk., 2020). Survei batimetri *multibeam* merupakan

tugas yang kompleks karena melibatkan pengumpulan data yang terintegrasi di bawah kondisi lingkungan yang dinamis (Nugroho dkk., 2022).



Gambar 2.6 Pengukuran MBES (a) Batimetri (b) *Backscatter* (c) *Watercolumn* (Colbo dkk., 2014)

Prinsip kerja dari *Multibeam Echosounder* yaitu gelombang akustik dari wahana kapal atau perahu di pancarkan dengan transduser ke dasar laut dalam satu kali pancaran sinyal. Sinyal yang dipancarkan tersebut akan melewati kolom air dan jika sinyal sudah mencapai dasar maka akan dikembalikan atau dipantulkan ke permukaan kembali. Sinyal tersebut akan diterima oleh *receiver* atau penerima yang ada di kapal. Apabila titik-titik kedalaman yang telah diukur dihubungkan satu sama lain dapat membentuk suatu profil permukaan dasar laut. Jika kapal bergerak maju, pola pancaran *beam* yang melebar dan melintang terhadap badan wahana akan membentuk peta kedalaman (Manik dkk., 2016). Sonar *Multibeam Echosounder* tidak hanya digunakan untuk mengumpulkan data batimetri, tetapi juga mampu mengambil data *backscatter* dan data kolom air dari objek sebagai bagian dari proses investigasi, termasuk di perairan Indonesia yang kaya akan objek seperti bangkai kapal tenggelam (*wreck*), gunung api bawah laut, dan jalur pipa dari pengeboran minyak bumi dan gas lepas pantai, sehingga alat *Multibeam Echosounder* dapat memberikan informasi dan data yang lebih detail tentang objek-objek tersebut (Ibrahim dkk., 2022). Pengolahan data *Multibeam Echosounder* memiliki proses yang rumit dan mendetail melibatkan serangkaian koreksi yang diperlukan untuk mendapatkan nilai kedalaman yang akurat, termasuk koreksi pergerakan kapal selama pengukuran dan faktor oseanografi di wilayah survei, seperti koreksi pasang surut, profil kecepatan suara, dan pergerakan kapal (Brammadi dkk., 2017).

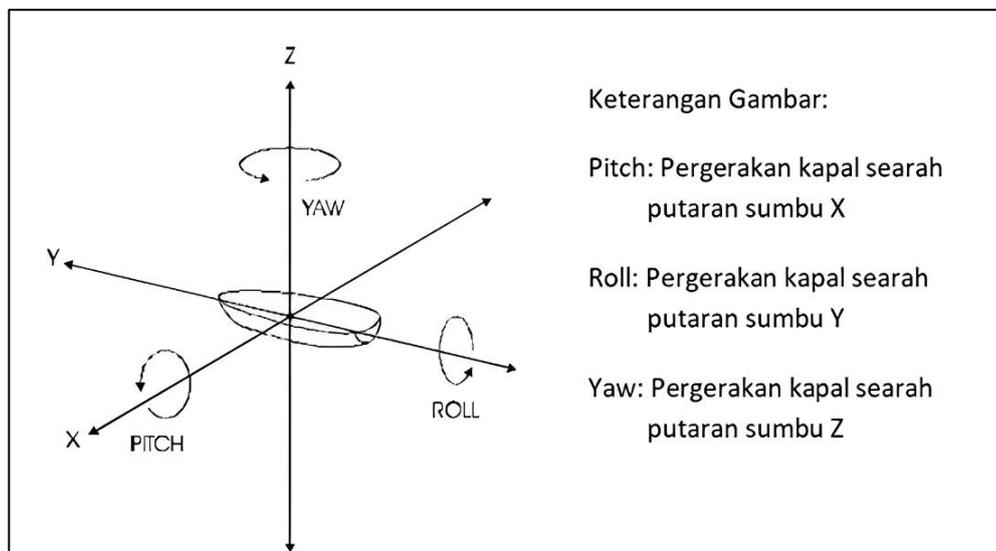
#### 2.4.2 Kalibrasi *Multibeam Echosounder*

Pada pengukuran kedalaman di perairan atau laut menggunakan *Singlebeam Echosounder*, hanya parameter *heave* yang biasanya digunakan sebagai parameter pergerakan wahana survei, sedangkan *roll*, *pitch*, dan *yaw* tidak mempengaruhi hasil pengukuran kedalaman. Namun, hal ini tidak berlaku pada pengukuran kedalaman menggunakan *Multibeam Echosounder*. Pada *Multibeam Echosounder*, pergerakan wahana survei menjadi parameter penting yang harus diketahui secara *real-time* untuk perhitungan kedalaman. *Multibeam Echosounder* tidak langsung mengukur kedalaman, tetapi mengukur jarak dan arah (*range* dan *bearing*). Kedalaman perairan dengan posisi yang tetap diperoleh melalui perhitungan yang melibatkan *range*, *bearing*, posisi, pergerakan (*roll*, *pitch*, dan *yaw*), dan refraksi. Oleh karena itu, data *roll*, *pitch*, dan *yaw* memainkan peran penting dalam menghasilkan data kedalaman dari akuisisi data *Multibeam Echosounder*. Inilah alasan mengapa kalibrasi harus dilakukan untuk

menentukan sudut kesalahan pemasangan transduser *Multibeam Echosounder*, seperti sudut kesalahan *roll*, *pitch*, dan *heading*. Untuk mendapatkan sudut kesalahan tersebut, dapat dilakukan pengukuran dengan pola lajur tertentu (Muljawan dkk., 2020).

Pergerakan wahana survei dapat diukur menggunakan parameter *roll*, *pitch*, dan *yaw*, yang mengacu pada rotasi pada sumbu Y (*roll*), rotasi pada sumbu X (*pitch*), dan rotasi pada sumbu Z (*yaw*). Sistem koordinat yang terbentuk oleh ketiga sumbu ini dikenal sebagai sistem koordinat kapal (*Ship Reference Frame/SRF*) (Muljawan dkk., 2020). Dengan menggunakan SRF, data yang dikumpulkan selama survei hidrografi dapat direkam dengan koordinat yang terkait dengan kerangka referensi kapal. Hal ini memungkinkan pengolahan dan analisis data yang akurat dan konsisten dalam konteks kerangka referensi kapal. Berdasarkan penelitian yang dilakukan oleh Clarke (2003), sistem koordinat kapal secara tiga dimensi ditentukan dengan sumbu X positif dari *Reference Point* (RP) ke arah kiri (*port*), sumbu Y positif dari RP ke arah depan kapal (*fore*), dan sumbu Z positif dari RP ke arah bawah kapal.

Dalam pelaksanaan survei *Multibeam Echosounder*, penting untuk mengukur secara *real-time* parameter *roll*, *pitch*, dan *yaw*. Hal ini diperlukan agar posisi transduser *Multibeam Echosounder* tetap stabil dan tidak mengalami gerakan yang tidak sejalan dengan gerakan kapal. Pemasangan transduser *Multibeam Echosounder* harus dilakukan dengan meminimalkan kesalahan pemasangan dan ketidaksejajaran transduser terhadap sistem koordinat kapal. Untuk mengukur besarnya ketidaksejajaran transduser, dapat digunakan metode statis atau metode dinamis. Metode statis dapat menggunakan survei dimensional. Dengan melakukan survei dimensional dalam kondisi kapal yang stabil, akan didapatkan kerangka 3D dari sistem koordinat kapal serta mengetahui secara akurat ketidaksejajaran transduser terhadap SRF. Hal ini penting untuk memastikan bahwa data yang dihasilkan oleh *Multibeam Echosounder* dapat direferensikan dengan benar ke kerangka referensi kapal, sehingga memungkinkan pengolahan dan analisis data yang akurat dan konsisten (Muljawan dkk., 2020).



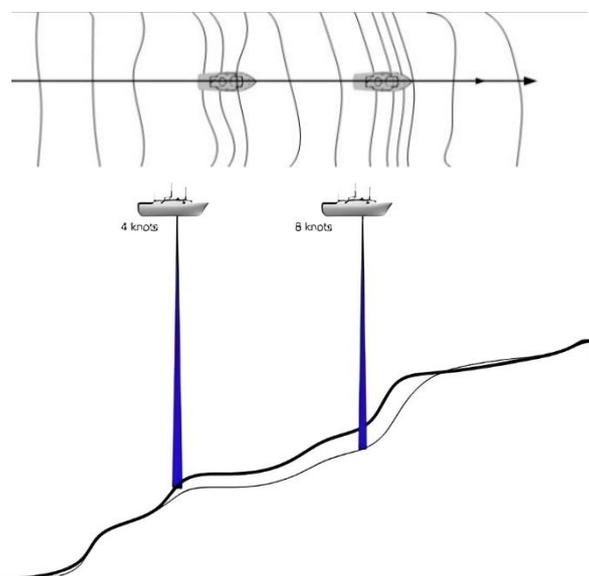
Gambar 2.7 Sistem Koordinat Kapal dan Perputaran (Muljawan dkk., 2020)

Dalam melakukan pengukuran kedalaman perairan menggunakan *Multibeam Echosounder* terdapat kesalahan-kesalahan yang mungkin terjadi. Kesalahan-kesalahan tersebut dapat dipengaruhi oleh beberapa faktor yaitu faktor *miss-alignment* transduser dan *sensor offset* dari peralatan yang terpasang di kapal. Penyimpangan (*transducer miss-*

*alignment*) yang signifikan harus diketahui dan dimasukkan nilainya ke dalam perangkat lunak sistem akuisisi data *Multibeam Echosounder* agar dapat secara otomatis mengoreksi data perekaman *Multibeam Echosounder* (Muljawan dkk., 2020). Untuk mengoreksi kesalahan-kesalahan yang terdapat dalam pengukuran menggunakan *Multibeam Echosounder* dilakukan kalibrasi *Multibeam Echosounder* (*patch test*) (Haryanto dkk., 2020). *Patch test* adalah sebuah metode yang sering digunakan dalam proses kalibrasi *Multibeam Echosounder*. Tujuannya adalah untuk memeriksa dan mengoreksi pergeseran atau distorsi pada data yang dihasilkan oleh *array* transduser *Multibeam Echosounder*. Metode ini membantu memastikan bahwa data batimetri yang diperoleh memiliki tingkat akurasi yang tinggi dan distorsi yang minimal. Kalibrasi *patch test* dilakukan dengan merekam data *Multibeam Echosounder* saat kapal melewati beberapa lajur survei yang sejajar dan saling tumpang tindih. Kecepatan kapal, kedalaman, dan karakteristik dasar laut yang ditentukan secara spesifik dalam kalibrasi ini. Pada tahap awal kalibrasi, sejumlah titik kontrol dengan kedalaman yang diketahui diperlukan untuk membandingkan hasil pengukuran *Multibeam Echosounder* dengan data referensi. Selanjutnya, dilakukan penyesuaian parameter instrumentasi agar data yang dihasilkan oleh *Multibeam Echosounder* sesuai dengan standar yang diharapkan. Parameter-parameter yang digunakan dalam kalibrasi *Multibeam Echosounder* meliputi:

1) Latensi atau *Time Delay*

Latensi adalah proses menentukan perbedaan waktu antara penandaan pemeruman (*sounding*) awal dengan akhir melalui perbedaan kecepatan wahana dari *multibeam*. Kecepatan awal yang ditentukan untuk pemeruman pertama kemudian disurvei kembali dalam arah yang sama untuk pemeruman kedua, tetapi pada kecepatan yang lebih lambat atau lebih cepat dari kecepatan survei pertama (Muljawan dkk., 2020).



Gambar 2.8 Latensi atau *Time Delay* (Brennan, 2017)

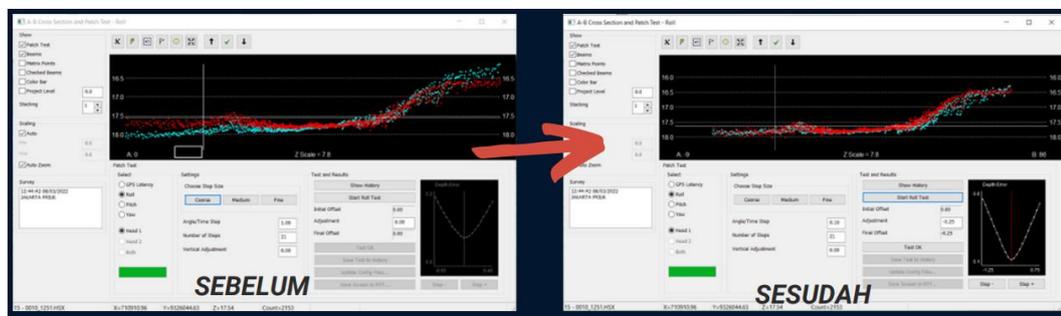
2) Kalibrasi *Roll*

*Roll* adalah parameter yang sangat penting ketika menggunakan *Multibeam Echosounder* karena biasanya sapuan akan menyapu tegak lurus searah dengan pergerakan wahana. Jika terjadi kesalahan *roll*, dasar laut yang sebenarnya datar akan terlihat miring ketika kapal melintasi lajur survei yang sama dengan arah berlawanan. Kalibrasi ini mengoreksi kesalahan akibat pergerakan rotasi kapal pada

sumbu X. Proses kalibrasi *roll* cukup dengan satu lajur perum pada dasar laut dengan kondisi yang datar. Kecepatan wahana sama dan dilakukan dalam arah yang berlawanan (Muljawan dkk., 2020).



Gambar 2.9 Kalibrasi *Roll* (Hoy & Kissinger, 2010)



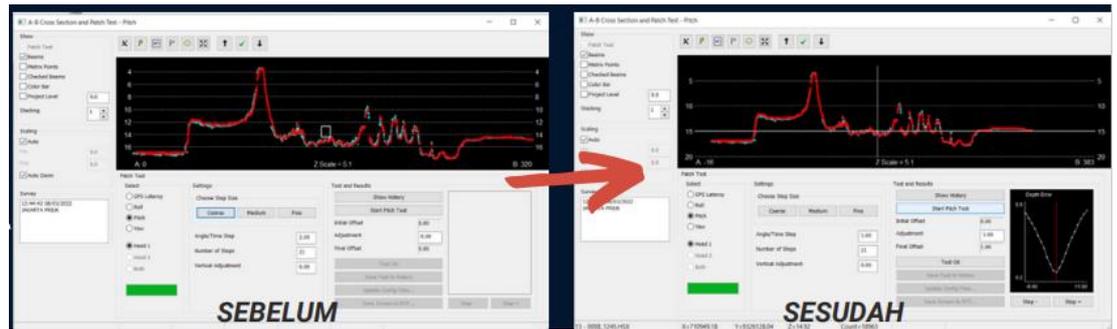
Gambar 2.10 Hasil Kalibrasi *Roll* Sebelum dan Sesudah

### 3) Kalibrasi *Pitch*

Kalibrasi *pitch* merupakan proses pengoreksian kesalahan akibat pergerakan rotasi wahana atau kemiringan pemasangan transduser pada sumbu Y atau dalam artian gerak depan atau belakang wahana. Apabila terjadi kesalahan *pitch*, fitur-fitur atau objek-objek akan terlihat bergeser sepanjang lintasan saat melakukan pemindaian *Multibeam Echosounder* di tengah sapuan (*coverage*). Prosedur kalibrasi *pitch* yaitu menentukan lajur perum yang memiliki kontur kemiringan atau *slope* curam atau terdapat objek terdefinisi di dasar laut. Dalam proses kalibrasi ini menggunakan satu lajur baris perum yang disurvei dengan arah yang berlawanan dan kecepatannya sama. Pada umumnya, semakin curam lereng yang digunakan untuk melakukan kalibrasi maka semakin akurat penentuan kesalahan *pitch* (Muljawan dkk., 2020).



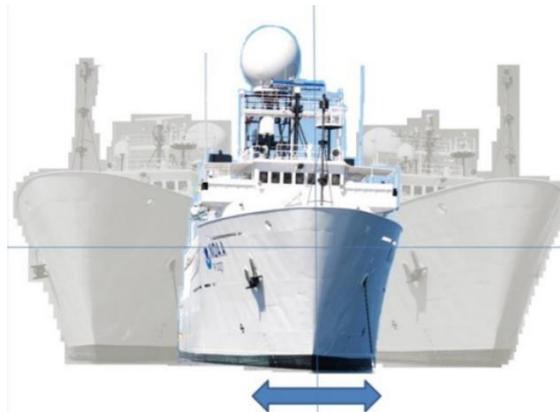
Gambar 2.11 Kalibrasi *Pitch* (Hoy & Kissinger, 2010)



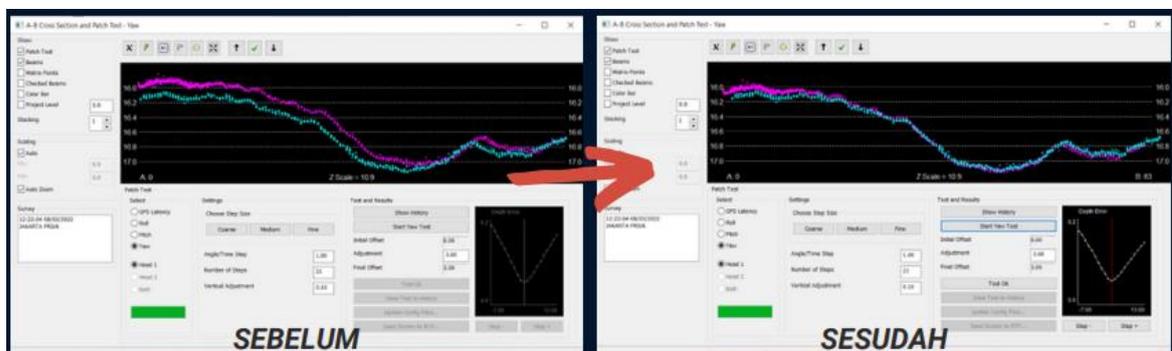
Gambar 2.12 Hasil Kalibrasi *Pitch* Sebelum dan Sesudah

#### 4) Kalibrasi *Yaw* atau *Heading*

Kalibrasi *yaw* dapat disebut kalibrasi *gyro* yang merupakan kalibrasi untuk pengoreksian kesalahan yang disebabkan rotasi wahana terhadap sumbu Z atau *heading* wahana selama survei berlangsung. Kesalahan *yaw* mengakibatkan objek-objek di dasar laut terlihat berpindah sepanjang jalur survei yang berada persis di tengah cakupan MBES. Kalibrasi ini diterapkan untuk menentukan ketidaksejajaran sudut antara orientasi kepala sonar *multibeam* dengan antera GPS. Kalibrasi ini dilakukan dengan dua lajur perum berbeda dan jarak antara lajur harus bertampalan (*overlap*) di tengah jalur dengan kecepatan yang sama. Pada umumnya, semakin curam lereng maka akan semakin akurat penentuan kesalahan *yaw* (Muljawan dkk., 2020).



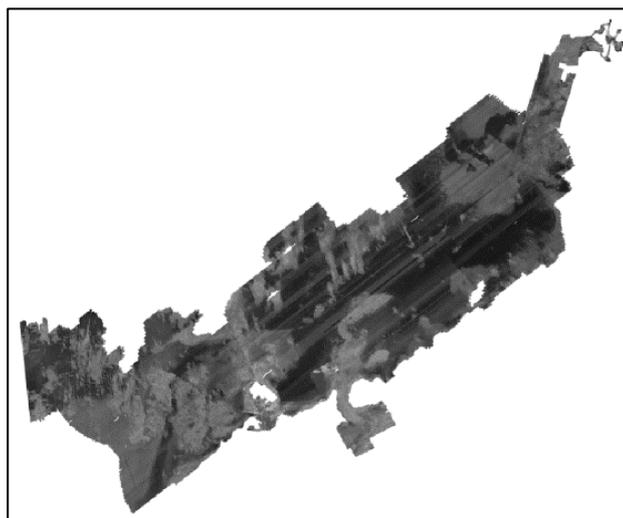
Gambar 2.13 Kalibrasi *Yaw* (Hoy & Kissinger, 2010)



Gambar 2.14 Hasil Kalibrasi *Yaw* Sebelum dan Sesudah

## 2.5 Mosaik *Backscatter*

*Backscatter* merupakan intensitas akustik yang dipantulkan dari dasar perairan menggunakan fungsi respon sudut pancaran (*angular response*) (Adi dkk., 2016). Perhitungan *backscatter* dilakukan dengan mengukur jumlah suara yang dipantulkan oleh dasar perairan dan diterima kembali oleh *receiver* (Kennedy dkk., 2005). Nilai intensitas atau amplitudo yang dihasilkan dari pemrosesan *backscatter* pada setiap titik di dasar laut kemudian akan digabungkan menjadi data citra (mosaik) *backscatter*. Mosaik *backscatter* adalah gambar berwarna keabu-abuan (*gray-scale*) yang tergeoreferensi serta merepresentasikan intensitas akustik yang diterima dari dasar laut, biasanya tingkat intensitas yang berbeda menunjukkan tipe dasar laut yang berbeda (Le Bas & Huvenne, 2009).



Gambar 2.15 Mosaik *Backscatter* (Ackerman dkk., 2015)

Pemetaan dasar laut menggunakan data mosaik *backscatter* memiliki kekurangan serta kelebihan. Terdapat 4 kelebihan yang didapatkan yaitu pertama penggunaan mosaik *backscatter* untuk pemetaan dasar laut memungkinkan identifikasi dan pemetaan kenampakan bentuk dasar laut atau objek di dasar laut. Dengan menganalisis pola serta intensitas *backscatter*, dapat diidentifikasi berbagai jenis sedimen seperti pasir, lumpur, karang atau batu. Kelebihan kedua yaitu mosaik *backscatter* memberikan informasi yang berharga tentang struktur dan komposisi dasar laut. Dengan memahami karakteristik *backscatter*, dapat diketahui lebih lanjut tentang jenis sedimen (seperti pasir, kerikil, batu, atau sedimen lainnya), topografi dasar laut, dan potensi adanya objek atau kenampakan lainnya. Kelebihan ketiga yaitu dengan mengintegrasikan data mosaik *backscatter* dengan data lainnya seperti data batimetri, dapat menghasilkan pemetaan dasar laut menjadi lebih lengkap. Untuk kelebihan terakhir yaitu penggunaan mosaik *backscatter* dalam pemetaan dasar laut memiliki berbagai aplikasi yang luas. Teknik ini digunakan dalam survei geologi laut, pemantauan lingkungan, penelitian biologi laut (mengidentifikasi area penting untuk keanekaragaman hayati, habitat ikan, atau spesies tertentu), dan pemetaan perairan (Adi dkk., 2016; Akbar dkk., 2017; Febriawan dkk., 2019).

Sedangkan untuk kekurangan penggunaan mosaik *backscatter* untuk pemetaan dasar laut adalah Hasil mosaik *backscatter* memiliki resolusi spasial yang terbatas. Hal ini dapat membatasi kemampuan untuk mengidentifikasi dan membedakan objek atau kenampakan kecil di dasar laut. Kekurangan yang lain yaitu intensitas *backscatter* dapat dipengaruhi oleh faktor lain seperti kekasaran dasar laut, orientasi objek, dan magnitudo. Hal ini dapat menyebabkan

kesulitan dalam interpretasi data *backscatter* dan mengidentifikasi dengan akurat jenis sedimen atau objek di dasar laut. Untuk kekurangan yang terakhir yaitu kualitas data *backscatter* dapat dipengaruhi oleh kondisi lingkungan seperti keadaan cuaca, kejernihan air, dan keberadaan gangguan akustik. Kondisi lingkungan yang buruk dapat mengurangi kualitas data dan akurasi hasil pemetaan. Meskipun memiliki beberapa kekurangan, penggunaan mosaik *backscatter* tetap menjadi metode yang berguna dalam pemetaan dasar laut (Adi dkk., 2016; Akbar dkk., 2017; Febriawan dkk., 2019; Manik dkk., 2016).

Kekurangan dari mosaik *backscatter* untuk pemetaan dasar laut dapat diminimalisir dengan berbagai cara seperti, untuk mengatasi resolusi spasial yang terbatas adalah dengan menggunakan peralatan yang memiliki kemampuan resolusi yang lebih tinggi, seperti *Multibeam Echosounder* dengan jumlah saluran yang lebih banyak atau frekuensi yang lebih tinggi. Untuk mengurangi pengaruh faktor lain seperti kekasaran dasar laut, orientasi objek, dan magnitudo, dapat dilakukan kalibrasi. Untuk meminimalkan gangguan akustik dan memastikan kualitas data yang baik, pemetaan dasar laut dengan mosaik *backscatter* sebaiknya dilakukan pada kondisi lingkungan yang optimal. Hal ini meliputi memilih waktu yang tepat, cuaca yang baik, dan kejernihan air yang optimal. Serta yang terakhir untuk melakukan validasi lapangan terhadap hasil pemetaan mosaik *backscatter* dapat dengan melakukan survei lapangan dan pengambilan sampel sedimen yang dapat memverifikasi dan memvalidasi interpretasi hasil mosaik *backscatter* secara langsung (Adi dkk., 2016; Febriawan dkk., 2019; Manik dkk., 2016).

Sebelum membentuk mosaik *backscatter* diperlukan normalisasi statistik dari variasi sudut dikarenakan intensitas akustik yang diterima oleh dasar laut berbeda-beda sesuai dengan sudut datang (*angle of incidence*) sinyal akustik pada saat pengambilan data, sehingga variasi intensitas pada gambar hanya disebabkan oleh perubahan geografis pada tipe dasar laut saja (Hellequin dkk., 1996). Proses normalisasi berarti aspek kuantitatif dari tingkat intensitas hilang, sehingga analisis mosaik *backscatter* yang dihasilkan memerlukan beberapa bentuk interpretasi kualitatif atau pengujian lapangan (*ground-truth*) (Clarke dkk., 1997). Mosaik *backscatter* yang berisi nilai intensitas akustik dapat mencirikan karakteristik material dasar laut, sehingga objek-objek di dasar laut yang memiliki material berbeda dengan sekitarnya akan mempunyai nilai intensitas yang berbeda dengan sekitarnya (Febriawan dkk., 2019).

## **2.6 Pola Fitur Mosaik *Backscatter***

Pola fitur atau *texture feature* dari mosaik *backscatter* merupakan fitur *extraction* dari data *backscatter*. Metode yang efektif untuk menggambarkan dan menganalisis pola fitur adalah *Gray Level Cooccurrence Matrix* (GLCM) (Zhu dkk., 2021). *Gray Level Cooccurrence Matrix* adalah sebuah matriks dimana jumlah baris dan kolom sama dengan jumlah tingkah keabuan (*gray-level*),  $G$ , pada sebuah gambar mosaik *backscatter* (Raut, dkk 2016). Dalam pengertian lain, GLCM adalah metode yang digunakan untuk menganalisis tekstur dan ekstraksi fitur, yang menggunakan matriks untuk menggambarkan frekuensi kemunculan pasangan dua piksel dengan intensitas tertentu pada jarak dan arah tertentu dalam citra (Widodo dkk., 2018). GLCM telah digunakan dalam berbagai aplikasi pengolahan citra, termasuk pengenalan pola, identifikasi objek, segmentasi citra, dan analisis tekstur dalam berbagai bidang. Haralick (1973) menentukan 14 (empat belas) pola fitur yang diukur dari matriks probabilitas untuk mengekstrak karakteristik statistik tekstur dari gambar atau citra. Keempat belas pola fitur tersebut yaitu *Angular Second Moment* (ASM), *Energy*, *Correlation*, *Entropy*, *Inverse Difference Moment* (IDM), *Contrast*, *Homogeneity*, *Variance*, *Sum Average*, *Sum Variance*, *Sum Entropy*, *Sum of Squares Variance*, *Difference Variance*, dan *Difference Entropy*. Dalam penelitian ini akan digunakan 4 pola fitur yaitu *Angular Second Moment* (ASM), Energi (*Energy*), Kontras (*Contrast*), dan Korelasi (*Correlation*) (Zhu dkk., 2021).

### 2.6.1 Angular Second Moment (ASM)

Angular Second Moment (ASM) adalah suatu pola fitur yang digunakan untuk mengukur keseragaman (*uniformity*) dari sebuah gambar, dimana saat piksel sangat mirip nilai ASM akan besar (Raut dkk., 2016). Persamaan 2.2 di bawah ini menjelaskan pola fitur ASM.

$$f_1 = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} p_{d,\theta}(i,j)^2 \quad (2.2)$$

dimana,

$N_g$  = jumlah tingkat abu-abu yang berbeda dalam gambar

$p$  = elemen matriks

$d$  = jarak

$\theta$  = sudut

$i$  = intensitas piksel referensi

$j$  = intensitas piksel tetangga

Pada Persamaan 2.2, dapat dilihat bahwa terdapat penjumlahan kuadrat dari elemen-elemen *co-occurrence matrix*. Operasi kuadrat seperti ini adalah operasi matematis murni dan tidak memerlukan satuan. Oleh karena itu, pada ASM tidak mempunyai satuan. Fitur mosaik *backscatter* ASM dalam klasifikasi jenis sedimen dasar laut dengan metode *Dense Neural Network* digunakan untuk menggambarkan kekuatan pantulan gelombang suara dari dasar laut. Dengan menganalisis pola kekuatan pantulan ini, DNN dapat mempelajari dan mengidentifikasi pola karakteristik yang berkaitan dengan jenis sedimen dasar laut. Nilai ASM yang tinggi menunjukkan adanya sedimen yang homogen atau seragam.

### 2.6.2 Kontras (Contrast)

Pola fitur kontras adalah ukuran variasi intensitas atau tingkat abu-abu (*gray-level*) antara piksel referensi dan tetangganya. Dalam persepsi visual dunia nyata, kontras ditentukan oleh perbedaan warna dan kecerahan objek dan objek lain dalam satu bidang pandang yang sama. Semakin tinggi nilai kontras, semakin besar perbedaan intensitas antara piksel-piksel dalam gambar (Raut dkk., 2016). Persamaan 2.3 di bawah ini menjelaskan pola fitur kontras.

$$f_2 = \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} p_{d,\theta}(i,j) \right\}, \text{ dimana } n = |i - j| \quad (2.3)$$

dimana,

$N_g$  = jumlah tingkat abu-abu yang berbeda dalam gambar

$p$  = elemen matriks

$d$  = jarak

$\theta$  = sudut

$i$  = intensitas piksel referensi

$j$  = intensitas piksel tetangga

$n$  = selisih intensitas piksel referensi dengan intensitas piksel tetangga

Pada Persamaan 2.3, dimana seluruh rumus hanya terdiri dari operasi matematis murni dan tidak memerlukan satuan. Oleh karena itu, pada kontras tidak mempunyai satuan. Fitur mosaik *backscatter* kontras dalam klasifikasi jenis sedimen dasar laut dengan metode *Dense Neural Network* digunakan untuk membedakan kecerahan atau kontras dalam citra *backscatter*. Perbedaan kecerahan menjadi indikator yang penting dalam mengklasifikasikan jenis sedimen dasar laut, karena sedimen yang berbeda cenderung memiliki karakteristik

kecerahan yang berbeda. Nilai kontras yang tinggi menunjukkan adanya batas yang jelas antara berbagai jenis sedimen.

### 2.6.3 Korelasi (*Correlation*)

Pola fitur korelasi adalah fitur yang menunjukkan ketergantungan linier dari nilai tingkat abu-abu (*gray-level*) dalam *co-occurrence matrix*. Hal ini menunjukkan bagaimana piksel referensi terkait dengan tetangganya, nol tidak terkait, satu sangat terkait (Raut dkk., 2016). Persamaan 2.4 di bawah ini menjelaskan pola fitur korelasi.

$$f_5 = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} p_{d,\theta}(i,j) \frac{(i-\mu_x)(j-\mu_y)}{\sigma_x\sigma_y} \quad (2.4)$$

dimana,

$N_g$  = jumlah tingkat abu-abu yang berbeda dalam gambar

$p$  = elemen matriks

$d$  = jarak

$\theta$  = sudut

$i$  = intensitas piksel referensi

$j$  = intensitas piksel tetangga

$\mu_x$  = rata-rata x

$\mu_y$  = rata-rata y

$\sigma_x$  = standar deviasi x

$\sigma_y$  = standar deviasi y

Pada Persamaan 2.4, dimana seluruh rumus hanya terdiri dari operasi matematis murni dan tidak memerlukan satuan. Oleh karena itu, pada korelasi tidak mempunyai satuan. Fitur mosaik *backscatter* korelasi dalam klasifikasi jenis sedimen dasar laut dengan metode *Dense Neural Network* digunakan untuk memperhatikan pola hubungan spasial antara piksel-piksel dalam citra *backscatter*. Dalam citra *backscatter*, pola korelasi ini dapat mengungkapkan struktur dan tekstur yang berkaitan dengan jenis sedimen dasar laut.

### 2.6.4 Energi (*Energy*)

Pola fitur energi mempunyai arti yang sama dengan *Angular Second Moment* (ASM). Semakin homogen gambar tersebut, nilai energi lebih besar. Saat energi sama dengan satu, dipercaya bahwa gambar tersebut adalah gambar konstan (Raut dkk., 2016). Persamaan 2.5 di bawah ini menjelaskan pola fitur energi.

$$Energi = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_d^2(i,j) \quad (2.5)$$

dimana,

$N_g$  = jumlah tingkat abu-abu yang berbeda dalam gambar

$p$  = elemen matriks

$d$  = jarak

$i$  = intensitas piksel referensi

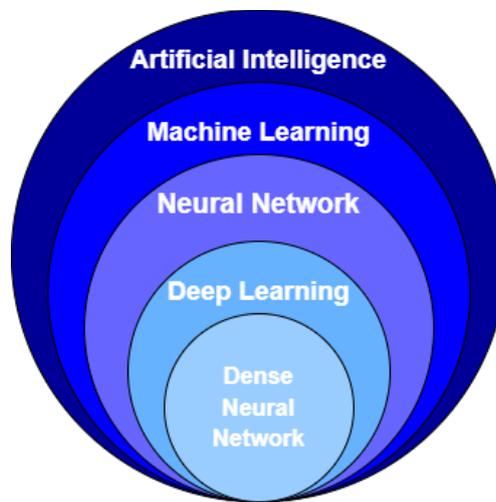
$j$  = intensitas piksel tetangga

Pada Persamaan 2.5, dimana seluruh rumus hanya terdiri dari operasi matematis murni dan tidak memerlukan satuan. Oleh karena itu, pada energi tidak mempunyai satuan. Fitur mosaik *backscatter* energi dalam klasifikasi jenis sedimen dasar laut dengan metode *Dense Neural Network* digunakan untuk mencerminkan kekuatan pantulan gelombang suara dari

dasar laut. Energi pantulan dapat berhubungan dengan karakteristik fisik sedimen, seperti kekasaran permukaan.

## 2.7 Dense Neural Network

*Dense Neural Network* (DNN) atau *Fully-connected Neural Network* adalah bentuk *Neural Network* yang paling sederhana (Amherd & Rodriguez, 2021). *Neural Network* adalah suatu usaha yang dilakukan untuk meniru otak manusia. *Neural Network* terinspirasi oleh struktur dan fungsi jaringan saraf biologis. *Neural Network* merupakan bagian dari *Artificial Intelligence* (AI). Jika diurutkan AI menempati urutan paling luar, selanjutnya terdapat *Machine Learning* (ML) dan *Neural Network* (NN). Semua pembelajaran ini (*Artificial Intelligence*, *Machine Learning*, *Neural Network*, dan *Dense Neural Network*) memiliki pendekatan, metode, teknik, dan aplikasi yang berbeda. Berikut ini merupakan gambar urutan hierarki dari AI hingga DNN.



Gambar 2.16 Urutan Hierarki DNN (dengan modifikasi Amherd & Rodriguez, 2021)

*Artificial Intelligence* atau kecerdasan buatan adalah teknik yang dilakukan untuk meniru kecerdasan manusia. Teknologi AI ini dapat membuat keputusan dengan cara menganalisis serta menggunakan data yang tersedia di dalam sistem (Lubis, 2021). Pengembangan kecerdasan buatan dimulai sejak komputer modern muncul pada tahun 1940 dan 1950. AI merujuk pada alat atau komputer yang dapat mengeksekusi tugas-tugas yang biasanya dilakukan oleh manusia. Secara teknis, kecerdasan buatan merupakan alat yang digunakan. Namun, secara teknologi, AI memiliki cakupan yang lebih luas. Permasalahan dalam bidang kecerdasan buatan ini saling terkait dengan kondisi sosial budaya masyarakat setempat dan kebijakan/strategi dari institusi pemerintahan serta industri telekomunikasi terkait. Tujuan utama AI adalah untuk mengembangkan sistem yang dapat memahami, menginterpretasikan, dan merespons data dan lingkungan dengan cara yang cerdas dan otonom. Tujuan lain dari penggunaan AI adalah untuk menciptakan komputer atau mesin yang mampu belajar, merencanakan, beradaptasi, dan menyelesaikan tugas secara mandiri. Keunggulan utama AI adalah kemampuannya untuk menganalisis dan memproses jumlah data yang besar dengan cepat, mengenali pola yang kompleks, dan membuat keputusan berdasarkan informasi yang tersedia. Aplikasi AI dapat ditemukan dalam berbagai bidang, termasuk otomatisasi proses industri, sistem cerdas, robotika, kendaraan otonom, asisten virtual, pengenalan wicara, penalaran medis, analisis data, dan banyak lagi.

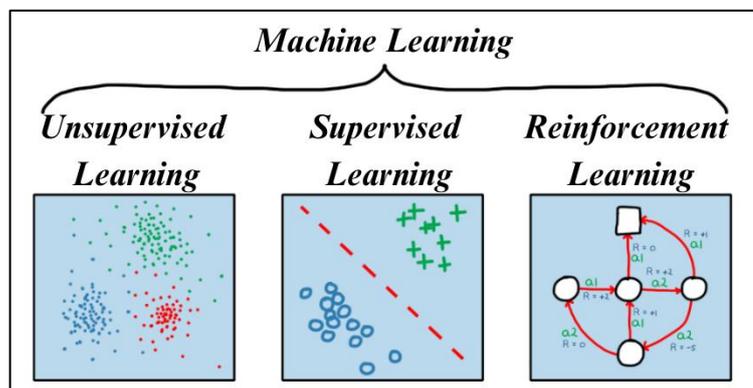
Menurut John Mc Carthy (1956), AI adalah sistem yang digunakan untuk mengetahui dan memodelkan proses-proses berpikir manusia dan mendesain mesin agar dapat menirukan

perilaku manusia. Beberapa macam bidang yang menggunakan kecerdasan buatan antara lain sistem pakar, permainan komputer, logika *fuzzy*, jaringan saraf tiruan dan robotika. Terdapat 3 macam jenis AI yaitu *Artificial Narrow Intelligence* (ANI), *Artificial General Intelligence* (AGI), dan *Artificial Super Intelligence* (ASI). ANI atau dikenal dengan *weak AI* adalah jenis AI yang paling sederhana. ANI mengacu pada sistem AI yang hanya dapat melakukan tugas tertentu secara mandiri menggunakan kemampuan seperti manusia. AGI atau dikenal dengan *strong AI* adalah sistem yang memiliki kecerdasan tingkat manusia di seluruh domain dan sistem ini sama multifungsional dengan manusia. AGI dapat berpikir, memahami, berdiskusi, dan bertindak seperti manusia dalam situasi apapun. Yang terakhir ASI yaitu sistem yang melebihi kemampuan manusia dalam segala aspek. ASI mampu menciptakan ide yang mustahil dipahami oleh manusia dalam bidang apapun (Amherd & Rodriguez, 2021).



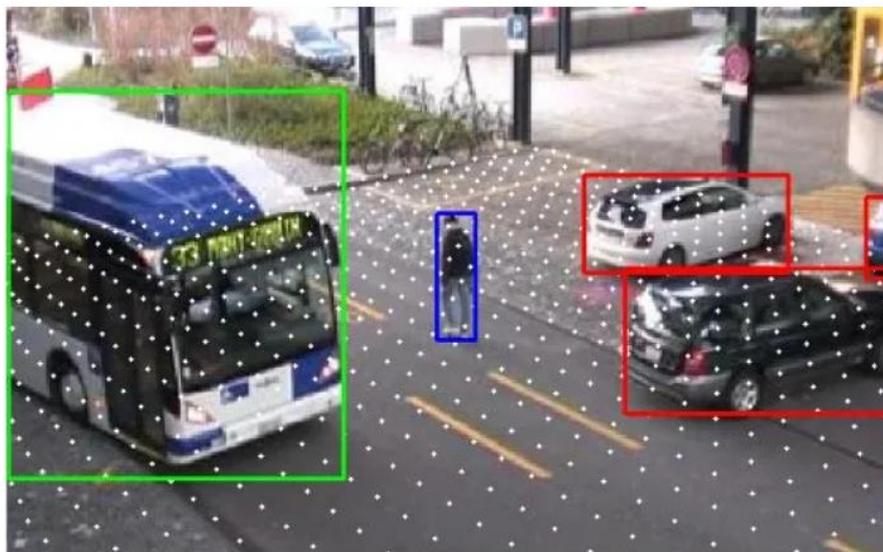
Gambar 2.17 Penerapan Artificial Intelligence

Salah satu metode yang dikembangkan dari *Artificial Intelligence* adalah *Machine Learning* (ML). *Machine Learning* adalah teknik yang paling populer karena banyak digunakan untuk menggantikan atau menirukan perilaku manusia untuk menyelesaikan masalah. Terdapat dua pengaplikasian utama dari ML yaitu klasifikasi dan prediksi. Klasifikasi adalah metode dalam ML yang digunakan untuk memilah atau mengklasifikasikan obyek berdasarkan ciri tertentu sebagaimana manusia mencoba membedakan benda satu dengan benda yang lainnya. Sedangkan prediksi adalah metode dalam ML yang digunakan untuk menerka keluaran dari suatu data masukan (*input*) berdasarkan data yang sudah dipelajari dalam *training*. Ciri khas dari ML adalah adanya proses pelatihan atau *training*. Oleh karena itu di dalam ML membutuhkan data untuk dipelajari yang disebut dengan *data training*. ML memiliki perbedaan dengan kode komputer biasa yang dikembangkan oleh pengembang perangkat lunak untuk menghasilkan *output* berdasarkan *input* yang diberikan. Terdapat beberapa jenis dari ML termasuk *supervised learning*, *unsupervised learning*, dan *reinforcement learning*.



Gambar 2.18 Jenis Machine Learning (MathWorks, 2023)

Pada ML, data digunakan untuk menghasilkan model statistik (model ML) yang berfungsi untuk menghasilkan *output* yang akurat berdasarkan pola yang dikenali dari contoh *input* sebelumnya, dengan menggunakan teknik pengawasan. Akurasi model ML tergantung pada kualitas dan jumlah data historis yang ada. Untuk menemukan fungsi optimal yang dapat memprediksi hasil berdasarkan input yang diberikan, Model ML dapat menganalisis masalah dengan dimensi tinggi yang melibatkan miliaran contoh data. Contoh pengaplikasian ML yaitu filter otomatis dan *reminder* pada *handphone*. Perangkat tersebut tidak mampu belajar sendiri kecuali jika diberikan kode yang menginstruksikan mesin untuk memecahkan masalah yang ada. Namun, mesin tidak dapat memikirkan atau mencerna makna dari kode operasional yang telah diinputkan. Secara sederhana, perangkat diberikan contoh uji coba yang luas untuk melaksanakan tugas yang kompleks. Saat diuji, mesin akan melalui proses pembelajaran untuk mengenali dan menyimpan kasus-kasus yang diberikan dalam memori, sehingga dengan banyak percobaan, mesin akan mengembangkan keahlian dalam pengenalan pola, bentuk, wajah, dan sejenisnya. Dalam perkembangannya, ML terus mengalami kemajuan dengan penggunaan algoritma dan teknik yang lebih canggih, memungkinkan sistem untuk belajar dan beradaptasi dengan data yang semakin besar dan kompleks (Ahmad, 2017).



Gambar 2.19 Aplikasi *Machine Learning* (Singh, 2018)

Dalam ML terdapat teknik yang menirukan saraf manusia yang merupakan bagian fundamental dari otak yaitu *Neural Network* (NN). NN dapat dilatih dengan menggunakan data *training*. Semakin banyak data *training* yang digunakan maka akan semakin baik kerja NN tersebut. Namun, kemampuan NN terbatas pada jumlah lapisan, dimana semakin banyak lapisan (*layer*) maka semakin tinggi kapasitas NN. hal ini juga akan menambah semakin banyaknya jumlah data *training*. NN digunakan dalam aplikasi khusus, seperti pengenalan pola atau pengelompokan data melalui proses pembelajaran. Jaringan ini terdiri dari banyak elemen yang saling terhubung yang disebut *neuron*, yang bekerja bersama dalam memecahkan masalah tertentu. Penerapan jaringan syaraf tiruan ini berfokus pada pemrosesan sinyal digital dan pengenalan pola. Sebagai contoh, sejak tahun 2000, jaringan ini telah mampu mengenali tulisan tangan dalam pembayaran cek, menganalisis data, mengenali wajah, dan melakukan prediksi cuaca. Jaringan saraf tiruan (*Neural Network/NN*) dapat dilatih dengan menggunakan data *training*. Semakin banyak data *training* yang digunakan, maka unjuk kerja dari NN tersebut akan semakin baik. Namun, kemampuan NN juga terbatas oleh jumlah lapisan yang digunakan. Semakin banyak lapisan yang digunakan, kapasitas NN akan semakin tinggi. Namun,

penambahan lapisan juga menyebabkan kekurangan yaitu memerlukan lebih banyak iterasi atau *training* untuk mencapai hasil yang optimal. Untuk mengatasi masalah ini, dikembangkanlah teknik *Deep Learning* (Ahmad, 2017).

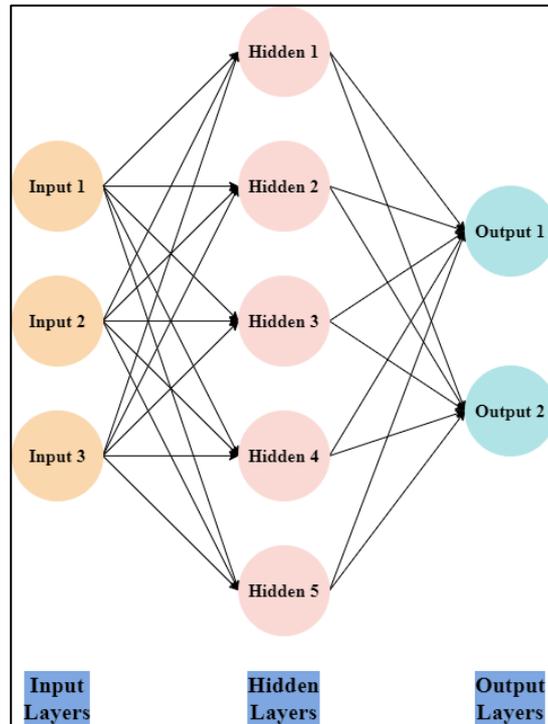
*Deep Learning* (DL) merupakan cabang dari *Machine Learning* yang menggunakan *neural network* untuk menangani permasalahan dengan *dataset* yang besar. Teknik *Deep Learning* menyediakan arsitektur yang sangat kuat untuk *Supervised Learning*. Dengan menambahkan lebih banyak lapisan, model pembelajaran tersebut dapat menggambarkan data citra berlabel dengan lebih baik. Penerapan konsep jaringan syaraf tiruan yang dalam (dengan banyak lapisan) telah mengubah cara algoritma *Machine Learning* beroperasi, memungkinkan komputer untuk belajar dengan kecepatan, akurasi, dan skala yang besar. Prinsip ini telah berkembang pesat, menjadikan *Deep Learning* semakin populer di kalangan komunitas riset dan industri untuk memecahkan berbagai masalah data besar, seperti *Computer Vision*, *Speech Recognition*, dan *Natural Language Processing*. Salah satu fitur kunci dari *Deep Learning* adalah *Feature Engineering*, yang memungkinkan ekstraksi pola yang relevan dari data, sehingga memudahkan model untuk membedakan berbagai kelas. *Feature Engineering* menjadi teknik yang sangat penting untuk mencapai hasil yang baik dalam tugas prediksi. Namun, teknik ini sulit untuk dipelajari dan dikuasai, karena berbagai kumpulan data dan jenis data memerlukan pendekatan teknik yang berbeda pula (Nurhikmat, 2018).

Salah satu subbidang dari *Machine Learning* disebut *Deep Learning*. Bidang ini mengatasi pengembangan Jaringan Saraf Tiruan (*Artificial Neural Networks/ANN*), yang strukturnya terinspirasi secara kuat oleh otak manusia. Tugas-tugas yang lebih kompleks seringkali memerlukan jaringan yang lebih besar dengan lebih banyak lapisan berikutnya. Jika sebuah ANN terdiri dari lebih dari tiga lapisan, kita menyebutnya sebagai *Deep Neural Network*. Terdapat puluhan arsitektur berbeda dari ANN yang sangat unggul dalam jenis masalah tertentu. Arsitektur-arsitektur tersebut seperti bentuk paling sederhana, yaitu *Dense Neural Network* (DNN) (Amherd & Rodriguez, 2021).

Di dalam otak manusia terdapat jutaan unit pengolahan kecil yang disebut neuron. *Neuron-neuron* saling terhubung satu sama lain melalui koneksi atau jaringan *neuron* (Shukla dkk, 2010). Dalam lapisan *neuron*, *neuron-neuron* dikelompokkan dalam satu lapisan atau disebut *neuron layer*. Kemudian, *neuron-neuron* pada lapisan tersebut dihubungkan dengan lapisan sebelum dan sesudahnya, kecuali lapisan *input* dan *output*. Informasi dari langkah input awal akan ditransmisikan dari lapisan ke lapisan hingga mencapai lapisan *output*. Lapisan ini biasa disebut sebagai *hidden layer* (lapisan tersembunyi). Biasanya, setiap *neuron* yang berada pada lapisan yang sama memiliki keadaan yang serupa. Sehingga, pada setiap lapisan yang sama, setiap *neuron* akan memiliki fungsi aktivasi yang serupa. Koneksi antara lapisan dan *neuron* harus selalu terhubung. Faktor yang paling berperan dalam menentukan perilaku suatu *neuron* terletak pada pola bobotnya dan fungsi aktivasi yang digunakan. Pada DNN terdiri dari ribuan hingga jutaan *node* yang diatur pada lapisan yang berbeda. Setiap *node* terhubung dengan *node* lain di lapisan sebelum atau sesudahnya. Setiap *neuron* dalam DNN menggunakan fungsi aktivasi untuk menghitung *output*-nya berdasarkan bobot yang ditetapkan pada setiap koneksi. Jadi dapat dikatakan bahwa ada banyak koneksi atau jaringan lebih dari jumlah *node*. Hal ini menyebabkan lapisan individunya disebut *Dense Layers* atau *Fully-connected Layers* (FC) (Amherd & Rodriguez, 2021).

Pada *Dense Neural Network* terdapat istilah *input layers*, *hidden layers*, dan *output layers*. *Input layers* adalah lapisan pertama dari NN yang nantinya akan berisi input data untuk proses klasifikasi. Lapisan berikutnya adalah *hidden layers* yaitu lapisan tengah antara *input* dan *output layers* yang memiliki jumlah biasanya tidak lebih dari 2 layer untuk *Dense Neural Network*. Terakhir ada *output layers* yaitu lapisan akhir yang merupakan hasil dari *neural network*. DNN telah terbukti sangat efektif dalam berbagai tugas ML seperti klasifikasi gambar,

pengenalan suara, pemrosesan bahasa alami, dan prediksi data. Dengan memiliki banyak lapisan dan banyak parameter yang dapat disesuaikan, DNN dapat menangkap representasi yang semakin kompleks dari data, sehingga memungkinkan pembelajaran dan pemodelan yang lebih mendalam (Amherd & Rodriguez, 2021).



Gambar 2.20 Bentuk *Dense Neural Network*

### 2.7.1 Data Training dan Data Testing

Data *training* adalah sekumpulan data yang memiliki atribut label atau kelas yang digunakan oleh mesin untuk mempelajari karakteristik dari kumpulan data tersebut sehingga dapat menghasilkan sebuah pola atau model data. Di sisi lain, data *testing* adalah sekumpulan data yang juga memiliki atribut label atau kelas yang digunakan untuk menguji seberapa akurat pola atau model dalam mengklasifikasikan data *testing* tersebut. Saat melakukan proses pengujian model, atribut label dari data *testing* tidak ditampilkan selama proses klasifikasi berlangsung, namun akan digunakan nantinya untuk membandingkan hasil klasifikasi dengan label yang sebenarnya. Hal ini digunakan sebagai tolak ukur untuk menilai seberapa akurat atau tepat model tersebut dalam melakukan klasifikasi. Pembagian presentasi jumlah data *training* dan data *testing* dari sebuah dataset menjadi salah satu faktor penentu besaran nilai akurasi. Sehingga kesalahan menentukan komposisi antara kedua jenis data tersebut akan mempengaruhi nilai akurasi yang diperoleh. Data *training* yang baik harus mencakup variasi yang mencakup kasus-kasus yang mungkin dihadapi oleh model dalam penggunaan praktisnya. Data *training* yang representatif dan beragam dapat membantu model untuk belajar secara lebih umum dan dapat diandalkan dalam mengatasi berbagai situasi. Sedangkan penggunaan data *testing* penting untuk menilai kinerja model secara obyektif dan memahami sejauh mana model dapat menggeneralisasi pada data baru. Dengan melakukan pengujian yang baik, akan mendapatkan wawasan tentang kehandalan dan kemampuan prediksi model sebelum menerapkannya dalam lingkungan produksi atau penggunaan praktis (Musu & Ibrahim, 2021)



Gambar 2.21 Pembagian Data *Training* dan Data *Testing* (JavaTPoint, 2023)

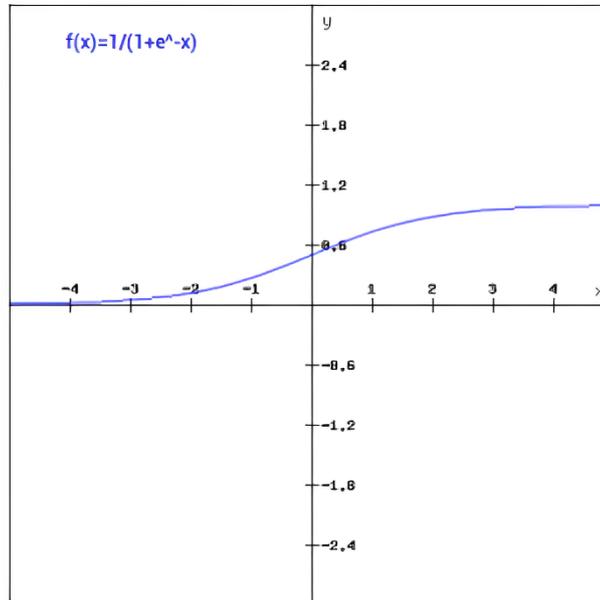
### 2.7.2 Activation Function

Fungsi aktivasi (*Activation Function*) adalah sebuah fungsi pada model yang bertugas untuk mengolah jumlah data masukan menjadi data keluaran (Pamungkas & Alam, 2022). Definisi lain dari fungsi aktivasi yaitu fungsi yang digunakan pada jaringan saraf untuk mengaktifkan atau tidak mengaktifkan *neuron* (Julpan dkk., 2015). Fungsi aktivasi sangat umum digunakan dalam *Neural Network*. Tujuan utama penggunaan fungsi aktivasi adalah agar *Neural Network* mengenali data yang non-linear, karena *output* yang dihasilkan dari *Neural Network* jarang bersifat linear. Jika tidak menggunakan fungsi aktivasi dalam *Neural Network*, maka keluaran sinyal akan menjadi fungsi linear sederhana yang hanya merupakan polinomial derajat satu. Meskipun persamaan linear sederhana dan mudah dipecahkan, kompleksitasnya terbatas dan tidak mampu mempelajari dan mengenali pemetaan yang kompleks dari data. *Neural Network* tanpa fungsi aktivasi berperan sebagai model regresi linear dengan kinerja dan kapabilitas yang terbatas sebagian besar waktu. Idealnya, *Neural Network* tidak hanya belajar dan menghitung fungsi linear, tetapi juga dapat melakukan tugas yang lebih kompleks seperti memodelkan jenis data yang rumit seperti gambar, video, audio, ucapan, teks, dan lainnya (Sharma dkk., 2020). Setiap *neuron* dalam *Neural Network* memiliki fungsi aktivasi yang mengubah jumlah bobot dan input yang diterimanya menjadi output yang kemudian diteruskan ke *neuron* berikutnya. Fungsi aktivasi ini memperkenalkan ambang batas atau aturan tertentu untuk menentukan apakah *neuron* tersebut akan mengaktifkan atau menghambat sinyal yang diteruskan. Terdapat banyak jenis fungsi aktivasi pada *Neural Network*, namun yang digunakan dalam penelitian ini fungsi aktivasi Sigmoid dan ReLU.

Fungsi aktivasi sigmoid adalah fungsi aktivasi yang mempunyai rentang nilai antara nol hingga satu (Amherd & Rodriguez, 2021). Fungsi aktivasi ini paling sering dipakai dalam pengolahan data. Selain itu, fungsi sigmoid tidak simetris terhadap nol yang berarti bahwa tanda dari semua nilai keluaran *neuron* yang sama (Sharma dkk., 2020). Masalah ini dapat diperbaiki dengan menskalakan fungsi sigmoid. Fungsi sigmoid memiliki beberapa kelemahan. Salah satunya adalah masalah gradien yang melemah ketika nilai input berada pada ekstrem, yang dapat menghambat pelatihan jaringan saraf yang lebih dalam. Selain itu, sigmoid juga rentan terhadap saturasi, yang berarti ketika input yang besar diberikan, *output*-nya akan mendekati nol atau satu dan gradiennya menjadi sangat kecil, menyebabkan masalah pelatihan yang lambat. Fungsi sigmoid umumnya digunakan pada lapisan tersembunyi jaringan saraf tiruan untuk memperkenalkan *non-linearitas*. *Non-linearitas* ini memungkinkan jaringan saraf untuk mempelajari pola-pola yang lebih kompleks dalam data. Selain itu, fungsi sigmoid juga digunakan dalam jaringan saraf dengan satu lapisan *output* untuk menghasilkan probabilitas sebagai *output*-nya. Pada Persamaan 2.6 disajikan secara matematis fungsi aktivasi sigmoid (Amherd & Rodriguez, 2021).

$$\sigma(z) = \frac{1}{1+e^{-z}} \quad (2.6)$$

dimana,  
 $\sigma$  = fungsi sigmoid  
 $z$  = input

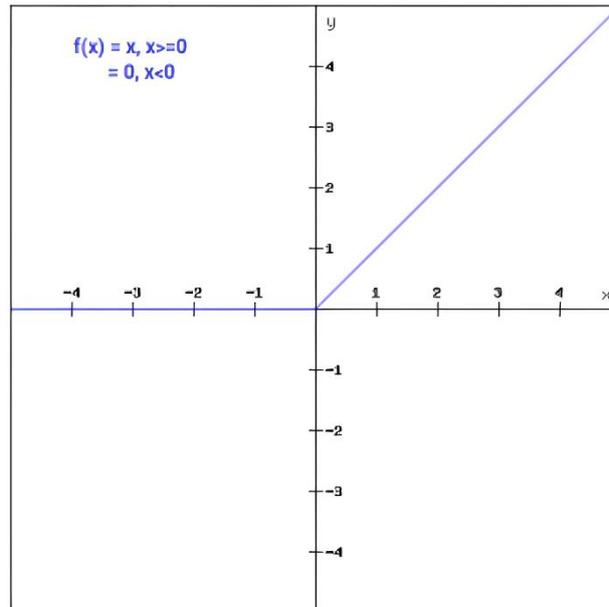


Gambar 2.22 Fungsi Aktivasi Sigmoid (Sharma dkk., 2020)

Selain fungsi aktivasi sigmoid, fungsi aktivasi lain yang sering dipakai dalam pengolahan *Neural Network* adalah fungsi aktivasi ReLU. Fungsi aktivasi ReLU (*Rectified Linier Unit*) merupakan fungsi aktivasi yang mudah dihitung dan membantu mengatasi masalah gradien yang hilang, dimana gradien kerugian menjadi sangat kecil (Amherd & Rodriguez, 2021). Keuntungan menggunakan fungsi aktivasi ReLU adalah bahwa semua *neuron* tidak diaktifkan pada waktu yang sama (Sharma dkk., 2020). Hal ini berarti *neuron* akan dinokaktifkan hanya ketika *output* dari transformasi linier bernilai nol. Fungsi aktivasi ReLU terutama digunakan dalam jaringan saraf untuk tugas-tugas seperti pengenalan gambar, klasifikasi, dan tugas-tugas dimana fitur-fitur yang relevan dengan *output* dapat diekstraksi secara efektif. Namun, perlu diingat bahwa ReLU memiliki kelemahan yaitu bisa menyebabkan masalah *neuron* yang mati (*dead neuron*) jika sejumlah besar *input* negatif terjadi. Fungsi aktivasi ReLU lebih efisien daripada fungsi aktivasi yang lainnya karena semua *neuron* tidak diaktifkan pada saat yang sama. Dalam beberapa kasus terdapat nilai gradien yang nol, karena itu bobot dan bias tidak diperbarui selama proses *training Neural Network*. Fungsi ReLU juga relatif sederhana dalam perhitungannya dan memiliki kemampuan untuk mempercepat proses pelatihan jaringan saraf. Pada Persamaan 2.7 disajikan secara matematis fungsi aktivasi sigmoid (Amherd & Rodriguez, 2021).

$$ReLU(x) = \max(0, x) \quad (2.7)$$

dimana,  
 $ReLU$  = fungsi ReLU  
 $x$  = input



Gambar 2.23 Fungsi Aktivasi ReLU (Sharma dkk., 2020)

### 2.7.3 Hyperparameter

*Hyperparameter* adalah parameter yang ditentukan sebelum proses pelatihan model dimulai dan tetap tidak berubah selama pelatihan berlangsung. *Hyperparameter* mengendalikan karakteristik dan perilaku algoritma pembelajaran mesin, seperti jaringan saraf (*Neural Network*), yang digunakan dalam pelatihan dan pengoptimasian model. *Hyperparameter* tidak diperoleh langsung dari data, melainkan ditentukan oleh pengguna atau peneliti berdasarkan pengetahuan *domain*, kebijakan eksperimen, atau melalui teknik pencarian dan optimasi. Dalam penelitian ini digunakan *hyperparameter optimizer*, *loss function*, dan *metrics*.

*Hyperparameter* yang digunakan untuk *optimizer* adalah Adam. Adam (*Adaptive Moment Estimation*) adalah metode yang digunakan untuk menghitung tingkat pembelajaran adaptif untuk setiap parameter (Ruder, 2016). Algoritma Adam berfungsi untuk optimasi berbasis gradien orde pertama dari fungsi objektif stokastik, berdasarkan estimasi adaptif dari momen orde rendah (Kingma & Ba, 2014). Metode ini sangat mudah diimplementasikan dan efisien secara komputasi, memiliki sedikit kebutuhan memori, tidak berubah-ubah terhadap perubahan skala diagonal gradien, dan sangat cocok untuk masalah yang besar dalam hal data dan/atau parameter. Metode ini juga sesuai untuk tujuan dan masalah non-stasioner dengan gradien yang sangat bising dan/atau jarang. Untuk *hyperparameter loss function* digunakan *categorical crossentropy*. *Categorical crossentropy* adalah *loss function* yang bertujuan untuk menghitung kerugian *crossentropy* antara label dan prediksi (Keras, 2023). Penggunaan *loss function* ini ketika ada dua atau lebih kelas. Sedangkan untuk *hyperparameter metrics* digunakan *accuracy*. Penggunaan akurasi karena sesuai dengan tujuan klasifikasi yaitu untuk mengukur sejauh mana model dapat melakukan klasifikasi dengan benar.

## 2.8 Pasang Surut Air Laut

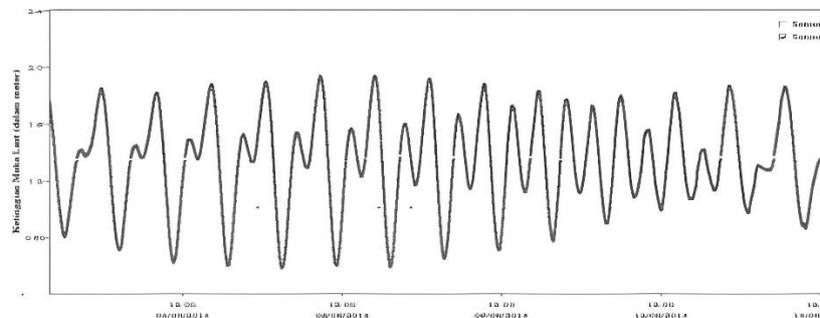
Pasang surut atau sering disingkat pasut merupakan fenomena pergerakan naik turunnya permukaan air secara berkala yang diakibatkan oleh gaya gravitasi bumi dan gaya tarik menarik dari benda-benda astronomi terutama oleh matahari dan bulan. Faktor utama yang mempengaruhi pasang surut adalah tarikan gravitasi antara Bulan, Matahari, dan Bumi. Meskipun Matahari jauh lebih besar dari Bulan, tetapi karena jaraknya yang lebih dekat, tarikan gravitasi Bulan memiliki pengaruh yang lebih besar terhadap pasang surut daripada Matahari.

Data pengukuran pasang surut dapat dimanfaatkan secara langsung maupun secara tidak langsung. Secara langsung, pemanfaatan pasang surut dapat membantu nelayan dalam menempatkan perahu-perahu mereka di pesisir pantai. Sedangkan pemanfaatan tidak langsung seperti referensi pemetaan bidang vertikal (Fitriana dkk., 2022). Selain itu pengetahuan mengenai waktu, ketinggian, dan arus pasang surut berperan dalam keperluan navigasi, pekerjaan rekayasa kelautan seperti pelabuhan, bangunan penahan gelombang, juga untuk keperluan lainnya seperti militer, penangkapan ikan, dan olahraga bahari (Suhaemi dkk., 2018).

Secara umum pasang surut di berbagai daerah dibedakan dalam empat tipe yaitu:

- Pasang Surut Harian Ganda (*Semi Diurnal Tide*)

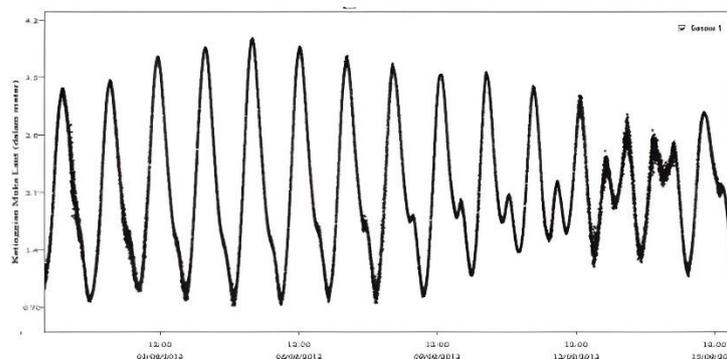
Yaitu pasang surut yang memiliki sifat dalam satu hari terjadi dua kali pasang dan dua kali surut dengan tinggi yang hampir sama dan terjadi secara berurutan serta teratur. Dalam pasang surut harian ganda, setiap siklus pasang surut terdiri dari dua pasang tinggi yang terpisah oleh dua pasang rendah. Pasang tinggi pertama biasanya kurang lebih setengah hari setelah pasang tinggi sebelumnya, dan pasang rendah pertama biasanya setengah hari setelah pasang rendah sebelumnya. Ini berarti bahwa terdapat sekitar dua pasang tinggi dan dua pasang rendah dalam periode 24 jam.



Gambar 2.24 Pola Gerak Pasut Harian Ganda (Widyantoro, 2014)

- Pasang Surut Harian Tunggal (*Diurnal Tide*)

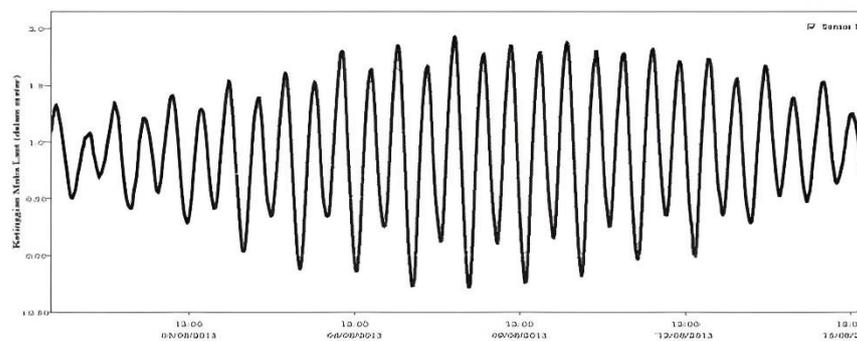
Yaitu pasang surut yang memiliki sifat dalam satu hari hanya terjadi satu pasang dan satu kali surut. Dalam pasang surut harian tunggal, setiap siklus pasang surut terdiri dari satu pasang tinggi dan satu pasang rendah. Pasang tinggi biasanya terjadi sekitar setengah hari setelah pasang rendah sebelumnya. Ini berarti bahwa terdapat satu pasang tinggi dan satu pasang rendah dalam periode 24 jam.



Gambar 2.25 Pola Gerak Pasut Harian Tunggal (Widyantoro, 2014)

- Pasang Surut Campuran Condong ke Harian Ganda (*Mixed Tide Prevailing Semi Diurnal*)

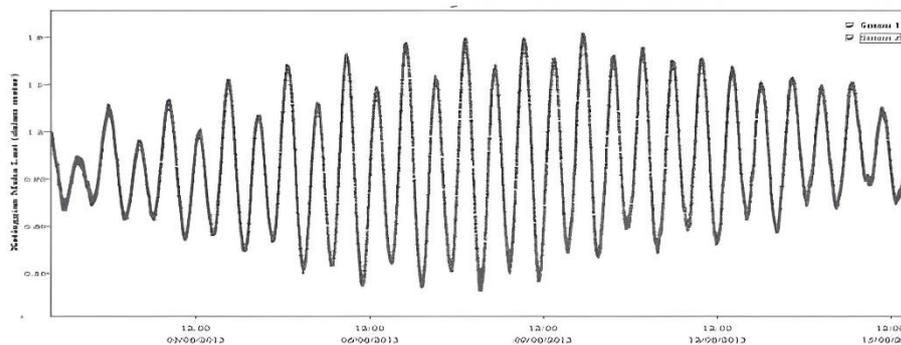
Yaitu pasang surut yang memiliki sifat dalam sehari terjadi dua kali pasang dan dua kali surut tetapi tinggi dan periodenya berbeda. Dalam pola ini, terdapat dua pasang tinggi dan dua pasang rendah yang memiliki amplitudo yang serupa dalam periode harian, mirip dengan pasang surut harian ganda. Namun, ada juga variasi tambahan yang terjadi dalam pola pasang surut, seperti perubahan amplitudo atau pola frekuensi yang tidak teratur.



Gambar 2.26 Pola Gerak Pasut Campuran Condong Harian Ganda (Widyantoro, 2014)

- Pasang Surut Campuran Condong ke Harian Tunggal (*Mixed Tide Prevailing Diurnal*)

Yaitu pasang surut yang memiliki sifat dalam sehari terjadi satu kali pasang dan satu kali surut tetapi kadang-kadang untuk sementara waktu terjadi dua kali pasang dan dua kali surut dengan tinggi dan periode yang berbeda. Dalam pola ini, terdapat satu pasang tinggi dan satu pasang rendah yang terjadi dalam periode harian, mirip dengan pasang surut harian tunggal. Namun, ada variasi tambahan dalam pola pasang surut, seperti perubahan amplitudo, pola frekuensi yang tidak teratur, atau interaksi kompleks antara siklus bulan dan faktor-faktor lainnya.



Gambar 2.27 Pola Gerak Pasut Campuran Condong Harian Tunggal (Widyantoro, 2014)

Menurut Djaja (1989), tujuan dari pengamatan pasang surut secara umum adalah sebagai berikut:

1. Untuk menentukan permukaan air laut rata-rata dan untuk keperluan survei rekayasa dengan melakukan satu sistem pengikatan terhadap bidang referensi.
2. Untuk memberikan data peramalan pasut dan arus serta mempublikasikan data peramalan pasut dan arus dalam tabel tahunan.
3. Untuk menyelidiki perubahan kedudukan air laut dan pergerakan kerak bumi.
4. Untuk menyediakan informasi yang menyangkut keadaan pasut.
5. Untuk memberikan data yang tepat guna studi muara sungai tertentu.

6. Untuk melengkapi informasi dalam penyelesaian masalah hukum yang berkaitan dengan batas-batas wilayah yang ditentukan berdasarkan pasut.

Pengamatan pasang surut pada kegiatan survei hidrografi bertujuan untuk menentukan bidang acuan kedalaman (muka air laut rerata, muka surutan) serta menentukan koreksi hasil pemeruman. Pasang surut mempengaruhi kedalaman air laut, sehingga penting untuk mengamati dan memperhitungkan pasang surut dalam pengukuran kedalaman laut. Pada saat pasang tinggi, kedalaman air lebih dalam, sementara pada saat surut rendah, kedalaman air lebih dangkal. Data pasang surut digunakan untuk mengoreksi nilai kedalaman yang diukur, sehingga menghasilkan data kedalaman yang relatif terhadap referensi tetap dan akurat.

### 2.9 *Sound Velocity Profiler*

*Sound Velocity Profiler* (SVP) adalah instrumen yang digunakan untuk mengukur kecepatan suara di dalam air dan menjadi instrumen penting untuk berbagai peralatan akustik bawah air (Zhang dkk., 2022). SVP digunakan dalam berbagai aplikasi, termasuk dalam penelitian dan pemetaan lingkungan bawah air, pengukuran suhu dan salinitas laut, studi migrasi hewan laut, navigasi bawah air, dan sistem komunikasi dan pemetaan dalam jaringan bawah air. Kecepatan suara di dalam air dipengaruhi oleh beberapa faktor seperti suhu, tekanan, salinitas, dan kedalaman. SVP digunakan dalam pengolahan data *Multibeam Echosounder*. Data hasil perekaman alat ini merupakan salah satu koreksi yang diperlukan untuk mendapatkan nilai kedalaman yang akurat (Setiadarma dkk., 2019). Ketidaksesuaian dalam mengukur kecepatan suara dapat menyebabkan ketidakakuratan dalam data yang dikumpulkan oleh *Multibeam Echosounder*. Koreksi lain yang juga dipakai untuk mendapatkan nilai kedalaman yang akurat meliputi koreksi pasang surut dan koreksi pergerakan kapal.



Gambar 2.28 *Sound Velocity Profiler* (Valeport, 2023)

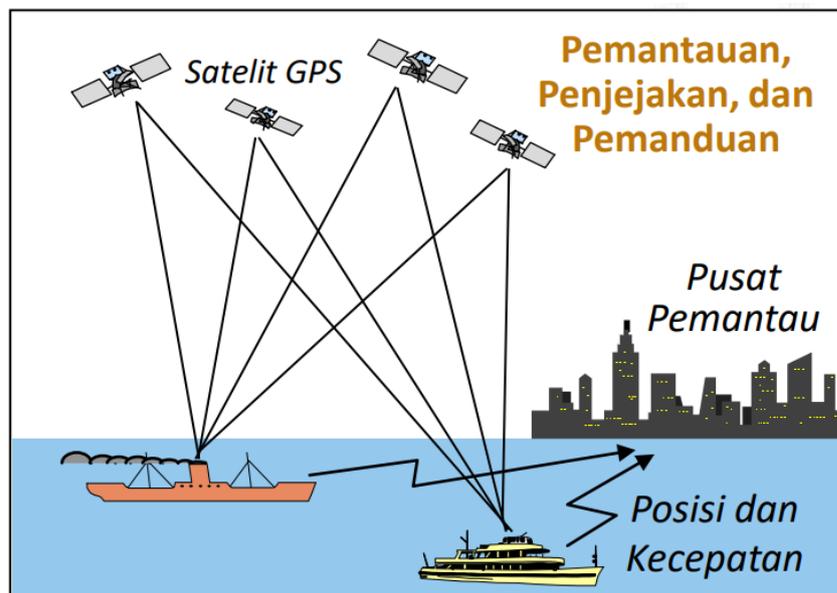
Dalam SVP umumnya terdiri dari tiga bagian yaitu unit kontrol, modul penyimpanan, dan modul pengukuran ultrasonik. Unit kontrol berfungsi untuk mengatur dan mengendalikan seluruh operasi SVP. Modul penyimpanan berfungsi untuk menyimpan data yang dihasilkan dari pengukuran kecepatan suara. Sedangkan modul pengukuran ultrasonik berfungsi untuk mengukur kecepatan suara di dalam air. Keseluruhan bagian pada SVP berukuran kecil dan tidak membutuhkan daya yang besar. Sumber daya SVP berasal dari baterai *internal* atau sumber daya eksternal. Beberapa SVP dilengkapi dengan koneksi *interface* seperti USB atau *ethernet*, yang memungkinkan SVP terhubung dengan komputer sehingga dapat mentransfer data atau mengatur pengaturan (Zhang dkk., 2022).

### 2.10 *Global Positioning System*

*Global Positioning System* (GPS) merupakan suatu sistem untuk menentukan posisi dan navigasi secara global dengan menggunakan satelit. Sistem GPS mempunyai tiga segmen yaitu satelit, pengontrol, dan penerima atau pengguna. Satelit bertugas untuk menerima dan menyimpan data yang ditransmisikan oleh stasiun-stasiun pengontrol, menyimpan dan menjaga informasi waktu dan informasi secara kontinyu. Pengontrol bertugas untuk mengontrol dan

mengendalikan serta mengecek satelit dari bumi. Sedangkan penerima bertugas untuk menerima data dari satelit dan memprosesnya untuk menentukan posisi (Rianandra dkk., 2015).

Dalam penentuan posisi menggunakan GPS untuk pemetaan laut biasanya digunakan sistem GNSS (*Global Navigation Satellite System*). GNSS merupakan sistem navigasi dan penentuan posisi baik horizontal maupun vertikal. Dalam survei hidrografi, GNSS digunakan untuk navigasi kapal, penentuan titik kontrol di pantai, penentuan garis pantai, penentuan posisi titik perum, dan penentuan posisi sensor lainnya. Penggunaan GNSS dalam survei hidrografi memiliki beberapa keuntungan seperti peningkatan presisi, efisiensi biaya, data *real-time*, dan efisien. GNSS dapat memberikan akurasi posisi yang sangat tinggi sehingga meningkatkan presisi survei untuk posisi horizontal dan vertikal serta pengukuran akustik dasar laut. Teknik *Precise Point Positioning* (PPP) pada GNSS menggunakan hanya satu instrumen *dual-frequency* sehingga menjadi solusi dalam hal efisiensi biaya untuk survei hidrografi (Dardanelli dkk., 2021). Data yang dihasilkan GNSS merupakan data *real-time*. GNSS juga dapat memberikan cara yang lebih cepat dan lebih efisien untuk mendapatkan koordinat pada survei hidrografi (Stateczny dkk., 2021).



Gambar 2.29 Penentuan Posisi dengan GNSS (Abidin, 2021)

## 2.11 EIVA

EIVA adalah perusahaan yang didirikan tahun 1978 yang menyediakan peralatan, sensor dan layanan untuk industri minyak dan gas lepas pantai. Pada tahun 1982, perusahaan ini mulai mengembangkan perangkat lunak navigasi terintegrasi NaviPac. Sejak saat itu, perusahaan ini terus berkembang menjadi perusahaan teknologi tinggi yang diakui saat ini dengan fokus utama pada perangkat lunak dan solusi sistem terintegrasi untuk digunakan dalam survei hidrografi, pemetaan dasar laut, pekerjaan konstruksi dan inspeksi lepas pantai, survei geofisika, penelitian oseanografi, dan lain-lain. Dalam proses pengembangan perangkat lunak, EIVA menggunakan alat bantu standar untuk spesifikasi dan desain, implementasi, manajemen, dan dokumentasi. Produk perangkat lunak EIVA terdiri dari perangkat lunak untuk semua aspek survei kelautan, mulai dari navigasi dan akuisisi data yang terintegrasi, hingga pasca-pemrosesan data dan produksi lembar kerja. Perangkat lunak ini mencakup: navigasi terpadu dan perencanaan misi (NaviPac), akuisisi data *multibeam* dan sonar (NaviScan), pengeditan data survei (NaviEdit), pemodelan *Digital Terrain Model* (NaviModel), dan produksi dan

pembuatan peta (NaviPlot). Dalam pengolahan data MBES digunakan dua *software* EIVA yaitu NaviEdit dan NaviModel (Herskind, 2008)

NaviEdit merupakan perangkat lunak yang bertujuan untuk mengubah secara keseluruhan pengaturan survei *raw data* dan parameter lainnya. Fokus utama NaviEdit adalah mengoreksi kesalahan manusia atau sistem yang terjadi pada pengaturan awal survei dan mempersiapkan data sensor secara komprehensif untuk pemrosesan selanjutnya. Perangkat lunak ini dirancang khusus untuk mengubah data survei yang berasal dari perangkat *multibeam* dan *Singlebeam Echosounder, scanning sonars, pipe/cable trackers*, serta data dari berbagai sensor pendukung seperti GPS, Gyro, Doppler Log, attitude sensor, dan lain sebagainya. NaviEdit dapat menerima dan menginterpretasikan berbagai jenis data survei mentah dan mengubahnya ke dalam format-format tertentu seperti *multibeam files, singlebeam files, ASCII files, tide files, sound velocity files, CTD profiles, run-lines, dan XYZ files*. Selain itu, NaviEdit juga menyediakan kemampuan untuk mengelola database SQL secara simultan. Untuk menggunakan fitur database pada NaviEdit, pengguna perlu menginstal SQL server sebelum menginstal perangkat lunak tersebut. NaviEdit juga menyediakan fasilitas bagi pengguna untuk mengubah data melalui editor *header* dan editor data survei (Herskind, 2008).

NaviModel adalah perangkat lunak yang menyediakan alat untuk membuat *Digital Terrain Model* (DTM) berdasarkan model *Triangular Regular Network* (TRN) atau model *Triangular Irregular Network* (TIN). Pada model TRN, area survei dibagi menjadi sel-sel segitiga seperti pada perangkat lunak EIVA Contour, sedangkan pada model TIN segitiga dibuat berdasarkan *raw data*. Data untuk pemodelan diimpor sebagai data survei yang telah dibersihkan dari NaviEdit (NED), data ASCII XYZ, *run-line* dan profil, atau file survei mentah NaviScan dan NaviPac. NaviModel mendukung integrasi data dari berbagai sumber dan format, sehingga memungkinkan untuk menggabungkan data dari berbagai instrumen survei. Dengan menggunakan NaviModel, kita dapat melakukan pemodelan area survei dengan menentukan parameter pemeruman, ukuran sel, dan tipe data pemeruman yang diinginkan seperti minimum, maksimum, rata-rata, nilai terakhir, standar deviasi, dan densitas. Selain itu, NaviModel juga memiliki kemampuan untuk melakukan kalibrasi dan koreksi data melalui proses *patch test*. *Patch test* merupakan langkah penting dalam proses kalibrasi yang melibatkan perhitungan pergerakan kapal seperti *pitch, roll, yaw*, dan koreksi waktu.

## 2.12 FMGT

*Fledermaus Geocoder Toolbox* (FMGT) adalah perangkat lunak yang dapat digunakan untuk pemrosesan *backscatter* dan mosaik multifrekuensi, normalisasi *intra-vessel*, dan karakterisasi dasar laut. Dikembangkan oleh QPS (*Quality Positioning Services*), *software* ini dirancang untuk membantu dalam geokoding dan *georeferencing* data yang dikumpulkan selama survei hidrografi atau kegiatan lain yang berhubungan dengan laut. Perusahaan perangkat lunak QPS, yang berpusat di Zeist, Belanda, didirikan pada tahun 1986. QPS berfokus pada integrasi sistem sensor survei, pengembangan perangkat lunak yang digunakan untuk survei geomatika maritim, *Portable Pilot Units*, dan produksi Peta Navigasi Elektronik (*Electronic Navigation Charts/ENC*). QPS mempunyai beberapa perangkat lunak yang banyak digunakan dalam bidang hidrografi seperti *QPS navigation and positioning software* (QINSy), Qimera, Fledermaus (Fledermaus Geocoder, Fledermaus, Fledermaus Midwater, Fledermaus GIS and Plugins) dan Qastor (Geo-matching, 2023).

FMGT memberikan keuntungan dalam akurasi tinggi, fleksibilitas penggunaan, pengolahan data cepat, interpolasi data, pemrosesan *real-time*, dan dukungan dan pengembangan lanjutan. *Software Fledermaus Geocoder* dapat membaca beberapa file data *backscatter*, menerapkan koreksi, dan kemudian membuat representasi 2D dari dasar laut yang disebut mosaik *backscatter*. Dalam memproses file sumber menjadi mosaik, perangkat lunak

ini dirancang untuk melakukan sebanyak mungkin koreksi guna memaksimalkan konten informasi dalam sinyal *backscatter*. Setelah mosaik telah dihasilkan, berbagai statistik dapat dihitung dan diekspor dalam berbagai format. Format yang dapat digunakan untuk ekspor hasil pengolahan pada FMGT seperti DXF, *GeoTiff*, *XY-class ascii*, *Fledermaus draped backscatter files* dan *ArcGIS grids*. FMGT juga dapat digunakan untuk mengetahui katakteristik dasar laut dari analisis *backscatter angular response*. Penggunaan *software* ini meliputi berbagai bidang atau disiplin ilmu yang berbeda. Contohnya saja untuk pemetaan habitat laut. Pemetaan ini bertujuan untuk menciptakan gambaran lengkap tentang dasar laut, termasuk morfologi, sedimentologi, dan biologi, serta menginterpretasikan hasilnya untuk membuat peta tematik dari habitat-habitat yang berbeda. Peta ini dapat digunakan sebagai panduan dalam kebijakan kelautan, pengelolaan, dan pemanfaatan sumber daya (Geo-matching, 2023).

### 2.13 SNAP

*Sentinel Application Platform* (SNAP) adalah perangkat lunak yang dirancang khusus untuk memproses data citra dari satelit Sentinel, termasuk *Sentinel-1 Toolbox*, *Sentinel-2 Toolbox*, dan *Sentinel-3 Toolbox* (Bekti, 2017). Meskipun demikian, perangkat lunak ini juga dapat digunakan untuk memproses data citra dari satelit lainnya. SNAP merupakan inovasi platform yang didanai oleh ESA SEOM. SNAP memungkinkan pengguna untuk mengeksplorasi, menganalisis, dan memproses data, memfasilitasi penelitian ilmiah, kegiatan pendidikan dan pelatihan, serta pengembangan berbagai aplikasi operasional. Penggunaan SNAP tidak hanya untuk citra satelit, namun untuk semua citra yang memiliki format sesuai dengan format SNAP. Dalam *software* SNAP terdapat *tools* yang dapat digunakan untuk analisis tekstur yaitu *Gray Level Co-occurrence Matrix* (GLCM). Kelebihan SNAP dalam pengolahan citra antara lain, SNAP tersedia secara gratis; dapat menampilkan citra dengan sangat cepat walaupun citra yang ditampilkan mempunyai ukuran penyimpanan yang besar; dapat menampilkan kerangka proses pengolahan; *layer management* yang handal dengan memungkinkan penambahan dan manipulasi *overlay* berbagai data.

### 2.14 ArcGIS

ArcGIS adalah salah satu *software* yang dikembangkan oleh ESRI (*Environment Science & Research Institute*) yang merupakan kompilasi fungsi-fungsi dari berbagai macam software GIS yang berbeda seperti desktop, server, dan GIS berbasis web. *Software* ini mulai dirilis oleh ESRI pada tahun 2000. Produk utama dari ArcGIS adalah ArcGIS desktop, dimana ArcGIS desktop merupakan software GIS profesional yang komprehensif dan dikelompokkan atas tiga komponen yaitu: ArcView (komponen yang fokus ke penggunaan data yang komprehensif, pemetaan dan analisis), ArcEditor (memiliki fokus ke arah editing data spasial) dan ArcInfo (memiliki fokus yang lebih lengkap dalam menyajikan fungsi-fungsi GIS termasuk untuk keperluan analisis geoprosesing). ArcGIS desktop sendiri terdiri atas 4 aplikasi dasar yakni (GIZ, 2012).

#### 1. ArcMap

ArcMap merupakan aplikasi utama yang digunakan dalam ArcGIS yang digunakan untuk mengolah (membuat, menampilkan, memilih, *editing*, *composing* dan *publishing*) peta. ArcMap menyediakan berbagai fitur dan fungsi yang memungkinkan pengguna untuk membuat, mengedit, dan menganalisis peta serta data geografis. Perangkat lunak ini menyediakan antarmuka pengguna yang intuitif dan beragam alat untuk mengelola data spasial, menampilkan informasi dalam bentuk peta, dan melakukan analisis spasial seperti *overlay*, *buffering*, analisis jarak, dan lainnya. ArcMap telah menjadi perangkat lunak yang populer di berbagai bidang, termasuk ilmu lingkungan, perencanaan kota,

manajemen sumber daya alam, dan banyak lagi, karena kemampuannya dalam pemetaan, analisis, dan visualisasi data geografis.

2. ArcCatalog

ArcCatalog adalah aplikasi yang berfungsi untuk mengatur/mengorganisasi berbagai macam data spasial yang digunakan dalam pekerjaan SIG. Fungsi ini meliputi *tools* untuk menjelajah, mengatur, membagi dan menyimpan data-data SIG. ArcCatalog menyediakan antarmuka pengguna yang intuitif untuk menjelajahi struktur data dalam format yang berbeda, seperti file *shapefile*, *geodatabase*, raster, dan data geografis lainnya. Pengguna dapat melihat metadata, mengelola atribut, membuat dan mengedit skema database, serta mengatur hubungan antara berbagai dataset. Selain itu, ArcCatalog juga memungkinkan pengguna untuk mencari dan menemukan data geografis dengan menggunakan berbagai kriteria pencarian, termasuk nama file, kata kunci, lokasi geografis, dan atribut yang terkait.

3. ArcScene

ArcScene merupakan aplikasi yang digunakan untuk mengolah dan menampilkan peta-peta ke dalam bentuk 3D. Dengan ArcScene, dapat mengimpor data geografis, seperti peta, model digital elevasi (*Digital Elevation Model/DEM*), citra satelit, dan data vektor, untuk membuat pemandangan virtual dalam tiga dimensi. Pengguna dapat menjelajahi pemandangan 3D dengan memutar, memperbesar, dan memiringkan pandangan untuk memperoleh pemahaman yang lebih baik tentang topografi, struktur, dan relasi spasial antara objek. Selain itu, ArcScene juga menyediakan fitur-fitur untuk analisis 3D, seperti analisis visibilitas, analisis relief, analisis cahaya dan bayangan, serta analisis jarak.

4. ArcToolbox

ArcToolbox terdiri dari kumpulan aplikasi yang berfungsi sebagai perangkat/*tools* dalam melakukan berbagai analisis keruangan. Dalam ArcToolbox, dapat mengakses berbagai alat *geoprocessing* yang digunakan untuk melakukan tugas-tugas seperti pengolahan data, transformasi koordinat, analisis spasial, pengabstraksian data, pengelolaan atribut, dan banyak lagi. Alat-alat ini dapat digunakan untuk memanipulasi dan menganalisis data geografis, menciptakan pemodelan analisis, serta menjalankan skrip dan proses otomatis.

## 2.15 Google Colab

Google Colab (Google Collaboratory) adalah platform *online* yang populer untuk menjalankan eksperimen *data science* dan *Machine Learning*. Platform ini menyediakan lingkungan yang kuat dan fleksibel untuk menjalankan kode Python, termasuk dukungan untuk *library* populer seperti TensorFlow, Keras, dan PyTorch. Google Colab merupakan produk berbasis browser yang dibuat oleh Google Research yang memungkinkan pengguna untuk menulis dan menjalankan kode Python tanpa konfigurasi khusus (Chouinard, 2022). Google Colab adalah Jupyter Notebook yang gratis dan mudah digunakan yang menyediakan *Graphics Processing Unit* (GPU) gratis, sumber daya yang berharga untuk tugas-tugas *Machine Learning*.

*Python* adalah bahasa pemrograman yang gratis atau *open source*, yang berarti tidak ada batasan dalam hal menyalin atau mendistribusikannya (Clinton & Sengkey, 2019). *Python* dilengkapi dengan *source code*, *debugger*, dan *profiler*, serta antarmuka yang mencakup antarmuka pengguna, fungsi sistem, GUI (*Graphical User Interface*), dan basis data. *Python* dapat digunakan di berbagai sistem operasi, termasuk sistem UNIX, PCs (DOS, Windows, OS/2), Macintosh, dan lainnya (Rosmala & Dwipa, 2012).

Dalam pengolahan DNN biasanya digunakan *library* TensorFlow dan Keras. TensorFlow, yang dikembangkan oleh tim Google Brain pada tahun 2015, merupakan kerangka

kerja yang digunakan untuk perhitungan numerik. Saat ini, banyak perusahaan besar menggunakan TensorFlow untuk mengembangkan aplikasi kecerdasan buatan, seperti klasifikasi citra, pemrosesan bahasa alami, dan pengembangan chatbot. TensorFlow menyediakan antarmuka untuk mengekspresikan algoritma *Machine Learning* dan aplikasi untuk mengeksekusi algoritma tersebut. Kerangka kerja ini mendukung pemodelan *Neural Network* seperti *Recurrent Neural Network* (RNN), *Restricted Boltzmann Machine* (RBM), *Convolutional Neural Network* (CNN), *Dense Neural Network* (DNN) dan *Dynamic Bayesian Network* (DBN), serta eksekusi paralel. Nama "TensorFlow" merujuk pada penggunaan tensor, yaitu jenis *array* multidimensi yang digunakan di seluruh jaringan (Wiranda dkk., 2020)

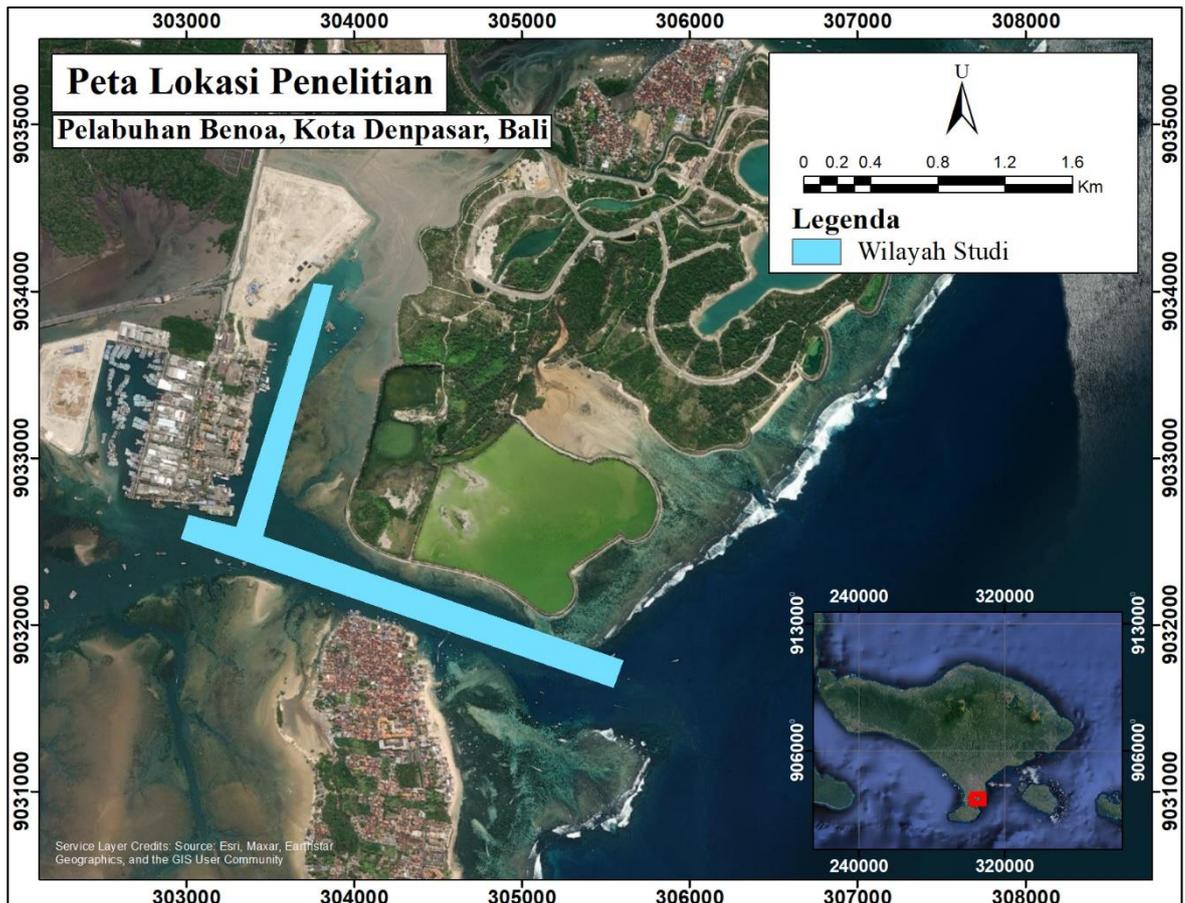
Keras adalah sebuah *library* tingkat tinggi untuk jaringan saraf tiruan (*Deep Learning*) yang ditulis dalam bahasa *Python* dan dapat berjalan di atas TensorFlow, CNTK, dan Theano. *Library* ini menyediakan fitur yang dirancang untuk memudahkan pengembangan dalam bidang *Deep Learning*. Keras dikembangkan untuk memungkinkan pengujian yang cepat baik menggunakan CPU (*Central Processing Unit*) maupun GPU (*Graphic Processing Unit*), dan mendukung algoritma-algoritma seperti *Convolutional Neural Tetwork* (CNN), *Dense Neural Network* (DNN), *Recurrent Neural Network* (RNN), maupun yang lainnya (Ekojono dkk., 2020). Keras diintegrasikan secara erat dengan TensorFlow, sehingga memungkinkan pengguna untuk menggabungkan kelebihan keduanya, seperti kemampuan komputasi GPU dan TPU dari TensorFlow dengan antarmuka yang mudah digunakan dari Keras.

*“Halaman ini sengaja dikosongkan”*

## BAB III METODOLOGI PENELITIAN

### 3.1 Lokasi Penelitian

Penelitian tugas akhir ini menggunakan *Multibeam Echosounder* sebagai instrumen pengambilan data. Pengambilan data dilakukan di wilayah perairan Pelabuhan Bena, Desa Pedungan, Kecamatan Denpasar Selatan, Kota Denpasar, Bali. Berikut merupakan gambar lokasi penelitian yang ditunjukkan oleh Gambar 3.1.



Gambar 3.1 Peta Lokasi Penelitian

### 3.2 Data dan Peralatan

Bahan dan peralatan yang digunakan pada Tugas Akhir ini adalah sebagai berikut:

#### 3.2.1 Data

Adapun bahan yang digunakan dalam penelitian ini adalah sebagai berikut:

1. *Raw data Multibeam Echosounder R2Sonic* sebagai data primer yang diambil pada tanggal 31 Januari 2022 di perairan sekitar Pelabuhan Bena, Bali
2. Data pasang surut sebagai data primer yang diambil tanggal 31 Januari 2022
3. Data *Sound Velocity Profiler* sebagai data primer
4. Data *insitu* berupa data sampel sedimen di perairan sekitar Pelabuhan Bena, Bali sebagai data primer

#### 3.2.2 Peralatan

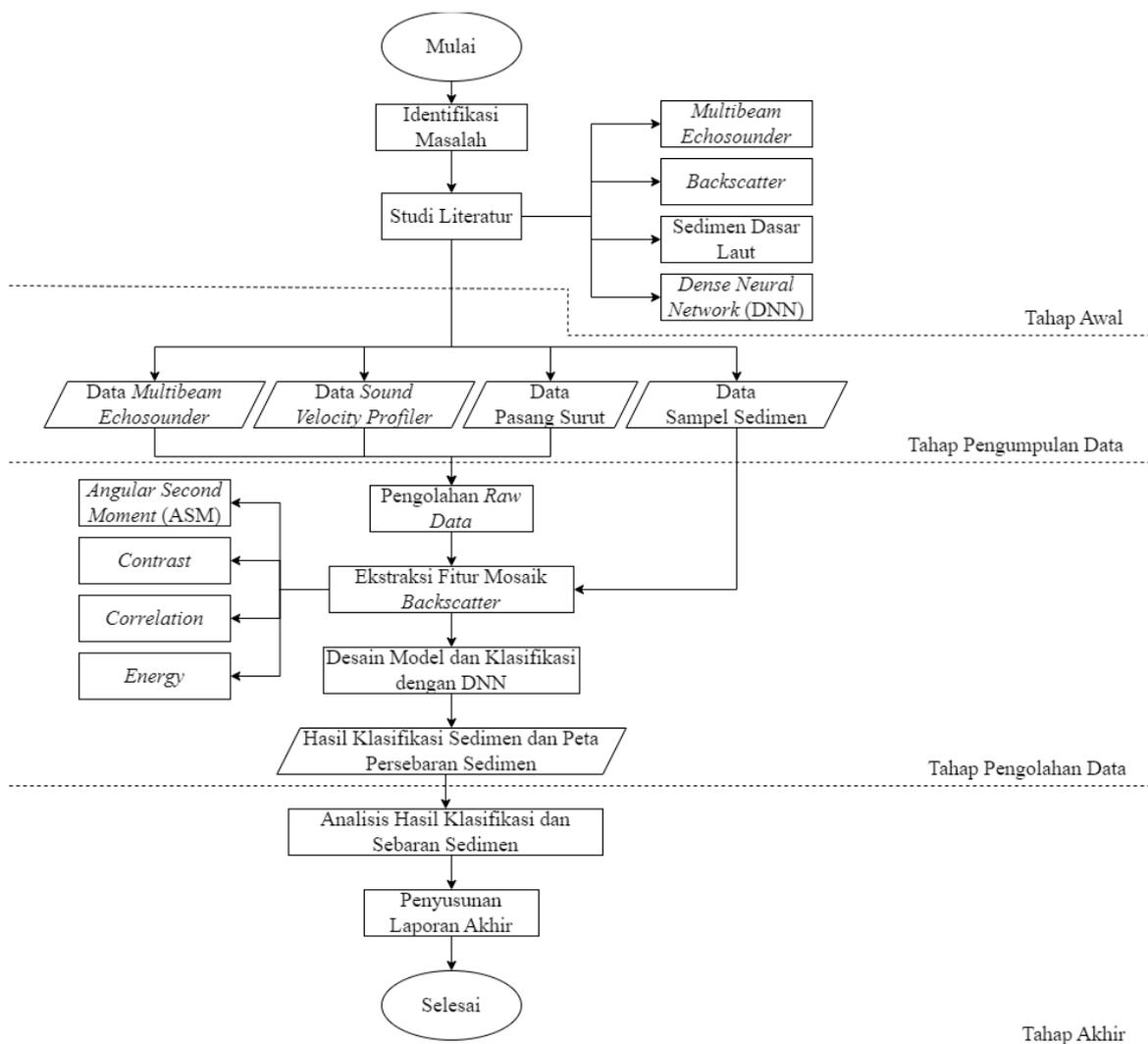
Adapun peralatan yang digunakan dalam penelitian ini untuk pengolahan data adalah sebagai berikut:

- d. Perangkat keras (*hardware*)
  1. Laptop dengan spesifikasi CPU 64-bit, *procesor* Intel Core-i5, RAM 12GB
- e. Perangkat lunak (*software*)
  1. EIVA (NaviEdit dan NaviModel) digunakan untuk mengolah *raw data Multibeam Echosounder*
  2. FMGT digunakan untuk membuat mosaik *backscatter*
  3. SNAP digunakan untuk melakukan ekstraksi fitur dari mosaik *backscatter* dan ekstraksi nilai intensitas tiap titik sedimen
  4. Google Colab digunakan untuk menjalankan pemrograman *Dense Neural Network*.
  5. TensorFlow dan Keras digunakan untuk pembuatan *script* pemrograman
  6. ArcGIS 10.8 digunakan untuk *layoutting* peta dan ekstraksi nilai data raster
  7. Microsoft Excel 365 untuk membuat tabel yang diperlukan dalam penelitian
  8. Microsoft Word 365 untuk membuat laporan akhir dari penelitian

### 3.3 Metodologi Penelitian

#### 3.3.1. Tahapan Pelaksanaan Penelitian

Urutan pelaksanaan penelitian Tugas Akhir ini digambarkan dalam diagram alir sebagai berikut:



Gambar 3.2 Diagram Alir Tahapan Pelaksanaan Penelitian

Penjelasan diagram alir tahap pelaksanaan dalam penelitian yang akan dilakukan adalah sebagai berikut.

## 1. Tahap Awal

### a. Identifikasi Masalah

Identifikasi masalah merupakan tahap awal dalam penelitian tugas akhir ini. Identifikasi masalah melibatkan mengenali, mendefinisikan, dan memahami masalah yang perlu diselesaikan atau ditangani dalam tugas akhir, dengan mengidentifikasi gejala, penyebab, dan konsekuensi masalah untuk memperoleh pemahaman yang lebih mendalam tentang cakupan masalah. Dalam tahap ini, pelaksana melakukan beberapa hal yang meliputi penjabaran latar belakang, merumuskan masalah, merumuskan batasan-batasan masalah, serta mengidentifikasi tujuan dan manfaat penelitian.

### b. Studi Literatur

Studi literatur merupakan tahap awal dimana pelaksana mempelajari lebih lanjut mengenai masalah yang akan diteliti. Dalam tahap ini, pelaksana lebih banyak mencari referensi yang relevan dengan penelitian yang akan dilaksanakan. Referensi yang dapat dijadikan literatur berupa jurnal, buku, maupun penelitian-penelitian yang telah dilakukan sebelumnya. Referensi yang dipelajari oleh penulis antara lain mengenai *Multibeam Echosounder* (termasuk survei hidrografi dan prinsip kerja), *backscatter* (termasuk mosaik *backscatter* dan fitur mosaik *backscatter*), sedimen dasar laut, dan *Dense Neural Network*.

## 2. Tahap Pengumpulan Data

### a. Pengumpulan Data

Pengumpulan data merupakan tahap dimana pelaksana mengumpulkan data-data yang nantinya akan dipakai untuk penelitian. Pada tahap ini pelaksana mendapatkan *raw data Multibeam Echosounder* dan data pasang surut serta data SVP untuk wilayah perairan sekitar perairan Pelabuhan Benoa pada tanggal 31 Januari 2022. Pelaksana juga mendapat data *insitu* berupa data sampel sedimen di wilayah penelitian.

## 3. Tahap Pengolahan Data

### a. Pengolahan *Raw Data*

Pada tahap ini dilakukan pengolahan *raw data* MBES menggunakan software NaviEdit dan NaviModel. Pada pengolahan *raw data* digunakan data *Multibeam Echosounder*, data *Sound Velocity Profiler*, dan Data Pasang Surut. Setelah itu dilakukan pengolahan *backscatter* dengan menggunakan *software* FMGT. Pengolahan lebih jelas akan dijelaskan pada Gambar 3.3.

### b. Ekstraksi Fitur Mosaik *Backscatter*

Pada tahap ini dilakukan pengolahan untuk mendapatkan fitur mosaik *backscatter*. Pengolahan ini dilakukan menggunakan *software* SNAP. Pada pengolahan ini dibutuhkan data sampel sedimen. Pengolahan lebih jelas akan dijelaskan pada Gambar 3.5.

### c. Desain Model dan Klasifikasi dengan DNN

Pada tahap ini dilakukan pembuatan desain model dan klasifikasi dengan DNN. Hasil dari tahap ini berupa klasifikasi jenis sedimen dan peta sebaran sedimen. Pengolahan lebih jelas akan dijelaskan pada Gambar 3.11.

#### 4. Tahap Akhir

##### a. Analisis

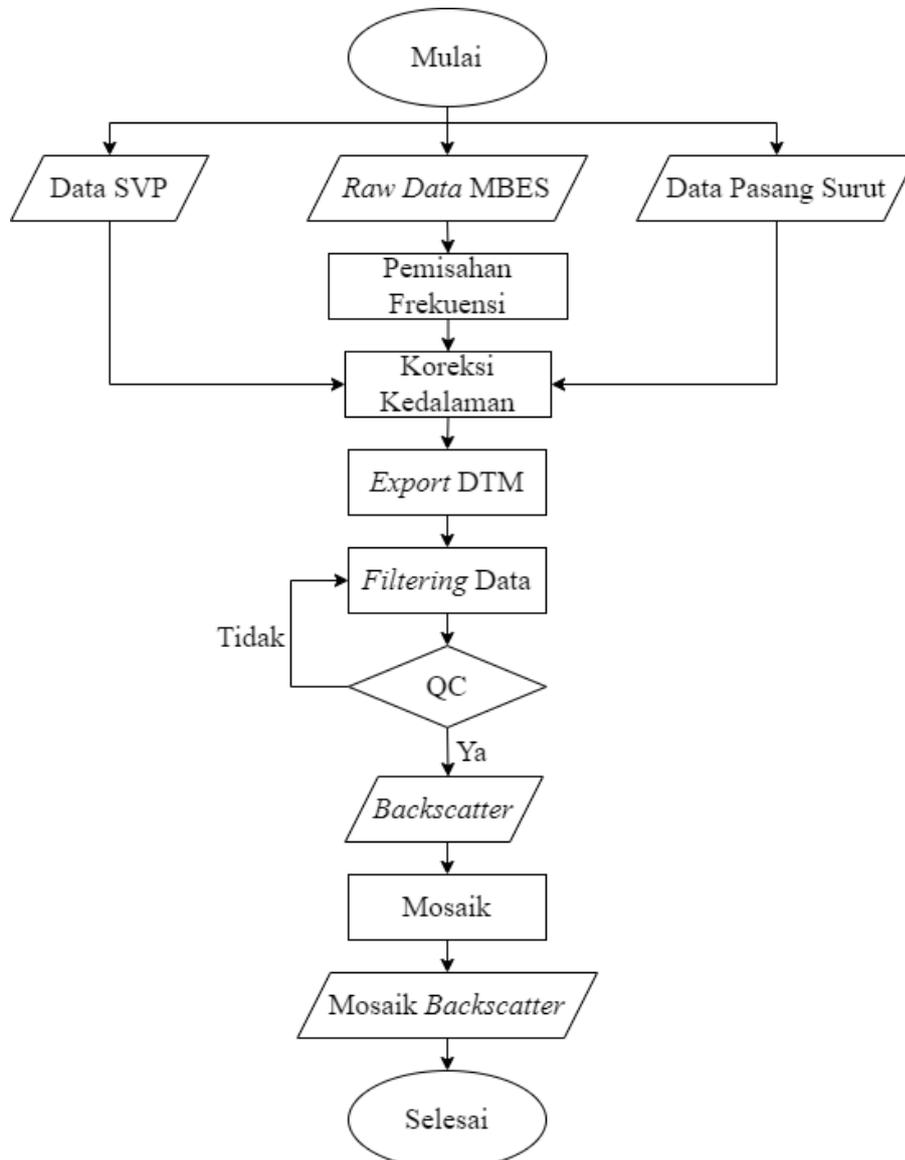
Kegiatan analisis dilakukan dengan menganalisis hasil pengolahan data. Kegiatan ini bertujuan untuk mengetahui model yang tepat yang mempunyai akurasi paling akurat, sehingga akan didapatkan klasifikasi jenis sedimen dasar laut menggunakan data mosaik dan pola fitur *backscatter* dengan metode *Dense Neural Network* di perairan sekitar Pelabuhan Benoa.

##### b. Penyusunan Laporan Akhir

Kegiatan penyusunan laporan akhir dilakukan untuk menyusun laporan dari semua kegiatan yang telah dilakukan pelaksana selama proses penelitian. Penyusunan laporan ini merupakan kegiatan pada tahap akhir dari penelitian yang dilakukan.

### 3.3.2. Tahapan Pengolahan *Raw Data*

Urutan tahapan pengolahan *raw data* pada Tugas Akhir ini digambarkan dalam diagram alir sebagai berikut:



Gambar 3.3 Diagram Alir Tahapan Pengolahan *Raw Data*

Penjelasan diagram alir tahap pelaksanaan dalam penelitian yang akan dilakukan adalah sebagai berikut.

a. Pemisahan Frekuensi

Data *raw Multibeam Echosounder* R2Sonic merupakan data multifrekuensi sehingga diperlukan pemisahan frekuensi. Frekuensi yang ada di dalam data ini yaitu 170 kHz, 230 kHz, 270 kHz, 350 kHz, dan 400 kHz. Pemisahan frekuensi dilakukan untuk menganalisis data MBES pada setiap frekuensi dan mendapatkan informasi yang lebih rinci untuk setiap frekuensinya. Pemisahan frekuensi ini bertujuan agar nilai pada setiap frekuensi dapat membantu mengidentifikasi jenis seimen yang berbeda berdasarkan karakteristik akustik pada berbagai frekuensi. Pemisahan frekuensi ini menggunakan *software* NaviEdit, dimana satu file *raw* data MBES digandakan menjadi 5 dan disesuaikan namanya sesuai dengan frekuensi. Setelah itu dilakukan pemisahan dengan bantuan SQL *Query*. Pemisahan frekuensi ini harus dilakukan dengan tepat, karena jika salah dalam memisahkan maka akan berdampak pada pengolahan selanjutnya. Untuk itu, pemisahan dapat dilakukan secara urut per lajur.

b. Koreksi Kedalaman

Data *raw Multibeam Echosounder* perlu dilakukan koreksi dengan data pasang surut dan data SVP. Hal ini bertujuan agar hasil pengukuran dapat terintegrasi dengan *chart datum* yang digunakan. Data MBES adalah hasil pantulan gelombang suara dari dasar laut, dan koreksi diperlukan untuk mengatasi berbagai faktor yang dapat mempengaruhi akurasi data. Tujuan lain yaitu agar dapat menyesuaikan dengan cepat rambat gelombang yang terukur sesuai dengan kondisi perairan. Koreksi ini akan mengoreksi dan memperbaiki estimasi kedalaman yang dihasilkan dari pengukuran. Data pasang surut digunakan untuk menyesuaikan pengaruh variasi tinggi air yang disebabkan oleh pasang surut. Karena pada saat pengukuran menggunakan MBES, tinggi permukaan laut berfluktuasi akibat pasang surut dan dapat mempengaruhi ketepatan pengukuran kedalaman. Untuk itu koreksi kedalaman dengan pasang surut penting untuk dilakukan. Sedangkan data SVP digunakan untuk memperbaiki pengaruh kecepatan suara di dalam air yang dapat mempengaruhi perhitungan kedalaman yang akurat. Karena kecepatan suara dalam air bervariasi tergantung suhu, salinitas dan tekanan air, diperlukan koreksi kedalaman dengan data SVP untuk membantu menghasilkan data yang lebih akurat. Dengan menggabungkan informasi kecepatan suara dalam air dari data SVP, kualitas data kedalaman dari MBES dapat ditingkatkan, dan hasil survei dapat lebih baik dalam analisis. Koreksi kedalaman ini menggunakan *software* NaviEdit dengan cara memasukkan data pasang surut dan data SVP ke dalam *software*.

c. *Export DTM*

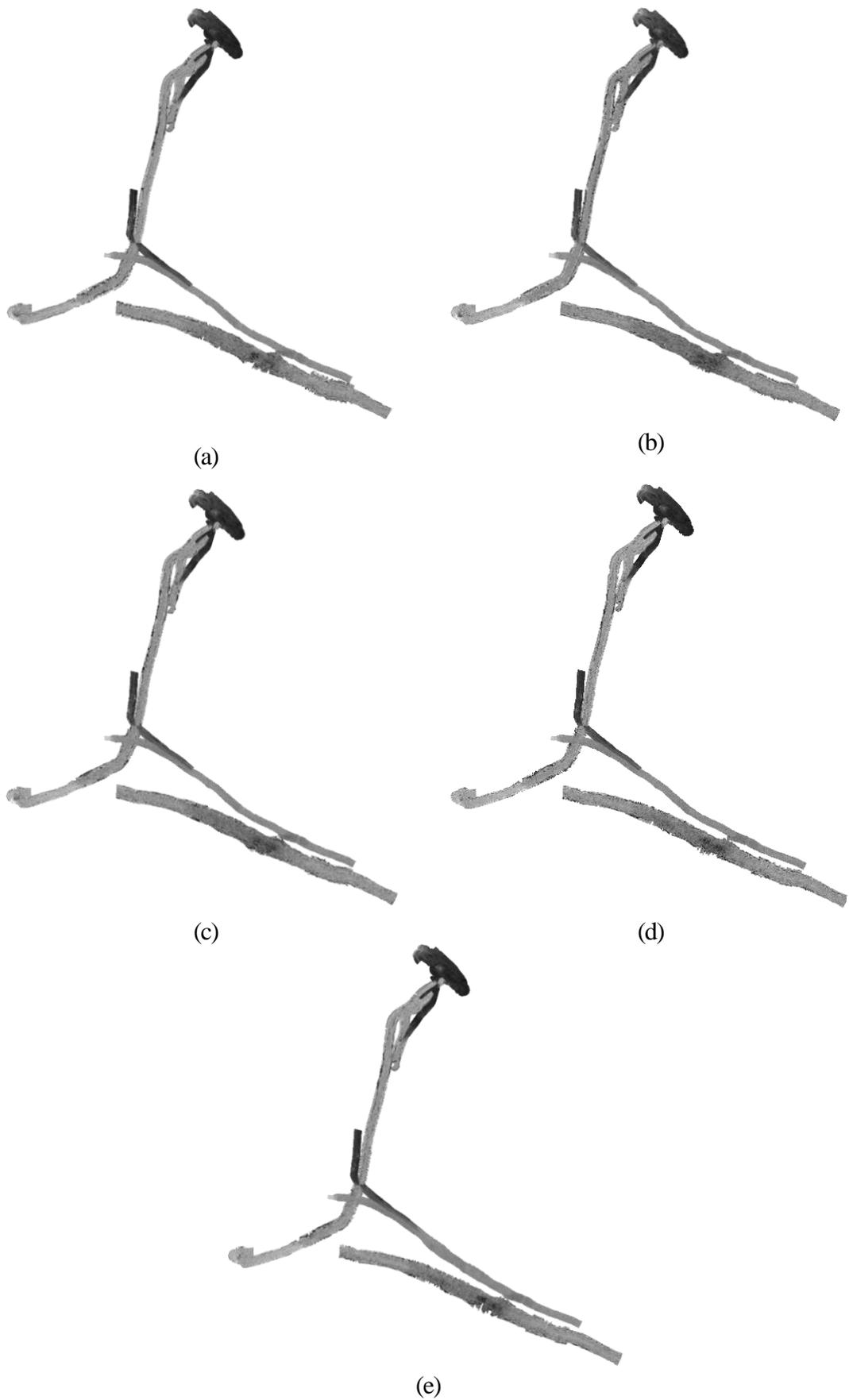
Data yang sudah terkoreksi selanjutnya dimodelkan dengan *Digital Terrain Model* (DTM). DTM merupakan model permukaan yang terbentuk dari titik-titik yang memiliki koordinat 3 dimensi yaitu X, Y, Z dan berupa titik-titik sampel. Dalam data MBES, mengubah ke bentuk DTM dapat merepresentasikan dasar laut atau permukaan bawah laut. DTM menggambarkan bentuk dan topografi dasar laut berdasarkan data kedalaman yang dihasilkan. *Export DTM* bertujuan untuk memudahkan pemrosesan selanjutnya yaitu *filtering* data. Proses *export DTM* pada penelitian ini dilakukan secara bergantian untuk setiap frekuensi. *Export DTM* dilakukan menggunakan *software* NaviModel. Karena *export DTM* dilakukan untuk setiap lajur dan setiap frekuensi, jadi terdapat 60 DTM yang akan terbentuk. Hal ini karena terdapat 12 lajur dan dalam satu lajur terdapat 5 frekuensi.

#### d. *Filtering Data*

Pada tahap ini dilakukan pembersihan terhadap data *noise* atau nilai kedalaman yang tidak sesuai. *Noise* dapat berasal dari sumber eksternal seperti pantulan dari permukaan air, vegetasi laut, atau sinyal elektromagnetik lainnya. Tujuan dari *filtering* data adalah untuk meningkatkan kualitas data dan mendapatkan representasi yang lebih akurat tentang kondisi dasar laut. Metode *line filtering* berdasarkan lajur perum atau metode *area filtering* berdasarkan luasan sapuan dapat digunakan untuk proses *data filtering*. *Filtering* data dilakukan menggunakan *software* NaviModel. Dalam melakukan *filtering* data, penting untuk mengetahui mana *noise* dan mana yang merupakan detail penting dalam data MBES. *Filtering* data juga dapat dilakukan secara manual maupun otomatis menggunakan *tools* yang telah tersedia di NaviModel. Jika membersihkan *noise* dengan menggunakan metode manual, dilakukan identifikasi secara manual dan menghilangkan *noise* secara visual dari data MBES. Sedangkan jika menggunakan metode otomatis, maka sistem pada NaviModel akan mendeteksi dan menghilangkan *noise* secara otomatis. Metode otomatis lebih cepat dan lebih efisien dalam proses membersihkan *noise*, namun metode ini seringkali mendeteksi hal yang bukan *noise*. Untuk itu, jika *noise* yang akan dihilangkan sangat sedikit, lebih baik menggunakan metode manual, tetapi jika *noise* yang akan dihilangkan banyak, akan lebih mudah jika menggunakan metode otomatis. Kombinasi kedua metode juga dapat digunakan sehingga akan memberikan hasil yang lebih baik dan efisien dalam membersihkan *noise* pada data MBES. Setelah *filtering* selesai, perlu dilakukan pemeriksaan data kembali setelah dilakukan *data filtering* sehingga akan dihasilkan data yang benar-benar bersih dari *noise*. Hal tersebut dapat dilakukan dengan memeriksa data per *scene*.

#### e. Mosaik

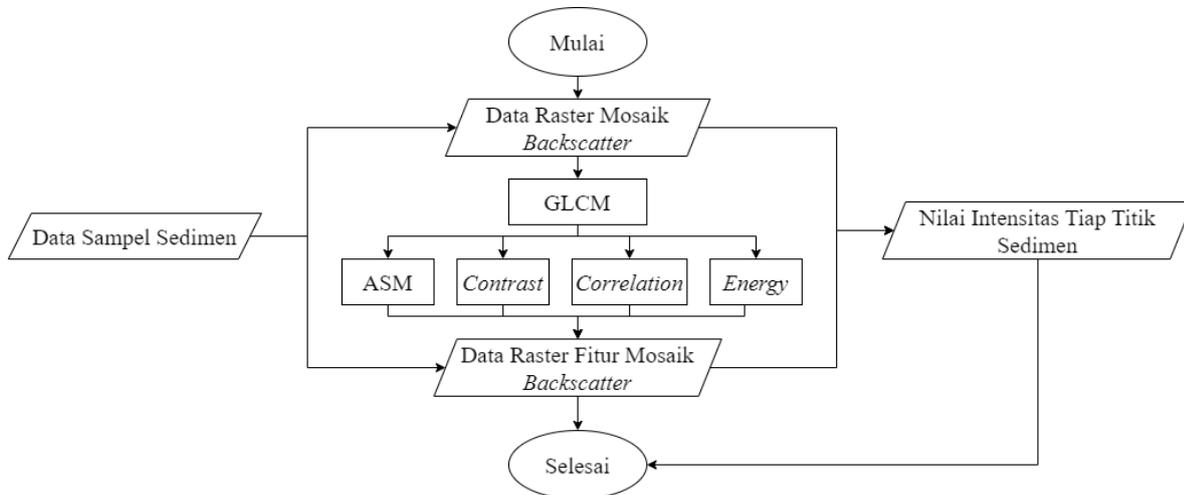
Pada tahap ini nilai intensitas yang dihasilkan dari pemrosesan *backscatter* pada setiap titik di dasar laut kemudian akan digabungkan hingga menjadi data citra (mosaik) *backscatter*. Proses mosaik dilakukan menggunakan *software* FMGT. Data berformat \*.gsf yang telah di-*filtering*, kemudian dimasukkan ke *software* FMGT yang diproses menggunakan *tools* *Mosaic*. Pada proses mosaik ini, ada beberapa hal yang perlu diatur dalam *processing parameter* sebelum melakukan *input file*. Hal-hal yang harus diatur seperti frekuensi, nilai *absorption*, dan *sonar type*. Karena pada tahap sebelumnya telah dilakukan pemisahan frekuensi, jadi untuk pengolahan mosaik juga dilakukan berdasarkan frekuensi. Sehingga dihasilkan lima mosaik *backscatter* sesuai dengan frekuensi yaitu frekuensi 170 kHz, 230 kHz, 270 kHz, 350 kHz, dan 400 kHz. Dalam pemrosesan ini juga dilakukan *re-compute size*. Proses *re-compute size* dilakukan untuk *pixel size* dimana hal ini bertujuan untuk memperbarui ukural piksel dalam citra mosaik *backscatter*. Pembaharuan ukuran piksel ini agar memori yang digunakan tidak terlalu besar. Jika ukuran piksel menggunakan ukuran asli, untuk proses selanjutnya akan menghabiskan waktu yang lama dikarenakan ukuran memori yang terlalu besar. Proses *re-compute size* dilakukan setelah proses mosaik *backscatter*. Berikut ini merupakan hasil dari proses mosaik dari *software* FMGT.



Gambar 3.4 Mosaik *Backscatter* a) 170kHz b) 220kHz c) 270kHz d) 350kHz e) 400kHz

### 3.3.3. Tahapan Ekstraksi Fitur Mosaik *Backscatter*

Urutan tahapan ekstraksi fitur mosaik *backscatter* pada tugas akhir ini digambarkan dalam diagram alir sebagai berikut:



Gambar 3.5 Diagram Alir Tahapan Ekstraksi Fitur Mosaik *Backscatter*

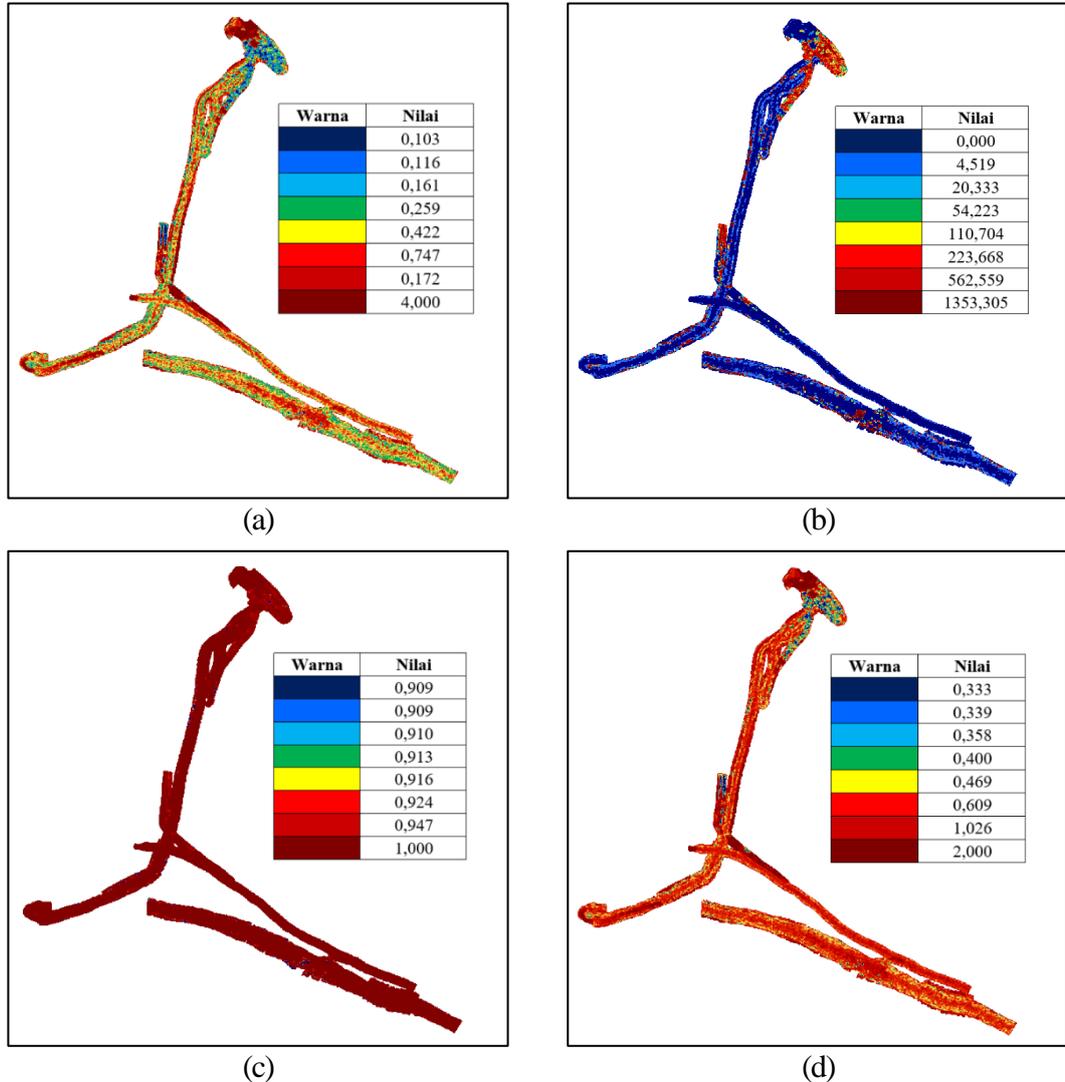
Penjelasan diagram alir tahap pelaksanaan dalam penelitian yang akan dilakukan adalah sebagai berikut.

#### a. Ekstraksi Fitur Menggunakan GLCM

Ekstraksi fitur menggunakan GLCM diawali dengan data raster mosaik backscatter akan di ekstrak menggunakan 4 fitur yaitu ASM, kontras, korelasi dan energi. Hasil dari ekstrak ini menghasilkan data raster fitur mosaik *backscatter*. Ekstraksi fitur dilakukan dengan *software* SNAP menggunakan *tools* tekstur analisis. Ekstraksi fitur dilakukan satu per satu untuk per frekuensi. Hal ini dilakukan agar tidak perlu melakukan pemisahan hasil, karena jika ekstraksi dilakukan menggunakan 4 fitur sekaligus, maka hasilnya akan menjadi satu dan akan diperlukan pemisahan untuk hasilnya. Terdapat 5 frekuensi mosaik *backscatter* yang diekstrak dengan 4 fitur sehingga dihasilkan 20 data raster hasil ekstraksi fitur mosaik *backscatter*.

Ekstraksi fitur menggunakan GLCM memiliki hasil yang berbeda untuk keempat fitur. Pada proses ekstraksi fitur terdapat *processing parameters* yang dapat diubah sesuai kebutuhan. Pada penelitian ini menggunakan *window size* 7×7 dan *quantitation levels* 32. *Window size* 7×7 berarti ukuran area yang digunakan dalam analisis data mosaik *backscatter* adalah 7 piksel lebar dan 7 piksel tinggi. Dengan memindai seluruh data menggunakan *window size* 7×7, informasi tekstur akan terkumpul dan akan membantu dalam mendapatkan informasi mengenai tekstur keseluruhan dari data mosaik *backscatter*. Penggunaan *window size* 7×7 mengacu pada ukuran *window* yang digunakan untuk menghitung *co-occurrence matrix* dari gambar. Jadi, setiap piksel dalam gambar, *software* SNAP akan memebentuk sebuah *window* berukuran 7×7 piksel di sekitar piksel tersebut. Kemudian *co-occurrence matrix* dihitung dari intensitas piksel dalam *window* tersebut. Sedangkan *quantitation levels* adalah tingkat yang berbeda yang dapat digunakan untuk mewakili data dalam suatu system. *Quantitation level* 32 berarti terdapat 32 tingkat atau *level* yang berbeda. Dengan menggunakan *quantitation level*, data dapat disederhanakan, sehingga mengurangi kompleksitas dan ukuran data. Semakin tinggi *quantitation level* maka semakin banyak tingkat yang digunakan untuk merepresentasikan data,

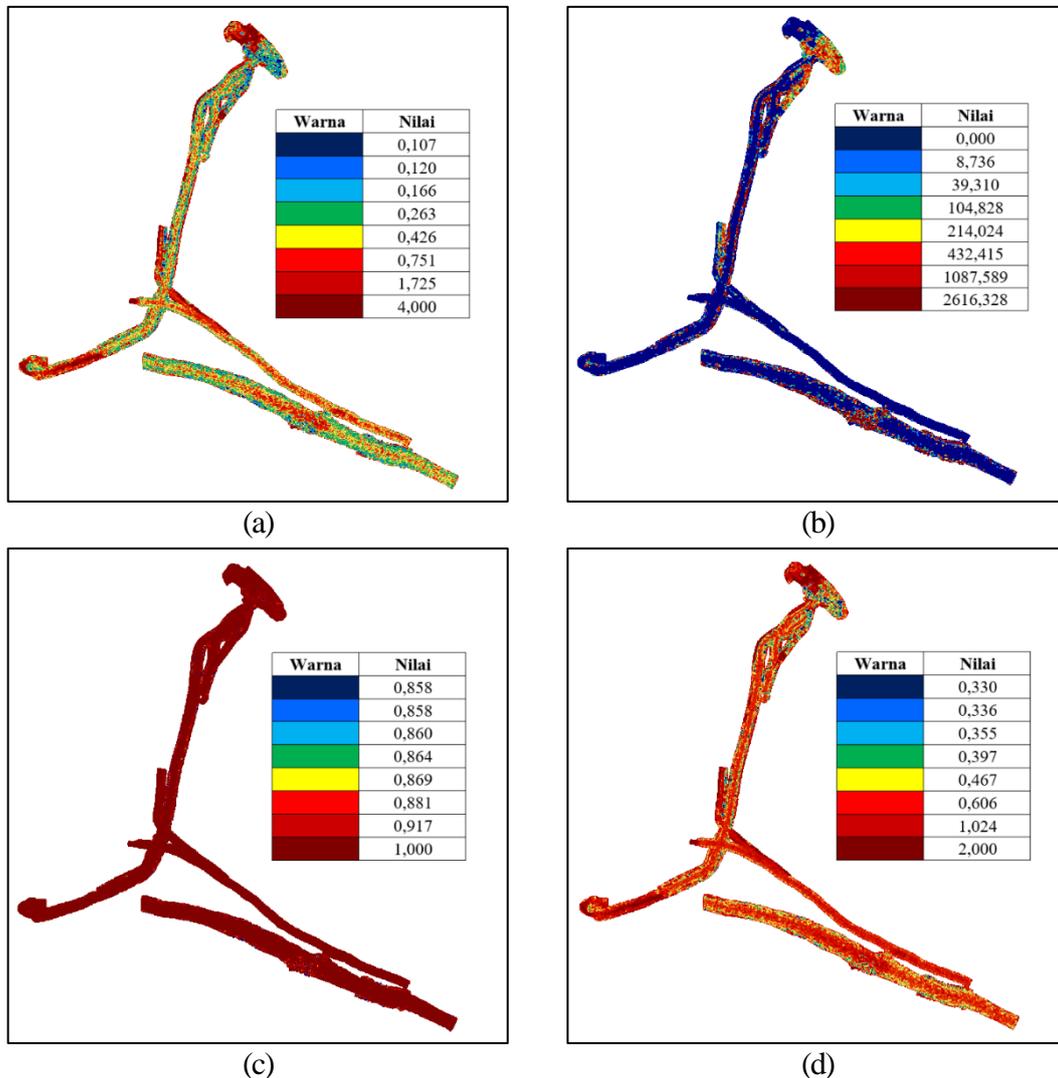
sehingga akan meningkatkan akurasi dan juga meningkatkan ukuran data. *Window size* dan *quantitation level* diatur sama untuk semua data mosaik *backscatter* yang akan di ekstrak fiturnya. Berikut merupakan hasil ekstraksi fitur mosaik *backscatter* per frekuensi.



Gambar 3.6 Hasil Ekstraksi Fitur Frekuensi 170 kHz a) ASM b) *Contrast* c) *Correlation* d) *Energy*

Pada Gambar 3.6a menunjukkan gambar ekstraksi fitur ASM yang didominasi oleh warna hijau dan merah. Rentang nilai untuk fitur ASM ini antara 0,103 sampai 4,000 dimana warna biru tua menandakan nilai yang paling kecil sedangkan warna merah tua menandakan nilai yang paling besar. Lalu Gambar 3.6b menunjukkan gambar ekstraksi fitur *contrast* yang didominasi oleh warna biru tua. Rentang nilai fitur *contrast* antara 0,000 sampai 1353,305 dimana warna biru tua menandakan nilai yang paling kecil dan warna merah tua menandakan nilai yang paling besar. Lalu Gambar 3.6c menunjukkan gambar ekstraksi fitur *correlation* yang didominasi oleh warna merah tua. Rentang nilai untuk fitur *correlation* antara 0,909 sampai 1,000 dimana warna biru tua menandakan nilai yang paling kecil sedangkan warna merah tua menandakan nilai yang paling besar. Untuk Gambar 3.6d menunjukkan ekstraksi

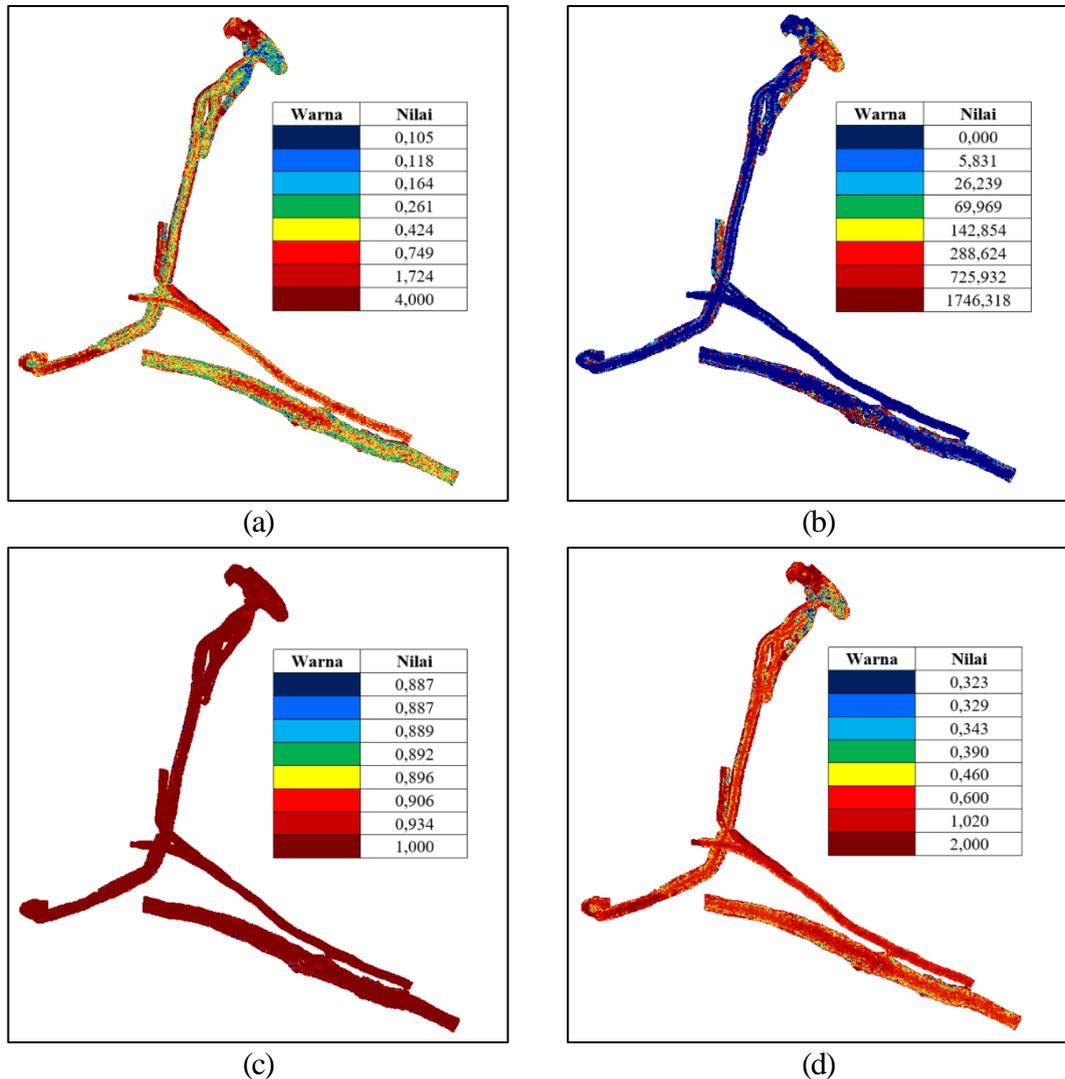
fitur *energy* yang didominasi oleh warna merah cerah. Rentang nilai untuk fitur *energy* antara 0,333 sampai 2,000 dimana warna biru tua menandakan nilai yang paling kecil sedangkan warna merah tua menandakan nilai yang paling besar.



Gambar 3.7 Hasil Ekstraksi Fitur Frekuensi 220 kHz a) ASM b) *Contrast* c) *Correlation* d) *Energy*

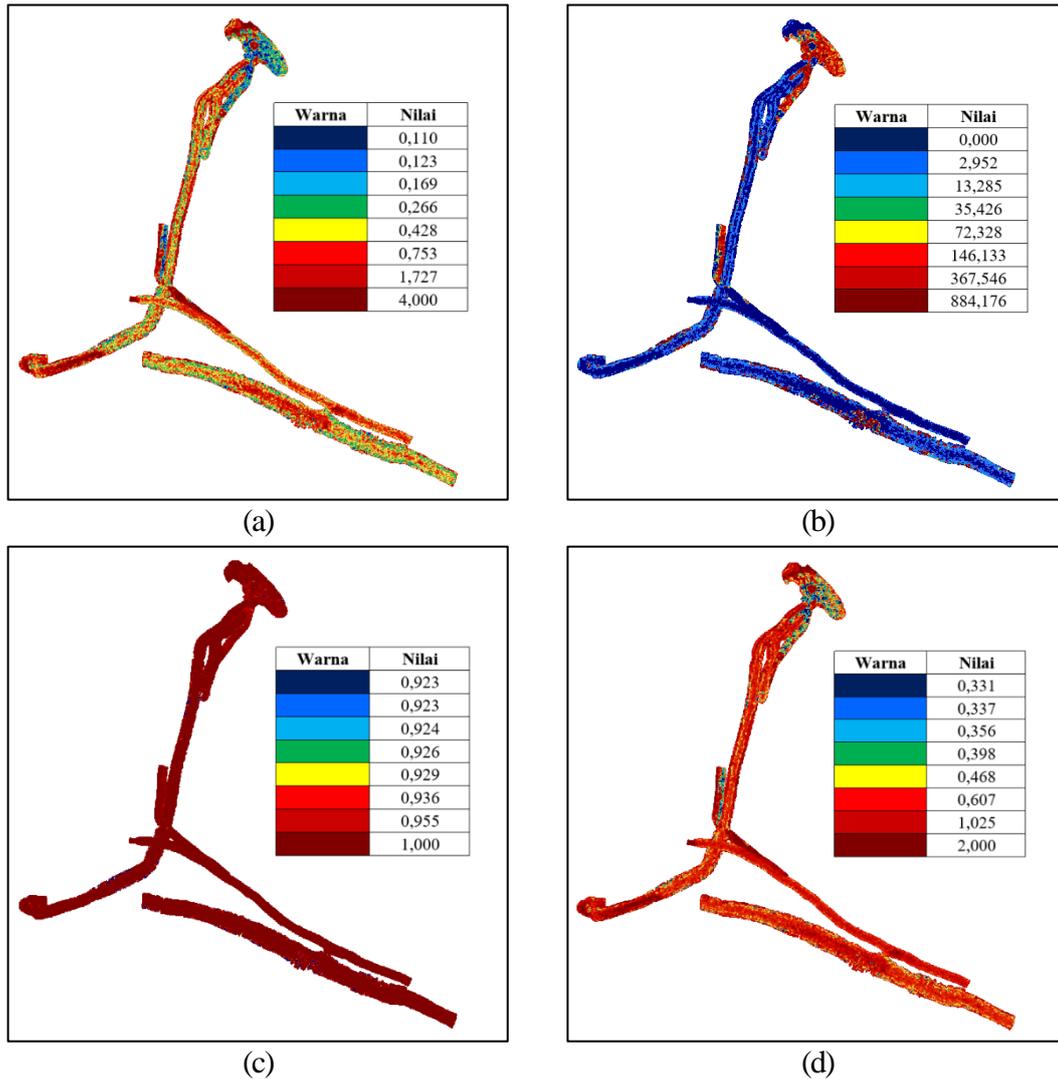
Pada Gambar 3.7a menunjukkan gambar ekstraksi fitur ASM yang didominasi oleh warna hijau dan merah. Rentang nilai untuk fitur ASM ini antara 0,107 sampai 4,000 dimana warna biru tua menandakan nilai yang paling kecil sedangkan warna merah tua menandakan nilai yang paling besar. Lalu Gambar 3.7b menunjukkan gambar ekstraksi fitur *contrast* yang didominasi oleh warna biru tua. Rentang nilai fitur *contrast* antara 0,000 sampai 2616,328 dimana warna biru tua menandakan nilai yang paling kecil dan warna merah tua menandakan nilai yang paling besar. Lalu Gambar 3.7c menunjukkan gambar ekstraksi fitur *correlation* yang didominasi oleh warna merah tua. Rentang nilai untuk fitur *correlation* antara 0,858 sampai 1,000 dimana warna biru tua menandakan nilai yang paling kecil sedangkan warna merah tua menandakan nilai yang paling besar. Untuk Gambar 3.7d menunjukkan ekstraksi fitur *energy* yang didominasi oleh warna merah cerah. Rentang nilai untuk fitur

*energy* antara 0,330 sampai 2,000 dimana warna biru tua menandakan nilai yang paling kecil sedangkan warna merah tua menandakan nilai yang paling besar.



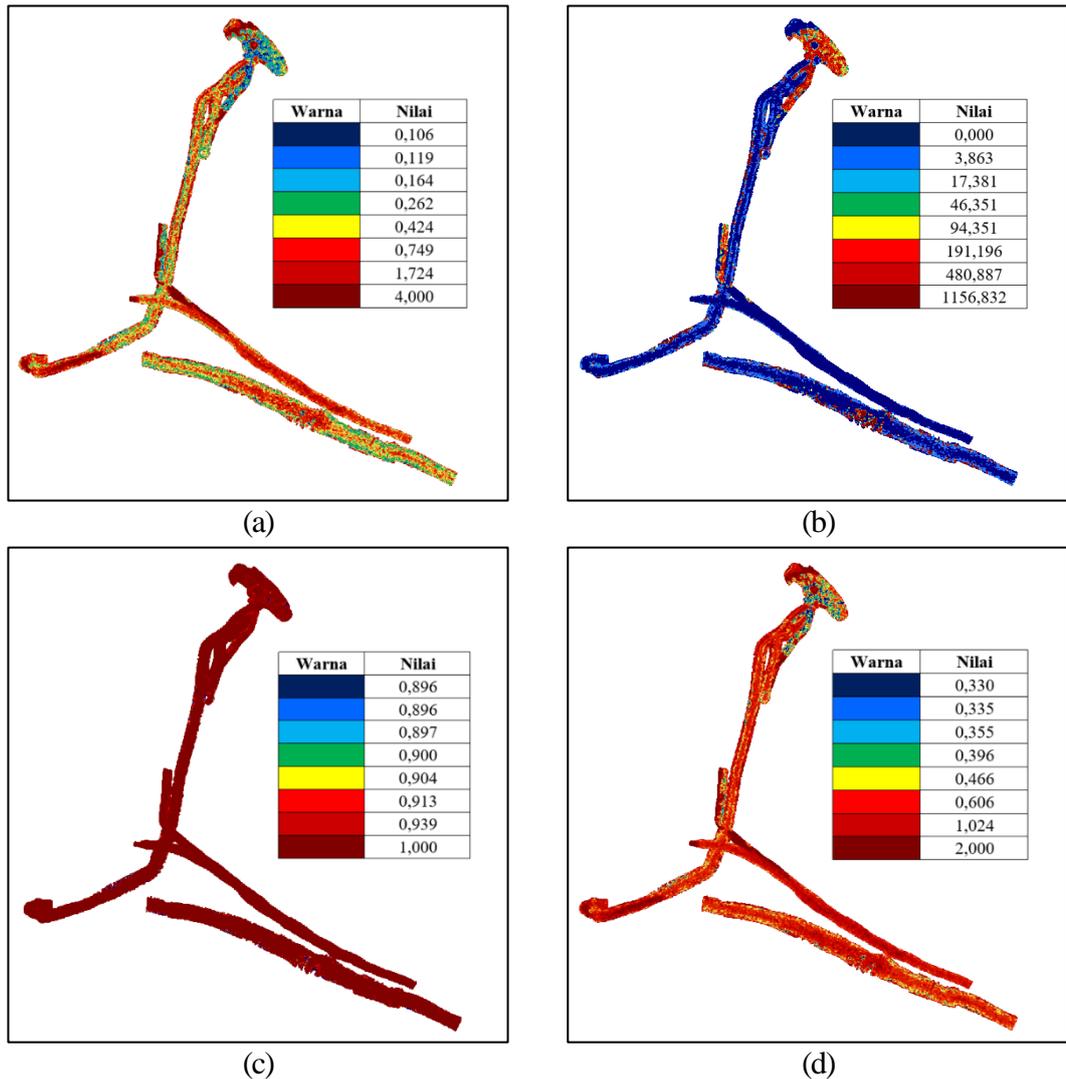
Gambar 3.8 Hasil Ekstraksi Fitur Frekuensi 270 kHz a) ASM b) *Contrast* c) *Correlation* d) *Energy*

Pada Gambar 3.8a menunjukkan gambar ekstraksi fitur ASM yang didominasi oleh warna hijau dan merah. Rentang nilai untuk fitur ASM ini antara 0,105 sampai 4,000 dimana warna biru tua menandakan nilai yang paling kecil sedangkan warna merah tua menandakan nilai yang paling besar. Lalu Gambar 3.8b menunjukkan gambar ekstraksi fitur *contrast* yang didominasi oleh warna biru tua. Rentang nilai fitur *contrast* antara 0,000 sampai 1746,318 dimana warna biru tua menandakan nilai yang paling kecil dan warna merah tua menandakan nilai yang paling besar. Lalu Gambar 3.8c menunjukkan gambar ekstraksi fitur *correlation* yang didominasi oleh warna merah tua. Rentang nilai untuk fitur *correlation* antara 0,887 sampai 1,000 dimana warna biru tua menandakan nilai yang paling kecil sedangkan warna merah tua menandakan nilai yang paling besar. Untuk Gambar 3.8d menunjukkan ekstraksi fitur *energy* yang didominasi oleh warna merah tua. Rentang nilai untuk fitur *energy* antara 0,323 sampai 2,000 dimana warna biru tua menandakan nilai yang paling kecil sedangkan warna merah tua menandakan nilai yang paling besar.



Gambar 3.9 Hasil Ekstraksi Fitur Frekuensi 350 kHz a) ASM b) *Contrast* c) *Correlation* d) *Energy*

Pada Gambar 3.9a menunjukkan gambar ekstraksi fitur ASM yang didominasi oleh warna hijau dan merah. Rentang nilai untuk fitur ASM ini antara 0,110 sampai 4,000 dimana warna biru tua menandakan nilai yang paling kecil sedangkan warna merah tua menandakan nilai yang paling besar. Lalu Gambar 3.9b menunjukkan gambar ekstraksi fitur *contrast* yang didominasi oleh warna biru tua. Rentang nilai fitur *contrast* antara 0,000 sampai 884,176 dimana warna biru tua menandakan nilai yang paling kecil dan warna merah tua menandakan nilai yang paling besar. Lalu Gambar 3.9c menunjukkan gambar ekstraksi fitur *correlation* yang didominasi oleh warna merah tua. Rentang nilai untuk fitur *correlation* antara 0,923 sampai 1,000 dimana warna biru tua menandakan nilai yang paling kecil sedangkan warna merah tua menandakan nilai yang paling besar. Untuk Gambar 3.9d menunjukkan ekstraksi fitur *energy* yang didominasi oleh warna merah tua. Rentang nilai untuk fitur *energy* antara 0,331 sampai 2,000 dimana warna biru tua menandakan nilai yang paling kecil sedangkan warna merah tua menandakan nilai yang paling besar.



Gambar 3.10 Hasil Ekstraksi Fitur Frekuensi 400 kHz a) ASM b) *Contrast* c) *Correlation* d) *Energy*

Pada Gambar 3.10a menunjukkan gambar ekstraksi fitur ASM yang didominasi oleh warna hijau dan merah. Rentang nilai untuk fitur ASM ini antara 0,106 sampai 4,000 dimana warna biru tua menandakan nilai yang paling kecil sedangkan warna merah tua menandakan nilai yang paling besar. Lalu Gambar 3.10b menunjukkan gambar ekstraksi fitur *contrast* yang didominasi oleh warna biru tua. Rentang nilai fitur *contrast* antara 0,000 sampai 1156,832 dimana warna biru tua menandakan nilai yang paling kecil dan warna merah tua menandakan nilai yang paling besar. Lalu Gambar 3.10c menunjukkan gambar ekstraksi fitur *correlation* yang didominasi oleh warna merah tua. Rentang nilai untuk fitur *correlation* antara 0,896 sampai 1,000 dimana warna biru tua menandakan nilai yang paling kecil sedangkan warna merah tua menandakan nilai yang paling besar. Untuk Gambar 3.10d menunjukkan ekstraksi fitur *energy* yang didominasi oleh warna merah tua. Rentang nilai untuk fitur *energy* antara 0,330 sampai 2,000 dimana warna biru tua menandakan nilai yang paling kecil sedangkan warna merah tua menandakan nilai yang paling besar.

b. Mengolah Nilai Intensitas Tiap Titik Sedimen

Nilai intensitas tiap titik sedimen diperlukan sebagai input untuk DNN. Nilai intensitas diambil dari data raster mosaik *backscatter* dan data raster fitur mosaik *backscatter* menggunakan data sampel sedimen sehingga akan dihasilkan nilai intensitas untuk tiap titik sedimen. Nilai ini nantinya akan dimasukkan ke dalam satu file excel yang akan digunakan untuk input DNN. Pada penelitian ini menggunakan lima frekuensi sehingga terdapat 5 data raster mosaik *backscatter* dan 20 data raster fitur mosaik *backscatter* yang setiap rasternya dilakukan ekstraksi nilai intensitas. Berikut merupakan data nilai intensitas pada tiap titik sedimen.

Tabel 3.1 Nilai Intensitas Mosaik *Backscatter* (dB)

<b>Nama Titik</b>	<b>170kHz</b>	<b>220kHz</b>	<b>270kHz</b>	<b>350kHz</b>	<b>400kHz</b>
PT1	-47,500	-34,707	-41,099	-41,675	-39,093
PT2	-31,214	-39,409	-34,342	-38,853	-26,471
PT3	-36,277	-27,440	-35,041	-38,853	-40,134
PT4	-34,088	-33,104	-36,905	-36,031	-36,230
PT5	-34,088	-38,554	-37,604	-38,584	-39,093
PT6	-48,732	-47,423	-47,391	-49,605	-49,632
PT7	-32,445	-37,913	-35,274	-35,359	-35,970
PT8	-37,920	-42,294	-44,828	-43,960	-41,695
PT9	-34,361	-47,744	-35,973	-45,573	-50,673
PT10	-37,235	-35,348	-36,322	-39,525	-41,044
PT11	-45,858	-42,615	-48,090	-49,605	-50,673
PT12	-33,403	-32,142	-33,759	-36,300	-36,620

Tabel 3.2 Nilai Intensitas ASM

<b>Nama Titik</b>	<b>170kHz</b>	<b>220kHz</b>	<b>270kHz</b>	<b>350kHz</b>	<b>400kHz</b>
PT1	0,255	0,488	0,123	0,257	0,287
PT2	1,270	0,345	0,484	0,645	1,148
PT3	0,682	0,395	0,303	0,696	0,583
PT4	1,518	0,984	1,708	2,995	2,264
PT5	0,409	0,311	0,481	0,479	0,794
PT6	2,446	1,562	2,293	1,660	2,090
PT7	0,398	0,597	0,613	1,015	2,935
PT8	0,633	1,977	0,662	0,314	0,531
PT9	0,759	0,167	0,386	0,286	0,564
PT10	0,394	0,927	0,853	0,242	0,302
PT11	1,173	3,026	3,260	0,685	2,140
PT12	1,449	3,596	0,837	1,096	1,416

Tabel 3.3 Nilai Intensitas *Contrast*

<b>Nama Titik</b>	<b>170kHz</b>	<b>220kHz</b>	<b>270kHz</b>	<b>350kHz</b>	<b>400kHz</b>
PT1	6,800	7,833	819,267	10,333	3,767
PT2	1,760	71,364	2,733	2,588	0,767
PT3	1,500	1573,167	5,433	1,961	3,429
PT4	0,867	979,667	0,800	0,281	0,467
PT5	4,800	5,233	3,333	3,033	2,167
PT6	13,533	13,333	19,200	0,867	1,600
PT7	4,233	2,633	3,300	1,033	0,300
PT8	2,133	0,533	2,133	154,900	7,700
PT9	1,900	185,567	290,267	6,667	2783,926
PT10	116,900	3,867	4,367	8,367	5,533
PT11	231,600	19,667	7,200	49,200	9,600
PT12	0,982	0,105	1,833	1,088	1,018

Tabel 3.4 Nilai Intensitas *Correlation*

<b>Nama Titik</b>	<b>170kHz</b>	<b>220kHz</b>	<b>270kHz</b>	<b>350kHz</b>	<b>400kHz</b>
PT1	0,99989	0,99988	0,98597	0,99984	0,99994
PT2	0,99997	0,99885	0,99995	0,99996	0,99999
PT3	0,99998	0,97220	0,99997	0,99997	0,99995
PT4	0,99999	0,98411	0,99999	1,00000	0,99999
PT5	0,99992	0,99992	0,99995	0,99995	0,99997
PT6	0,99977	0,99978	0,99968	0,99999	0,99981
PT7	0,99993	0,99996	0,99995	0,99998	1,00000
PT8	0,99997	0,99999	0,99997	0,99744	0,99987
PT9	0,99997	0,99698	0,99535	0,99989	0,94849
PT10	0,99812	0,99994	0,99993	0,99987	0,99991
PT11	0,99596	0,99967	0,99988	0,99917	0,99984
PT12	0,99998	1,00000	0,99999	0,99998	0,99998

Tabel 3.5 Nilai Intensitas *Energy*

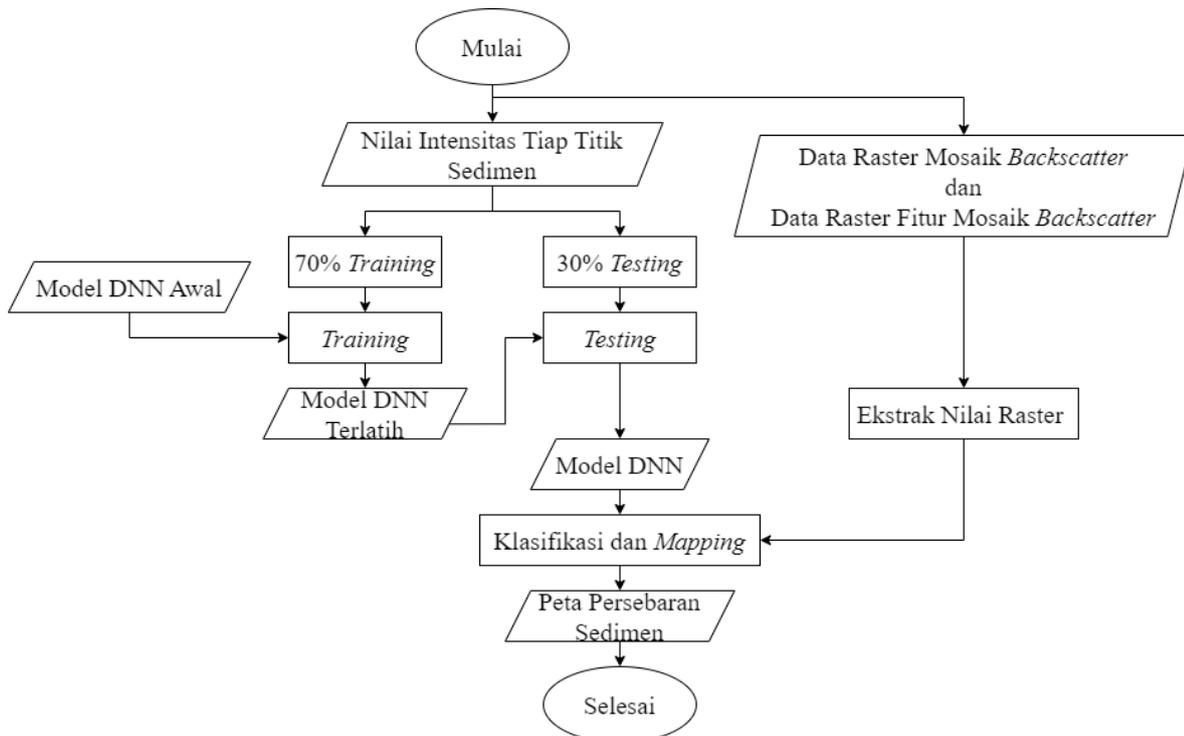
<b>Nama Titik</b>	<b>170kHz</b>	<b>220kHz</b>	<b>270kHz</b>	<b>350kHz</b>	<b>400kHz</b>
PT1	0,505	0,699	0,351	0,507	0,536
PT2	1,127	0,587	0,845	0,803	0,900
PT3	0,826	0,628	0,942	0,834	0,763
PT4	1,232	0,992	1,307	1,731	1,505
PT5	0,640	0,557	0,694	0,692	0,891
PT6	1,564	1,250	1,514	1,288	1,446
PT7	0,631	0,772	0,783	1,007	1,713
PT8	0,796	1,406	0,814	0,560	0,729

Tabel 3.5 Nilai Intensitas *Energy* (lanjutan)

Nama Titik	170kHz	220kHz	270kHz	350kHz	400kHz
PT9	0,871	0,408	0,621	0,535	0,751
PT10	0,628	0,963	0,923	0,492	0,550
PT11	1,083	1,739	1,806	0,828	1,463
PT12	1,204	1,896	1,080	1,047	1,190

### 3.3.4. Tahapan Desain Model dan Klasifikasi dengan DNN

Urutan tahapan desain model dan klasifikasi dengan DNN pada tugas akhir ini digambarkan dalam diagram alir sebagai berikut:



Gambar 3.11 Diagram Alir Desain Model dan Klasifikasi dengan DNN

Penjelasan diagram alir tahap pelaksanaan dalam penelitian yang akan dilakukan adalah sebagai berikut.

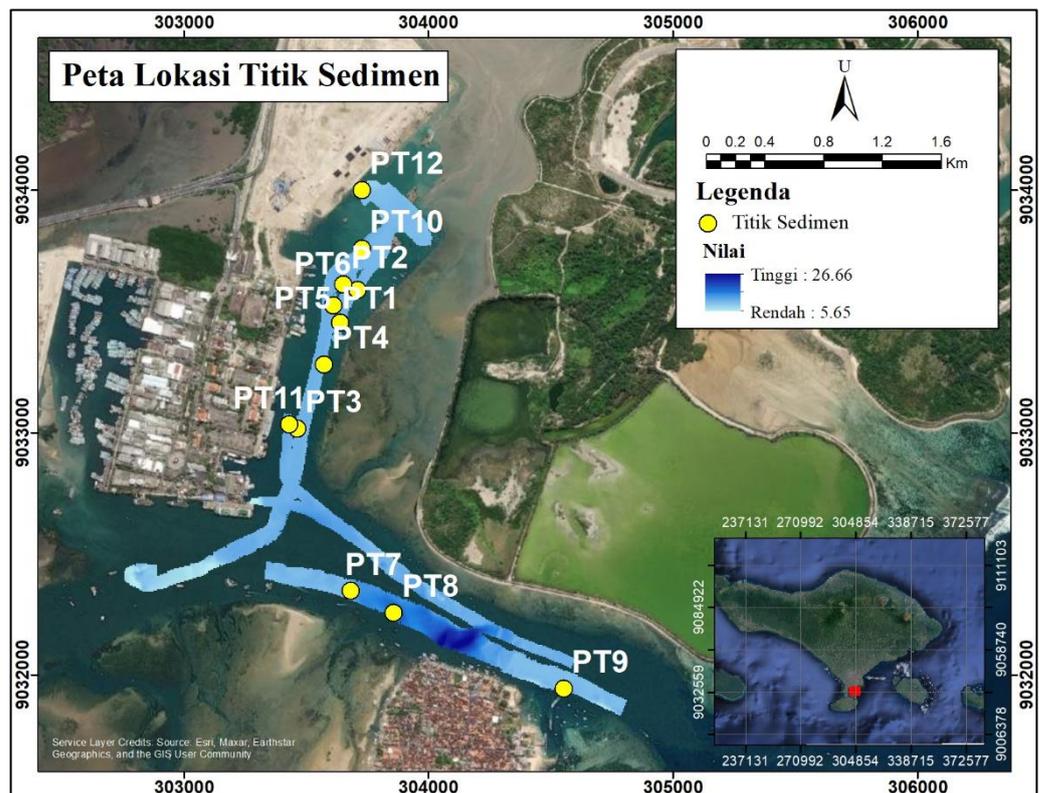
a. Pembagian Data *Training* dan Data *Testing*

Pembagian data *training* dan data *testing* dilakukan untuk menentukan mana data yang digunakan untuk training dan mana data yang digunakan untuk testing. Pembagian data ini dilakukan menggunakan persentase 70% untuk *training* dan 30% untuk *testing*. Persentase training lebih banyak karena dalam model DNN perlu dilakukan pembelajaran pola-pola model dengan baik. Pemilihan persentase ini karena data sampel sedimen yang dimiliki sedikit, sehingga perlu mengoptimalkan penggunaan data untuk *training* dan *testing* secara seimbang. Selain itu karena tidak semua data memiliki jumlah lebih dari satu titik, sehingga jika data tersebut digunakan sebagai *training* namun untuk testing menggunakan data yang berbeda, maka model tidak akan melakukan pembelajaran dengan baik. Terdapat 12 titik sedimen yang tersebar di sepanjang lajur pengukuran MBES. Dari ke duabelas titik sedimen tersebut, dibagi menjadi 9 titik sedimen untuk *training* dan 3 titik sedimen untuk *testing*. Berikut

merupakan data sampel sedimen yang digunakan dalam penelitian ini beserta gambar lokasi titik sedimen di sepanjang lajur MBES.

Tabel 3.6 Data Sampel Sedimen

No	<i>Easting</i> (m)	<i>Northing</i> (m)	Nama Titik	Kode Sedimen
1.	303610,03	9033525,81	PT1	MH
2.	303649,43	9033612,60	PT2	MH
3.	303461,40	9033018,47	PT3	MH
4.	303571,44	9033282,90	PT4	MH
5.	303636,99	9033454,99	PT5	ML
6.	303707,85	9033588,64	PT6	ML
7.	303680,01	9032349,00	PT7	SM
8.	303856,74	9032258,14	PT8	SM
9.	304552,01	9031945,99	PT9	SM
10.	303726,00	9033759,86	PT10	GM
11.	303429,01	9033035,01	PT11	CL
12.	303726,31	9034001,28	PT12	SW



Gambar 3.12 Peta Lokasi Titik Sedimen

Dalam pembagian data *training* dan data *testing* dilakukan secara manual. Khusus untuk data *testing* dipilih titik yang memiliki jenis sedimen lebih dari satu titik. Terdapat enam jenis sedimen yaitu pasir berlanau berkerikil karang dengan kode MH, lanau berpasir dengan kode ML, pasir berlanau dengan kode SM, kerikil karang berpasir berlanau dengan kode GM, lanau kelepungan lunak dengan kode CL, dan pasir berkerikil dengan kode SW. Dari ke enam jenis sedimen tersebut,

terdapat tiga jenis sedimen yang memiliki titik lebih dari satu. Sehingga dari ketiganya dipilih satu titik tiap jenis sedimen yang dijadikan sebagai data *testing*. Berikut merupakan hasil pembagian data *training* dan data *testing* yang akan digunakan pada pemodelan.

Tabel 3.7 Data *Training*

No	Nama Titik	Kode Sedimen	Jenis Sedimen
1.	PT1	MH	Pasir berlanau berkerikil karang
2.	PT2	MH	Pasir berlanau berkerikil karang
3.	PT3	MH	Pasir berlanau berkerikil karang
4.	PT5	ML	Lanau berpasir
5.	PT7	SM	Pasir berlanau
6.	PT9	SM	Pasir berlanau
7.	PT10	GM	Kerikil karang berpasir berlanau
8.	PT11	CL	Lanau kelepungan lunak
9.	PT12	SW	Pasir berkerikil

Tabel 3.8 Data *Testing*

No	Nama Titik	Kode Sedimen	Jenis Sedimen
1.	PT4	MH	Pasir berlanau berkerikil karang
2.	PT6	ML	Lanau berpasir
3.	PT8	SM	Pasir berlanau

b. Penentuan dan Pengujian Model DNN

Penentuan model DNN dilakukan dengan membuat *input layer*, *hidden layer*, dan *output layer*. Dalam setiap layer terdapat *neuron/node*. Untuk *input layer* dan *output layer* mempunyai node sama untuk setiap modelnya. Namun untuk *hidden layer* jumlah nodenya berbeda-beda untuk setiap modelnya. Hal ini dilakukan agar mendapatkan hasil akurasi terbaik dari berbagai bentuk model DNN. Model awal yang dibuat akan dilatih menggunakan data *training* dan di uji menggunakan data *testing* sehingga dihasilkan model DNN terbaik yang memiliki akurasi tinggi. Selain didapatkan nilai akurasi, saat proses ini juga dihasilkan data bobot (*weight*) dengan format \*.hdf5. Pemrograman ini dilakukan menggunakan Google Colab. Dalam pemrosesan DNN juga digunakan *library* TensorFlow dan Keras. Penggunaan *library* ini bertujuan untuk memudahkan penulis selama proses pengolahan DNN.

c. Ekstrak Nilai Raster

Nilai intensitas untuk tiap titik pada lajur MBES dibutuhkan pada proses klasifikasi. Nilai ini didapat dari data raster mosaik *backscatter* dan data raster fitur mosaik *backscatter*. Ekstrak nilai dilakukan menggunakan software ArcGIS. Dalam ekstraksi nilai ini digunakan salah satu data raster yang diubah formatnya menjadi vektor (*point*). Data raster mosaik *backscatter* frekuensi 350 kHz dipilih untuk menjadi titik utama ekstraksi nilai data raster lain. Terdapat 388.799 titik yang digunakan untuk ekstraksi nilai data raster lain, sehingga nantinya semua data raster akan memiliki jumlah titik yang sama.

d. Klasifikasi dan *Mapping*

Proses klasifikasi jenis sedimen dilakukan menggunakan data nilai intensitas yang telah di ekstrak dari tiap titik di lajur MBES. Data ini akan digabungkan dengan data bobot yang didapatkan dari hasil model DNN. Dari proses ini akan dihasilkan rentang nilai satu sampai enam sesuai jumlah jenis sedimen pada data sampel sedimen. Rentang nilai tersebut akan di plot di ArcGIS sehingga menghasilkan sebaran sedimen. Selanjutnya akan dilakukan layout untuk sebaran sedimen dan dihasilkan peta sebaran sedimen Pelabuhan Benoa.

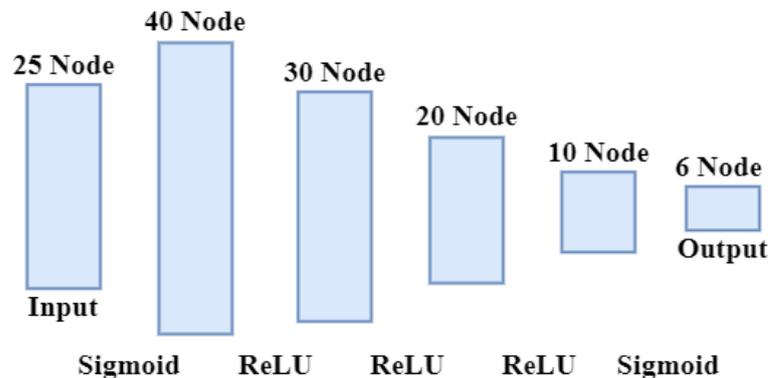
***“Halaman ini sengaja dikosongkan”***

## BAB IV HASIL DAN PEMBAHASAN

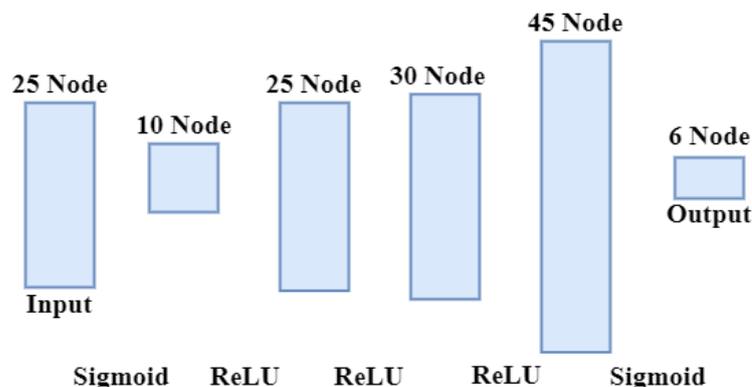
### 4.1 Model Dense Neural Network

*Dense Neural Network* merupakan salah satu metode dari *Neural Network* yang paling sederhana. Dalam metode DNN, setiap neuron dalam lapisan tertentu terhubung dengan setiap neuron dalam lapisan berikutnya. Penyusunan model *Dense Neural Network* dilakukan menggunakan bahasa pemrograman Python yang diolah secara *online* dengan platform *cloud* yang disediakan oleh Google yaitu Google Colab. Pemilihan Google Colab sebagai platform untuk pembuatan model DNN adalah karena Google Colab dapat diakses secara gratis melalui web. Selain itu, Google Colab terintegrasi dengan layanan Google lainnya seperti Google Drive. Google Colab juga menyimpan secara otomatis pemrograman yang dikerjakan, sehingga memungkinkan pemulihan jika terjadi kegagalan sistem atau kesalahan pengguna. Untuk menerapkan *Dense Neural Network* dengan Python, digunakan *library deep learning* TensorFlow dan Keras. *Library* ini membantu dalam pembuatan pemrograman untuk model DNN sehingga model yang dibuat dapat bekerja dengan baik.

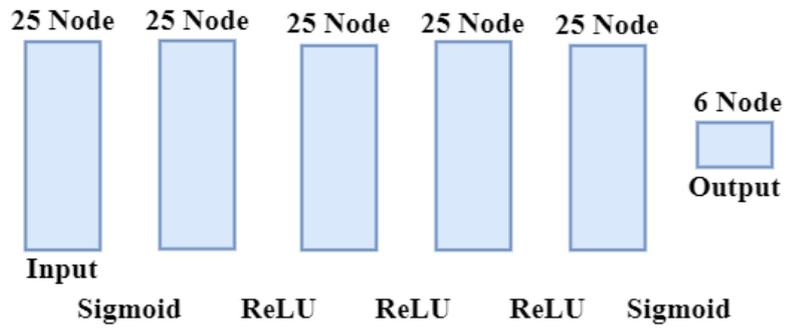
Model *Dense Neural Network* tersusun atas tiga lapisan atau *layer* yaitu *input layer*, *hidden layer*, dan *output layer*. Pembuatan model DNN pada penelitian ini dilakukan dengan pendekatan *trial and error*. Pendekatan *trial and error* dilakukan dengan mencoba berbagai kombinasi struktur model DNN. Kombinasi yang dapat diubah seperti jumlah *hidden layer*, jumlah *node* pada *hidden layer* (bentuk model), fungsi aktivasi, dan *epoch*. Pendekatan ini menggambarkan pembuatan model, melatihnya, dan mengevaluasi hasilnya. Jika kinerja dari model tidak memenuhi atau belum menghasilkan hasil yang baik, akan dilakukan penyesuaian dan percobaan lagi dengan konfigurasi atau kombinasi yang berbeda. Untuk mendapatkan model dengan hasil paling baik, penulis membuat enam model berbeda sebagai berikut.



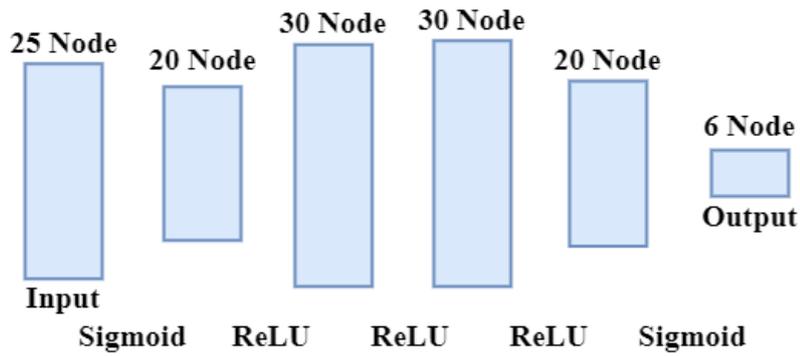
Gambar 4.1 Model 1



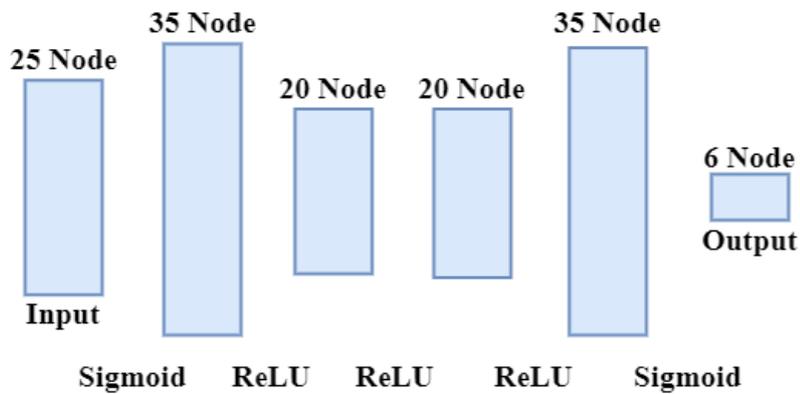
Gambar 4.2 Model 2



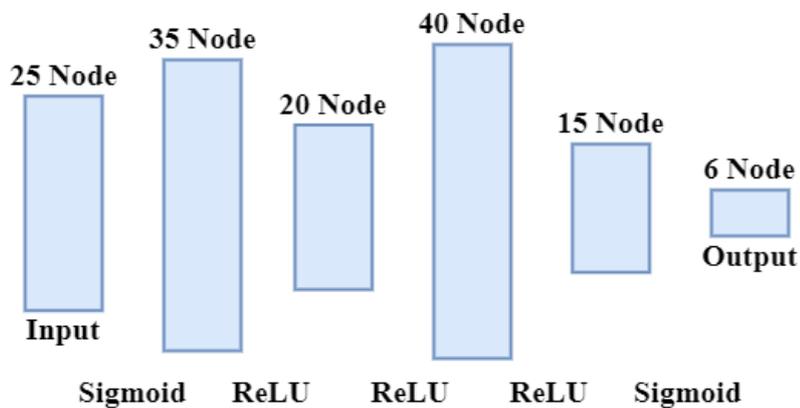
Gambar 4.3 Model 3



Gambar 4.4 Model 4



Gambar 4.5 Model 5



Gambar 4.6 Model 6

Pada penelitian ini, *input layer* berisi data nilai intensitas tiap titik sedimen sedangkan *output layer* berisi hasil pengolahan *Dense Neural Network*. Sedangkan untuk *hidden layer* dapat disesuaikan hingga mendapatkan susunan model yang baik. Pemodelan DNN di atas menggunakan beberapa hal yang sama seperti fungsi aktivasi, jumlah *neuron* di *input layer*, jumlah *hidden layer* dan jumlah *neuron* di *output layer*. Fungsi aktivasi yang digunakan untuk tiap modelnya yaitu Sigmoid dan ReLU (*Rectified Linear Unit*). Fungsi aktivasi sigmoid digunakan karena fungsi sigmoid menghasilkan output antara nol dan satu, yang dapat diinterpretasikan sebagai probabilitas terhadap kelas tertentu. Hal ini cocok untuk klasifikasi, dimana tujuannya untuk mendapatkan probabilitas prediksi untuk setiap kelas. Sedangkan penggunaan fungsi aktivasi ReLU karena fungsi ReLU memungkinkan model DNN untuk mempelajari hubungan yang kompleks dan non-linear antara fitur input dan keluaran. Hal ini penting dalam pengolahan klasifikasi jenis sedimen, di mana pola dan karakteristik data mungkin memiliki hubungan yang kompleks dan tidak linear. Penempatan fungsi aktivasi sigmoid berada di awal dan di akhir model. Fungsi aktivasi Sigmoid berada di awal karena pada nilai intensitas yang dijadikan *input* memiliki nilai yang negatif, dengan menggunakan Sigmoid nilai tersebut akan dinormalisasi dengan mengubah menjadi rentang nol sampai satu. Hal ini dikarenakan dalam *image processing* tidak ada nilai negatif. Sedangkan untuk fungsi aktivasi Sigmoid di akhir karena fungsi aktivasi ini dapat menghasilkan nilai probabilitas untuk setiap kelas secara terpisah. Untuk fungsi aktivasi ReLU berada di tengah model karena fungsi aktivasi ini dapat membantu model melakukan pembelajaran yang lebih kompleks. Fungsi ReLU memiliki keluaran 0 untuk nilai masukan negatif dan mengalirkan nilai masukan positif tanpa perubahan. Oleh karena itu, ReLU cenderung meninggalkan informasi yang lebih banyak dari data yang memiliki nilai positif.

Dalam model di atas menggunakan jumlah *neuron/node* yang sama untuk *input layer* yaitu 25. Jumlah ini sesuai dengan jumlah *data input* yaitu 5 data mosaik *backscatter* dan 20 data fitur mosaik *backscatter*. Jumlah *hidden layer* yang digunakan pada model juga dibuat sama yaitu berjumlah empat *hidden layer*. Hal ini bertujuan agar lebih mudah untuk membandingkan kinerja antara model satu dengan yang lainnya serta agar dapat mengamati efek dari perubahan struktur model DNN. Selain jumlah *neuron input layer* dan jumlah *hidden layer*, jumlah *neuron* pada *output layer* juga sama dikarenakan dalam penelitian ini akan dihasilkan 6 jenis sedimen sehingga pada *output layer* juga dituliskan 6 sesuai dengan jumlah jenis sedimen. Pada tiap model memiliki jumlah *neuron/node* berbeda untuk *hidden layer*, hal ini dikarenakan untuk mendapat model yang baik diperlukan percobaan yang banyak sehingga dari beberapa percobaan tersebut akan didapatkan hasil yang paling baik. Perbedaan jumlah *node* dalam setiap model didasarkan pada jumlah total parameter. Total parameter adalah jumlah dari semua parameter yang dihasilkan pada setiap *layer*. Dalam hal ini, untuk setiap pemodelan memiliki jumlah total parameter yang tidak terlalu jauh nilainya.

Dapat dilihat untuk Gambar 4.1 digunakan bentuk mengerucut ke bawah yang artinya pada *layer* pertama menggunakan jumlah *node* paling banyak dan semakin mengecil untuk *layer* selanjutnya. Total parameter untuk Pemodelan 1 adalah 3.166. Untuk pemodelan 2 dapat dilihat seperti Gambar 4.2 yang menggunakan jumlah *node* yang sama dalam setiap *layer*. Total parameter untuk Pemodelan 2 adalah 2.986. Gambar 4.3 menunjukkan bentuk model untuk pemodelan 3, dimana pada pemodelan ini jumlah *node* di *layer* pertama sedikit hingga di *layer* berikutnya ditambahkan jumlah *node* yang lebih banyak. Total parameter untuk Pemodelan 3 adalah 2.756. Pemodelan 4 ditunjukkan oleh Gambar 4.4, dimana untuk jumlah *node* yang digunakan pada *layer* pertama dan *layer* ke empat adalah sama dan untuk *layer* kedua dan *layer* ketiga adalah sama. Total parameter untuk Pemodelan 4 adalah 2.826. Dalam Gambar 4.5, dimana untuk jumlah *node* yang digunakan pada *layer* pertama dan *layer* ke empat adalah sama dan untuk *layer* kedua dan *layer* ketiga adalah sama. Total parameter untuk Pemodelan 5 adalah

3.001. Perbedaan pemodelan 4 dan pemodelan 5 ada pada jumlah *node* di setiap *layer*, jika pemodelan 4 *layer* yang berada di luar memiliki jumlah *node* lebih sedikit daripada *layer* yang di tengah, sedangkan pemodelan 5 *layer* yang berada di luar memiliki jumlah *node* lebih banyak daripada *layer* yang di tengah. Sedangkan untuk Pemodelan 6 yang dapat dilihat dalam Gambar 4.6, memiliki jumlah *node* yang acak di setiap *layer*. Total parameter untuk Pemodelan 6 adalah 3.181. Pembuatan 6 model yang berbeda ini bertujuan untuk memudahkan penulis dalam mencari hasil nilai akurasi yang terbaik.

Dalam model DNN juga digunakan *hyperparameter optimizer*, *loss function*, dan *metrics*. *Optimizer* Adam digunakan dalam pengolahan ini karena dalam pengolahan klasifikasi jenis sedimen, *learning rate* yang tepat sangat penting agar model dapat mengkonvergensi dengan baik. *Optimizer* Adam dapat menyesuaikan *learning rate* untuk setiap parameter secara adaptif berdasarkan momentum gradien sebelumnya dan perubahan dalam gradien. Hal ini membantu dalam mencapai konvergensi yang lebih cepat dan meningkatkan kinerja model. Dengan demikian, penggunaan *optimizer* Adam dalam pengolahan ini dapat membantu memperbaiki efisiensi dan performa model dalam klasifikasi jenis sedimen. Untuk *hyperparameter loss function* menggunakan *categorical crossentropy*. *Categorical crossentropy* adalah fungsi yang dirancang secara khusus untuk menangani masalah klasifikasi multikelas di mana setiap sampel hanya dapat termasuk dalam satu kelas. Ketika digunakan dalam klasifikasi jenis sedimen, di mana terdapat beberapa kategori sedimen yang berbeda, *categorical crossentropy* menjadi pilihan yang tepat untuk mengukur sejauh mana model DNN dapat memprediksi dengan akurat kelas yang tepat. Sedangkan untuk *hyperparameter metrics* digunakan akurasi karena hal ini sesuai dengan tujuan dari penelitian ini yaitu menghasilkan klasifikasi dari model DNN dengan akurasi paling tinggi. Dengan menggunakan metrik akurasi, akan dapat langsung mengetahui persentase prediksi yang benar dari seluruh sampel, yang merupakan informasi penting dalam mengevaluasi kinerja model.

#### 4.2 Hasil Evaluasi Model

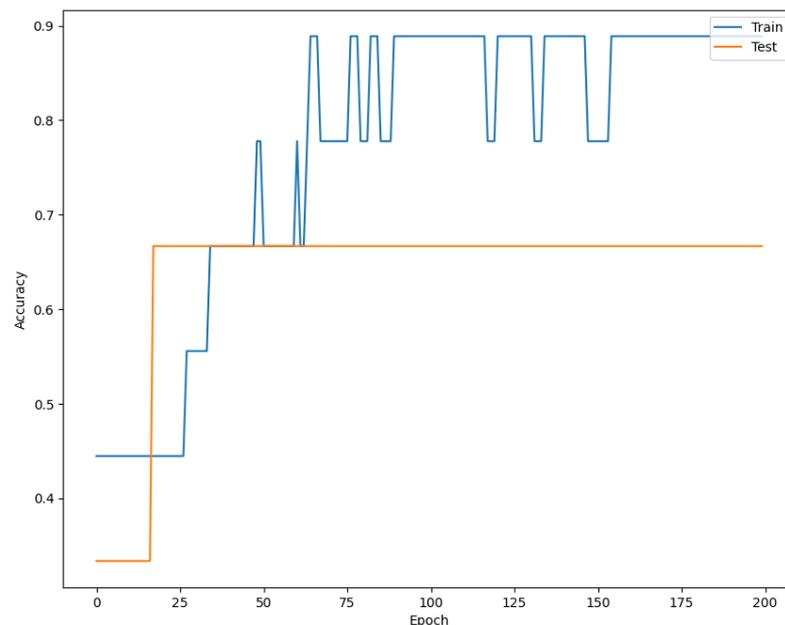
Hasil evaluasi model adalah hasil dari keakuratan pemodelan dalam melatih model yang selanjutnya akan dibuat klasifikasi. Dalam hasil ini terdapat nilai akurasi untuk *training* dan untuk *testing*. Semakin tinggi nilai akurasinya maka model yang dibuat semakin baik. Hasil akurasi untuk *training* dan *testing* harus baik untuk keduanya karena bisa saja hasil yang didapatkan berupa hasil *training* yang baik namun hasil *testing* kurang baik maupun sebaliknya. Jika hasil akurasi *training* tinggi namun hasil akurasi *testing* rendah, maka model DNN akan mengalami *underfitting*. Sebaliknya jika hasil akurasi *training* rendah namun hasil akurasi *testing* tinggi, maka model DNN akan mengalami *overfitting*. Hasil akurasi yang *underfitting* dan *overfitting* dapat menghambat kemampuan model DNN dalam melakukan generalisasi yang baik pada data. Oleh karena itu, sangat penting untuk mencari nilai yang seimbang ataupun tidak terlalu jauh agar hasil klasifikasi yang diperoleh akurat. Berikut merupakan hasil akurasi untuk setiap pemodelan.

Tabel 4.1 Hasil Akurasi Model DNN

No	Pemodelan	Epoch	Akurasi Training	Akurasi Testing
1.	Model 1	200	88%	66%
2.	Model 2		77%	66%
3.	Model 3		88%	33%
4.	Model 4		88%	66%
5.	Model 5		88%	100%
6.	Model 6		88%	33%

Iterasi atau *epoch* yang digunakan dalam semua model sama yaitu 200 kali. Proses iterasi ini berulang seiring dengan jumlah iterasi yang ditentukan. Jumlah *epoch* yang lebih besar memungkinkan model untuk melihat dan memproses data *training* lebih banyak, yang dapat membantu dalam pemahaman pola yang lebih baik dan peningkatan kinerja model. Pemilihan jumlah *epoch* yang tepat merupakan bagian dari penting dalam melatih model DNN. Terlalu sedikit *epoch* dapat menyebabkan model tidak memiliki cukup waktu untuk belajar, sementara terlalu banyak *epoch* dapat menyebabkan *overfitting*, di mana model terlalu mempelajari data pelatihan dengan cermat dan tidak dapat generalisasi dengan baik pada data baru. Untuk itu penulis memilih *epoch* 200 karena tidak sedikit namun juga tidak terlalu banyak.

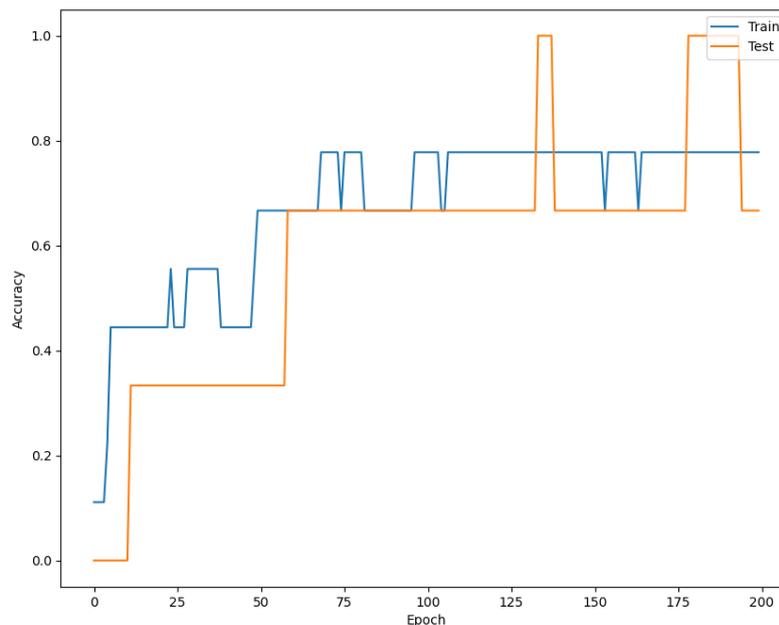
Jika dilihat dari hasil akurasi pada Tabel 4.1, pemodelan 1, pemodelan 3, pemodelan 4, pemodelan 5, dan pemodelan 6 memiliki hasil akurasi untuk *training* sama yaitu 88%, namun untuk akurasi *testing* pemodelan 5 memiliki hasil akurasi paling tinggi yaitu 100%. Akurasi *testing* 100% sangat mungkin terjadi karena keterbatasan data sampel yang ada. Jika jumlah sampel pada data *testing* sangat terbatas, terutama jika data tersebut sangat mewakili kelas-kelas yang ada, maka ada kemungkinan bahwa model DNN hanya mempelajari sampel-sampel tersebut dan dapat menghasilkan akurasi yang tinggi. Akurasi *testing* untuk pemodelan lain memiliki nilai yang cukup jauh dari 100%. Pada pemodelan 1, pemodelan 2, dan pemodelan 4 memiliki nilai akurasi *testing* sebesar 66%. Sedangkan pemodelan 3 dan pemodelan 6 memiliki nilai akurasi *testing* sebesar 33%. Untuk itu, penelitian ini menggunakan pemodelan 5 sebagai model DNN yang akan digunakan untuk proses klasifikasi. Pemilihan model 5 ini karena model inilah yang memiliki nilai akurasi paling tinggi. Model dengan akurasi *training* dan akurasi *testing* tinggi menunjukkan kemampuan yang baik dalam mempelajari pola dari data *training* ke data yang belum pernah dilihat sebelumnya. Selain itu, model dengan akurasi tinggi cenderung memberikan prediksi yang lebih akurat pada data baru. Berikut merupakan grafik yang menunjukkan akurasi *training* dan akurasi *testing* untuk setiap model.



Gambar 4.7 Grafik Hasil Akurasi Model 1

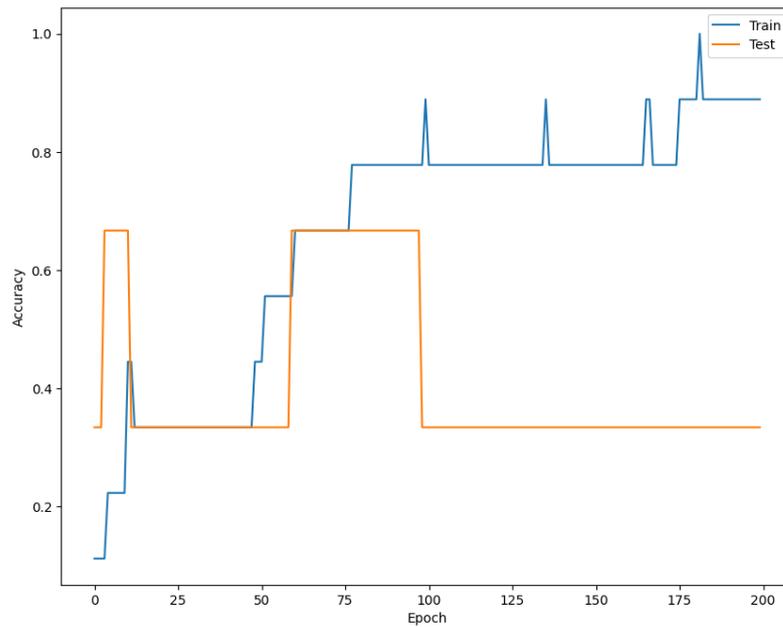
Berdasarkan informasi pada Gambar 4.7 yang merupakan grafik hasil akurasi model 1, sumbu X menunjukkan jumlah *epoch* yang digunakan dalam model DNN yaitu 200 kali. Sedangkan sumbu Y menunjukkan nilai *Accuracy* yang didapatkan dalam model DNN untuk setiap *epoch*. Pada data *training* yang ditandai dengan garis berwarna biru, nilai akurasi untuk

*epoch* 0 sebesar 0,45 sedangkan pada data *testing* yang ditandai dengan garis berwarna oranye, nilai akurasi sebesar 0,33. Dalam grafik tersebut, data *training* mengalami peningkatan nilai akurasi sampai dengan 0,79 pada *epoch* ke 50. Namun setelah itu, nilai akurasi cenderung naik turun hingga pada *epoch* ke 200 memiliki nilai akurasi sebesar 0,88. Untuk data *testing*, pada *epoch* ke 20 mengalami peningkatan hingga mencapai nilai akurasi 0,66. Setelah itu, nilai akurasi cenderung konstan sampai dengan *epoch* ke 200 dengan nilai akurasi sebesar 0,66. Nilai akurasi pada grafik model 1 untuk data *training* dan data *testing* memiliki selisih yang cukup besar. Grafik akurasi pada Gambar 4.7 memiliki pola yang tegak (tidak halus) karena nilai akurasi diambil dari nilai benar atau tidak. Sehingga akan dihasilkan pola yang tegak seperti pada gambar tersebut.



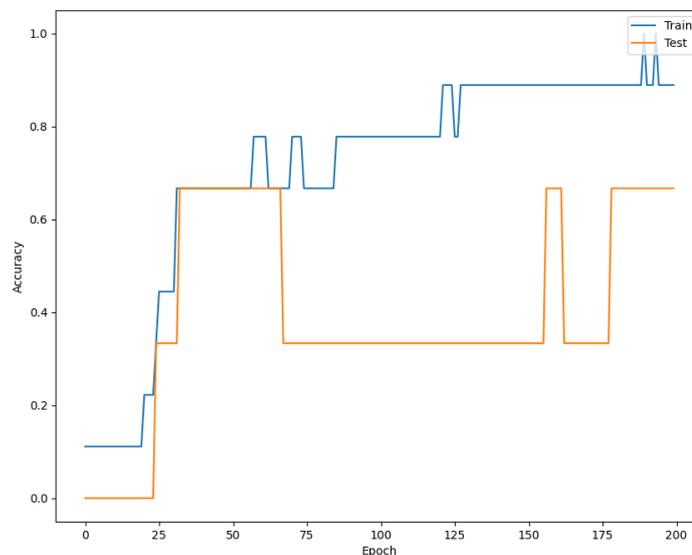
Gambar 4.8 Grafik Hasil Akurasi Model 2

Berdasarkan informasi pada Gambar 4.8 yang merupakan grafik hasil akurasi model 2, sumbu X menunjukkan jumlah *epoch* yang digunakan dalam model DNN yaitu 200 kali. Sedangkan sumbu Y menunjukkan nilai *Accuracy* yang didapatkan dalam model DNN untuk setiap *epoch*. Pada data *training* yang ditandai dengan garis berwarna biru, nilai akurasi untuk *epoch* 0 sebesar 0,12 sedangkan pada data *testing* yang ditandai dengan garis berwarna oranye, nilai akurasi sebesar 0,0. Dalam grafik tersebut, data *training* mengalami peningkatan nilai akurasi sampai dengan 0,44 pada *epoch* ke 5. Namun setelah itu, nilai akurasi cenderung naik turun hingga pada *epoch* ke 200 memiliki nilai akurasi sebesar 0,77. Untuk data *testing*, pada *epoch* ke 10 mengalami peningkatan hingga mencapai nilai akurasi 0,66. Setelah itu, nilai akurasi cenderung naik turun dari 0,66 hingga 1,0. Pada *epoch* 200, nilai akurasi data *testing* sebesar 0,66. Nilai akurasi pada grafik model 2 untuk data *training* dan data *testing* memiliki selisih yang cukup kecil. Grafik akurasi pada Gambar 4.8 memiliki pola yang tegak (tidak halus) karena nilai akurasi diambil dari nilai benar atau tidak. Sehingga akan dihasilkan pola yang tegak seperti pada gambar tersebut.



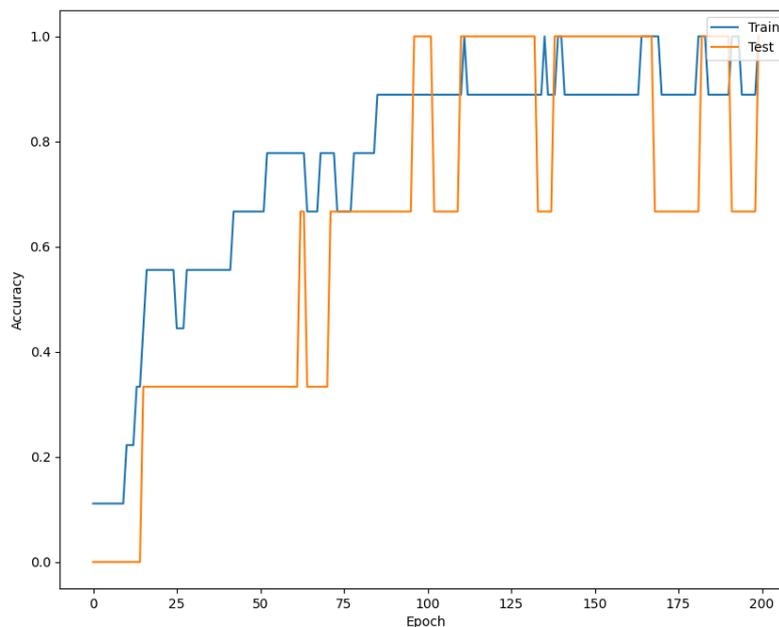
Gambar 4.9 Grafik Hasil Akurasi Model 3

Berdasarkan informasi pada Gambar 4.9 yang merupakan grafik hasil akurasi model 3, sumbu X menunjukkan jumlah *epoch* yang digunakan dalam model DNN yaitu 200 kali. Sedangkan sumbu Y menunjukkan nilai *Accuracy* yang didapatkan dalam model DNN untuk setiap *epoch*. Pada data *training* yang ditandai dengan garis berwarna biru, nilai akurasi untuk *epoch* 0 sebesar 0,12 sedangkan pada data *testing* yang ditandai dengan garis berwarna oranye, nilai akurasinya sebesar 0,33. Dalam grafik tersebut, data *training* mengalami peningkatan nilai akurasi sampai dengan 0,45 pada *epoch* ke 15. Namun setelah itu, nilai akurasinya cenderung naik turun hingga pada *epoch* ke 200 memiliki nilai akurasi sebesar 0,88. Untuk data *testing*, pada *epoch* ke 3 mengalami peningkatan hingga mencapai nilai akurasi 0,66. Setelah itu, nilai akurasinya cenderung naik turun hingga *epoch* 200 dengan nilai akurasi data *testing* sebesar 0,33. Nilai akurasi pada grafik model 3 untuk data *training* dan data *testing* memiliki selisih yang besar. Grafik akurasi pada Gambar 4.9 memiliki pola yang tegak (tidak halus) karena nilai akurasi diambil dari nilai benar atau tidak. Sehingga akan dihasilkan pola yang tegak seperti pada gambar tersebut.



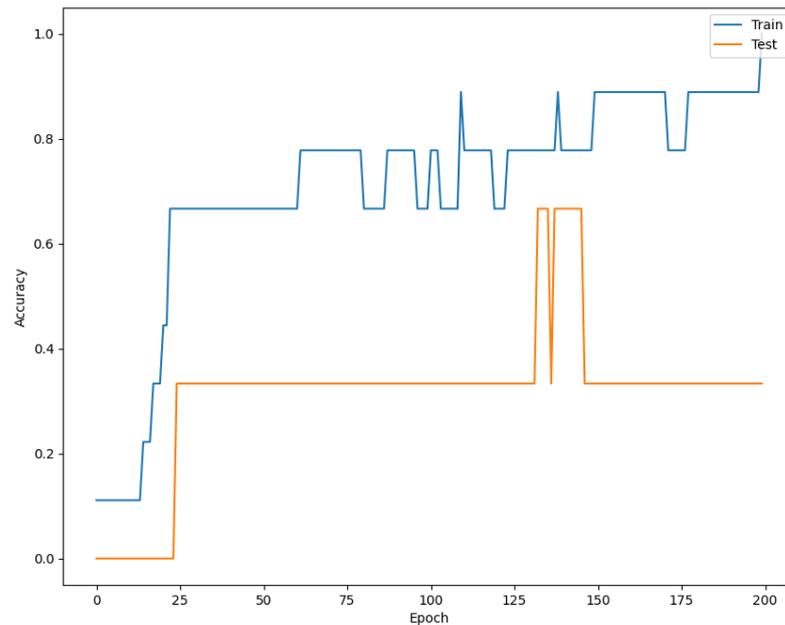
Gambar 4.10 Grafik Hasil Akurasi Model 4

Berdasarkan informasi pada Gambar 4.10 yang merupakan grafik hasil akurasi model 4, sumbu X menunjukkan jumlah *epoch* yang digunakan dalam model DNN yaitu 200 kali. Sedangkan sumbu Y menunjukkan nilai *Accuracy* yang didapatkan dalam model DNN untuk setiap *epoch*. Pada data *training* yang ditandai dengan garis berwarna biru, nilai akurasi untuk *epoch* 0 sebesar 0,12 sedangkan pada data *testing* yang ditandai dengan garis berwarna oranye, nilai akurasinya sebesar 0,0. Dalam grafik tersebut, data *training* mengalami peningkatan nilai akurasi sampai dengan 0,79 pada *epoch* ke 60. Namun setelah itu, nilai akurasinya cenderung naik turun hingga pada *epoch* ke 200 memiliki nilai akurasi sebesar 0,88. Untuk data *testing*, pada *epoch* ke 25 mengalami peningkatan hingga mencapai nilai akurasi 0,66. Setelah itu, nilai akurasinya cenderung naik turun hingga *epoch* 200 dengan nilai akurasi data *testing* sebesar 0,66. Nilai akurasi pada grafik model 3 untuk data *training* dan data *testing* memiliki selisih yang cukup besar. Grafik akurasi pada Gambar 4.10 memiliki pola yang tegak (tidak halus) karena nilai akurasi diambil dari nilai benar atau tidak. Sehingga akan dihasilkan pola yang tegak seperti pada gambar tersebut.



Gambar 4.11 Grafik Hasil Akurasi Model 5

Berdasarkan informasi pada Gambar 4.11 yang merupakan grafik hasil akurasi model 5, sumbu X menunjukkan jumlah *epoch* yang digunakan dalam model DNN yaitu 200 kali. Sedangkan sumbu Y menunjukkan nilai *Accuracy* yang didapatkan dalam model DNN untuk setiap *epoch*. Pada data *training* yang ditandai dengan garis berwarna biru, nilai akurasi untuk *epoch* 0 sebesar 0,12 sedangkan pada data *testing* yang ditandai dengan garis berwarna oranye, nilai akurasinya sebesar 0,0. Dalam grafik tersebut, data *training* mengalami peningkatan nilai akurasi sampai dengan 0,56 pada *epoch* ke 20. Namun setelah itu, nilai akurasinya cenderung naik turun hingga pada *epoch* ke 200 memiliki nilai akurasi sebesar 0,88. Untuk data *testing*, pada *epoch* ke 15 mengalami peningkatan hingga mencapai nilai akurasi 0,66. Setelah itu, nilai akurasinya cenderung naik turun hingga *epoch* 200 dengan nilai akurasi data *testing* sebesar 1,0. Nilai akurasi pada grafik model 5 untuk data *training* dan data *testing* memiliki selisih yang cukup kecil. Grafik akurasi pada Gambar 4.11 memiliki pola yang tegak (tidak halus) karena nilai akurasi diambil dari nilai benar atau tidak. Sehingga akan dihasilkan pola yang tegak seperti pada gambar tersebut.



Gambar 4.12 Grafik Hasil Akurasi Model 6

Berdasarkan informasi pada Gambar 4.12 yang merupakan grafik hasil akurasi model 6, sumbu X menunjukkan jumlah *epoch* yang digunakan dalam model DNN yaitu 200 kali. Sedangkan sumbu Y menunjukkan nilai *Accuracy* yang didapatkan dalam model DNN untuk setiap *epoch*. Pada data *training* yang ditandai dengan garis berwarna biru, nilai akurasi untuk *epoch* 0 sebesar 0,12 sedangkan pada data *testing* yang ditandai dengan garis berwarna oranye, nilai akurasinya sebesar 0,0. Dalam grafik tersebut, data *training* mengalami peningkatan nilai akurasi sampai dengan 0,79 pada *epoch* ke 70. Namun setelah itu, nilai akurasinya cenderung naik turun hingga pada *epoch* ke 200 memiliki nilai akurasi sebesar 0,88. Untuk data *testing*, pada *epoch* ke 25 mengalami peningkatan hingga mencapai nilai akurasi 0,33. Setelah itu, nilai akurasinya cenderung konstan, namun nilai ini kemudian naik turun hingga *epoch* 200 dengan nilai akurasi data *testing* sebesar 0,33. Nilai akurasi pada grafik model 6 untuk data *training* dan data *testing* memiliki selisih yang besar. Grafik akurasi pada Gambar 4.12 memiliki pola yang tegak (tidak halus) karena nilai akurasi diambil dari nilai benar atau tidak. Sehingga akan dihasilkan pola yang tegak seperti pada gambar tersebut.

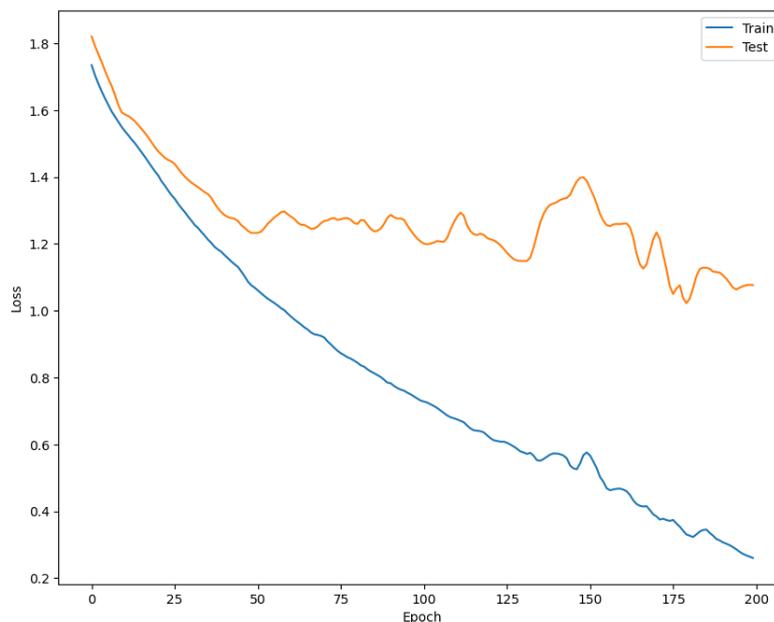
Dalam hasil evaluasi model, selain terdapat hasil nilai akurasi, juga terdapat hasil nilai *loss* untuk *training* dan *testing*. Nilai *loss* pada model DNN menunjukkan ukuran sejauh mana model tidak cocok dengan data pelatihan. Jumlah nilai *loss* digunakan untuk mengukur kesalahan prediksi model terhadap data yang diberikan. Jika nilai *loss* semakin kecil, maka semakin akurat prediksi model. Oleh karena itu, nilai *loss* yang mendekati nol memiliki kinerja model yang baik. Nilai *loss* pada penelitian ini dihitung menggunakan fungsi *loss* atau *loss function categorical crossentropy*. Penggunaan *loss function* ini dipilih karena sesuai dengan tujuan dari penelitian ini yaitu untuk klasifikasi. *Categorical crossentropy* pada dasarnya mengukur perbedaan antara distribusi probabilitas prediksi dan distribusi probabilitas label sebenarnya. Nilai *loss* dapat dianalisis untuk memahami seberapa baik model DNN dalam melakukan prediksi pada data *training*. Jika nilai *loss* terus menurun selama proses pelatihan, hal ini menunjukkan bahwa model semakin cocok dengan data *training*. Namun, tidak hanya memperhatikan nilai *loss* pada data *training*, tetapi juga perlu melihat kinerja dari *loss function* untuk data *testing*. Berikut merupakan hasil *loss* untuk setiap pemodelan.

Tabel 4.2 Hasil *Loss* Model DNN

No	Pemodelan	Epoch	<i>Loss Training</i>	<i>Loss Testing</i>
1.	Model 1	200	0,94	1,25
2.	Model 2		1,02	1,35
3.	Model 3		0,37	1,01
4.	Model 4		0,28	0,58
5.	Model 5		0,34	0,63
6.	Model 6		0,28	1,27

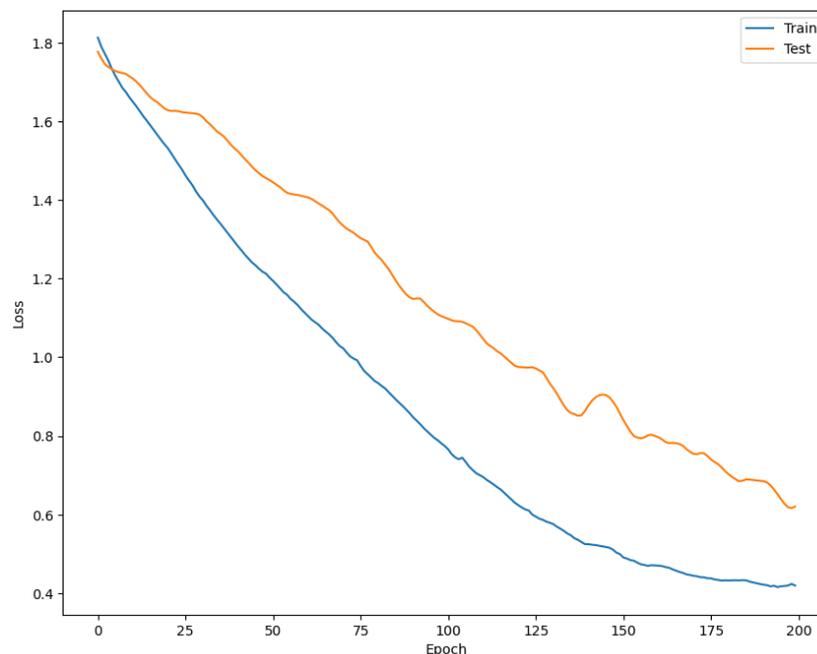
Iterasi atau *epoch* yang digunakan dalam semua model sama yaitu 200 kali. Jika dilihat pada Tabel 4.2, nilai *loss* untuk setiap modelnya bervariasi. Untuk nilai *loss* pada *training*, pemodelan 4 dan pemodelan 6 memiliki nilai *loss* paling kecil yaitu 0,28. Sehingga pemodelan 4 dan pemodelan 6 memiliki kinerja model paling baik untuk data *training*. Nilai *loss* pada pemodelan lain yang memiliki kinerja cukup baik yaitu pemodelan 3 dan pemodelan 5 dengan nilai *loss* masing-masing 0,37 dan 0,34. Selanjutnya diikuti pemodelan 1 dengan nilai *loss* 0,94 dan terakhir pemodelan 2 dengan nilai *loss* paling besar untuk data *training* yaitu sebesar 1,02. Sedangkan nilai *loss* untuk data *testing*, pemodelan 4 memiliki nilai *loss* paling kecil, diikuti dengan pemodelan 5 sebesar 0,63, pemodelan 3 sebesar 1,01, pemodelan 1 sebesar 1,25, pemodelan 6 sebesar 1,27, dan terakhir pemodelan 2 dengan nilai *loss* sebesar 1,35. Dari semua hasil nilai *loss* untuk *training* maupun *testing*, pemodelan 4 memiliki kinerja paling baik sedangkan pemodelan 2 memiliki kinerja paling buruk. Namun, dapat dilihat pada Tabel 4.2 untuk pemodelan 5 memiliki nilai *loss* yang kecil untuk keduanya. Hal ini membuktikan bahwa pemodelan 5 selain memiliki nilai akurasi paling tinggi, pemodelan ini juga memiliki nilai *loss* yang cukup kecil yang menandakan kinerja model DNN untuk pemodelan 5 bekerja dengan baik. Untuk itu, dapat disimpulkan bahwa model 5 memiliki hasil paling baik diantara model yang lain dari segi akurasi dan kinerja model.

Nilai *loss* yang dihasilkan oleh tiap model juga memiliki grafik yang menyatakan nilai *loss* untuk setiap *epoch*. Berikut merupakan grafik yang menunjukkan nilai *loss* untuk data *training* dan *testing* untuk setiap model.



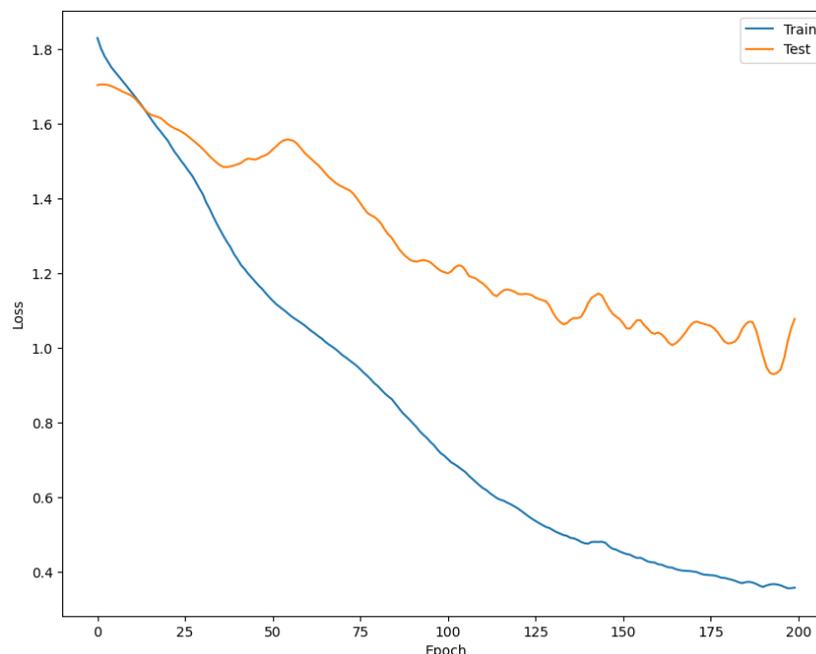
Gambar 4.13 Grafik Hasil *Loss* Model 1

Berdasarkan informasi pada Gambar 4.13 yang merupakan grafik hasil *loss* model 1, sumbu X menunjukkan jumlah *epoch* yang digunakan dalam model DNN yaitu 200 kali. Sedangkan sumbu Y menunjukkan nilai *loss* yang didapatkan dalam model DNN untuk setiap *epoch*. Pada awal dimulainya iterasi, nilai *loss* untuk *training* dan *testing* bernilai tinggi karena model belum menyesuaikan dengan baik terhadap data. Pada data *training* yang ditandai dengan garis berwarna biru, nilai *loss* untuk *epoch* 0 sebesar 1,74 sedangkan pada data *testing* yang ditandai dengan garis berwarna oranye, nilai *loss*nya sebesar 1,82. Dalam grafik tersebut, data *training* mengalami penurunan secara terus-menerus untuk nilai *loss* sampai dengan 0,26 pada *epoch* ke 200. Hal tersebut menunjukkan bahwa model secara efektif mengenali pola-pola dalam data *training* dan berhasil mengurangi kesalahan prediksi seiring berjalannya iterasi. Untuk data *testing*, nilai *loss* mengalami penurunan, namun pada *epoch* ke 50 hasil nilai *loss* tidak mengalami penurunan melainkan naik turun hingga pada *epoch* ke 200 yang memiliki nilai sebesar 1,07. Hal ini mengindikasikan bahwa model mungkin mengalami kesulitan dalam menerapkan pembelajaran pada data *testing* yang belum pernah dilihat sebelumnya. Dalam grafik dapat dilihat bahwa nilai *loss* pada data *training* terus mengalami penurunan, namun nilai *loss* pada data *testing* mulai naik turun. Hal ini dapat menunjukkan bahwa terjadi *overfitting*, dimana model terlalu baik dalam menyesuaikan diri dengan data *training* namun kesulitan dalam menggeneralisasi pada data baru. Terdapat beberapa penyebab hasil *loss* menjadi kurang baik. Penyebab pertama karena adanya *overfitting*. *Overfitting* dapat terjadi jika model terlalu kompleks dan mampu mempelajari data *training* dengan baik, namun tidak dapat melakukan generalisasi dengan baik pada data *testing*. Penyebab kedua karena pemilihan model dan parameter yang salah. Hal ini menyebabkan kinerja yang buruk pada data *testing*. Untuk itu diperlukan pemilihan yang baik dalam model dan juga parameter-parameter yang digunakan agar menghasilkan hasil yang baik. Pada grafik *loss*, terlihat bahwa garis yang dihasilkan lebih halus dan tidak seperti grafik pada akurasi. Hal ini dikarenakan nilai *loss* diambil dari probabilitas dengan rentang nol sampai satu serta terdapat eror yang menyebabkan garis pada grafik *loss* lebih halus dan kontinyu.



Gambar 4.14 Grafik Hasil *Loss* Model 2

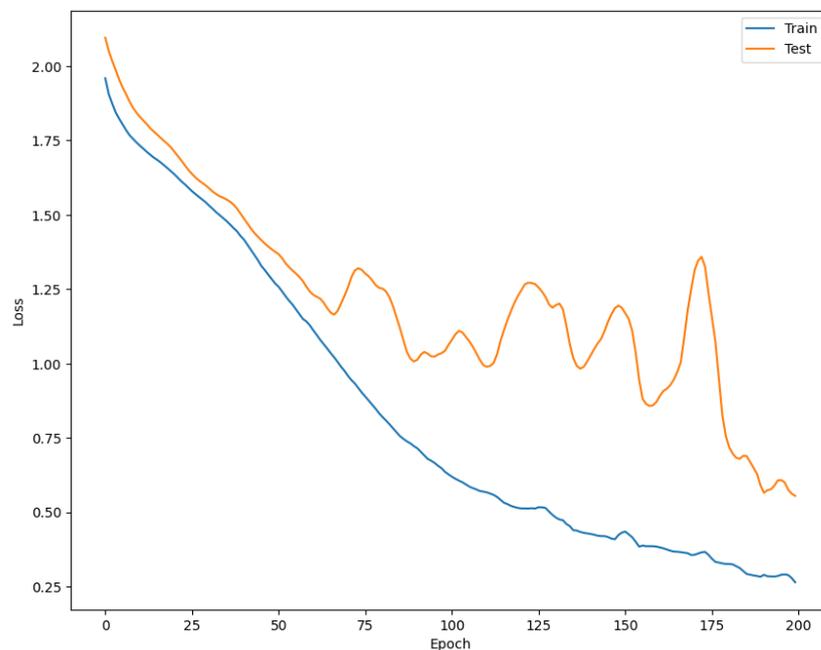
Berdasarkan informasi pada Gambar 4.14 yang merupakan grafik hasil *loss* model 2, sumbu X menunjukkan jumlah *epoch* yang digunakan dalam model DNN yaitu 200 kali. Sedangkan sumbu Y menunjukkan nilai *loss* yang didapatkan dalam model DNN untuk setiap *epoch*. Pada awal dimulainya iterasi, nilai *loss* untuk *training* dan *testing* bernilai tinggi karena model belum menyesuaikan dengan baik terhadap data. Pada data *training* yang ditandai dengan garis berwarna biru, nilai *loss* untuk *epoch* 0 sebesar 1,81 sedangkan pada data *testing* yang ditandai dengan garis berwarna oranye, nilai *loss*nya sebesar 1,77. Dalam grafik tersebut, data *training* mengalami penurunan secara terus-menerus untuk nilai *loss* sampai dengan 0,41 pada *epoch* ke 200. Hal tersebut menunjukkan bahwa model secara efektif mengenali pola-pola dalam data *training* dan berhasil mengurangi kesalahan prediksi seiring berjalannya iterasi. Untuk data *testing*, nilai *loss* mengalami penurunan hingga pada *epoch* ke 200 yang memiliki nilai sebesar 0,62. Hal tersebut menunjukkan bahwa model secara efektif mengenali pola-pola dalam data *testing* dan berhasil mengurangi kesalahan prediksi seiring berjalannya iterasi. Dalam grafik dapat dilihat bahwa nilai *loss* pada data *training* dan nilai *loss* pada data *testing* terus mengalami penurunan. Hal ini dapat menunjukkan bahwa model DNN mampu membuat prediksi yang akurat. Terdapat beberapa penyebab hasil *loss* dapat menghasilkan hasil yang baik. Penyebab pertama karena model DNN telah berhasil atau mampu menggeneralisasi dengan baik pada data *testing* dan tidak mengalami *overfitting*. Penyebab kedua karena pemilihan model yang sesuai. Model DNN yang dipilih sudah sesuai dengan karakteristik data yang diolah. Pemilihan struktur model, jumlah *layer*, fungsi aktivasi dan parameter lainnya yang tepat dapat membantu meningkatkan generalisasi model pada data *testing*. Pada grafik *loss*, terlihat bahwa garis yang dihasilkan lebih halus dan tidak seperti grafik pada akurasi. Hal ini dikarenakan nilai *loss* diambil dari probabilitas dengan rentang nol sampai satu serta terdapat eror yang menyebabkan garis pada grafik *loss* lebih halus dan kontinyu.



Gambar 4.15 Grafik Hasil *Loss* Model 3

Berdasarkan informasi pada Gambar 4.15 yang merupakan grafik hasil *loss* model 33, sumbu X menunjukkan jumlah *epoch* yang digunakan dalam model DNN yaitu 200 kali. Sedangkan sumbu Y menunjukkan nilai *loss* yang didapatkan dalam model DNN untuk setiap *epoch*. Pada awal dimulainya iterasi, nilai *loss* untuk *training* dan *testing* bernilai tinggi karena

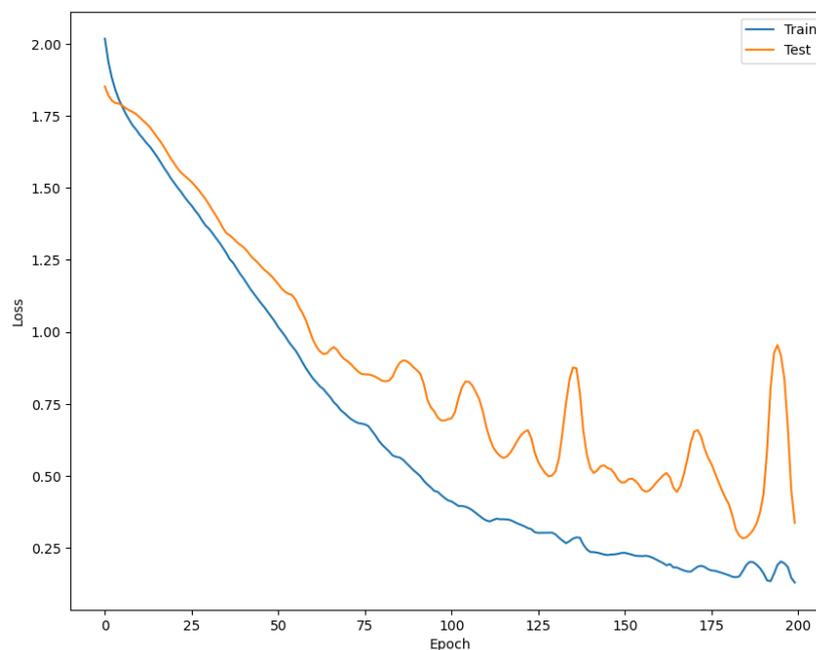
model belum menyesuaikan dengan baik terhadap data. Pada data *training* yang ditandai dengan garis berwarna biru, nilai *loss* untuk *epoch* 0 sebesar 1,81 sedangkan pada data *testing* yang ditandai dengan garis berwarna oranye, nilai *loss*nya sebesar 1,77. Dalam grafik tersebut, data *training* mengalami penurunan secara terus-menerus untuk nilai *loss* sampai dengan 0,35 pada *epoch* ke 200. Hal tersebut menunjukkan bahwa model secara efektif mengenali pola-pola dalam data *training* dan berhasil mengurangi kesalahan prediksi seiring berjalannya iterasi. Untuk data *testing*, nilai *loss* mengalami penurunan, namun pada *epoch* ke 40 hasil nilai *loss* tidak mengalami penurunan melainkan naik turun hingga pada *epoch* ke 200 yang memiliki nilai sebesar 1,07. Hal ini mengindikasikan bahwa model mungkin mengalami kesulitan dalam menerapkan pembelajaran pada data *testing* yang belum pernah dilihat sebelumnya. Dalam grafik dapat dilihat bahwa nilai *loss* pada data *training* terus mengalami penurunan, namun nilai *loss* pada data *testing* mulai naik turun. Hal ini dapat menunjukkan bahwa terjadi *overfitting*, dimana model terlalu baik dalam menyesuaikan diri dengan data *training* namun kesulitan dalam mengeneralisasi pada data baru. Terdapat beberapa penyebab hasil *loss* menjadi kurang baik. Penyebab pertama karena adanya *overfitting*. *Overfitting* dapat terjadi jika model terlalu kompleks dan mampu mempelajari data *training* dengan baik, namun tidak dapat melakukan generalisasi dengan baik pada data *testing*. Penyebab kedua karena pemilihan model dan parameter yang salah. Hal ini menyebabkan kinerja yang buruk pada data *testing*. Untuk itu diperlukan pemilihan yang baik dalam model dan juga parameter-parameter yang digunakan agar menghasilkan hasil yang baik. Pada grafik *loss*, terlihat bahwa garis yang dihasilkan lebih halus dan tidak seperti grafik pada akurasi. Hal ini dikarenakan nilai *loss* diambil dari probabilitas dengan rentang nol sampai satu serta terdapat eror yang menyebabkan garis pada grafik *loss* lebih halus dan kontinyu.



Gambar 4.16 Grafik Hasil *Loss* Model 4

Berdasarkan informasi pada Gambar 4.16 yang merupakan grafik hasil *loss* model 4, sumbu X menunjukkan jumlah *epoch* yang digunakan dalam model DNN yaitu 200 kali. Sedangkan sumbu Y menunjukkan nilai *loss* yang didapatkan dalam model DNN untuk setiap *epoch*. Pada awal dimulainya iterasi, nilai *loss* untuk *training* dan *testing* bernilai tinggi karena model belum menyesuaikan dengan baik terhadap data. Pada data *training* yang ditandai

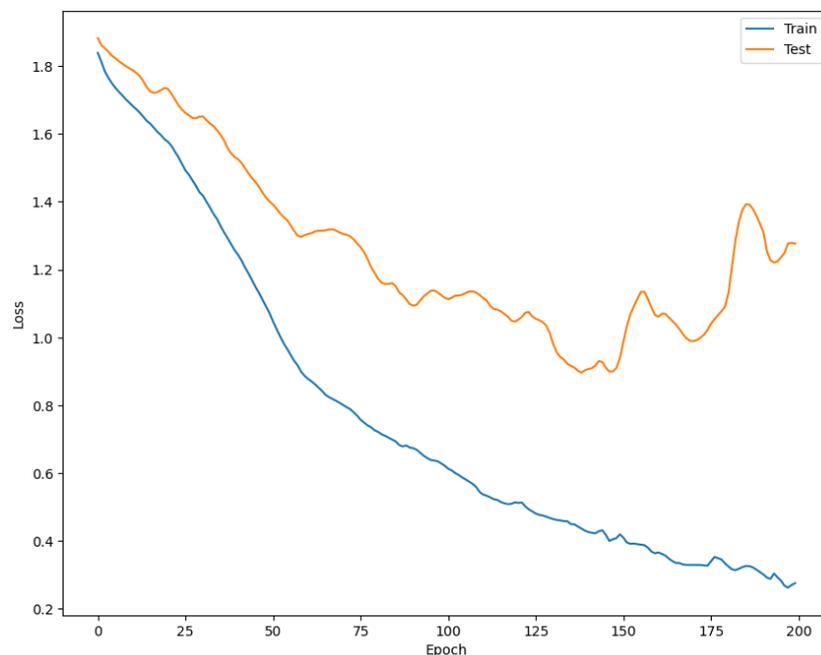
dengan garis berwarna biru, nilai *loss* untuk *epoch* 0 sebesar 1,94 sedangkan pada data *testing* yang ditandai dengan garis berwarna oranye, nilai *loss*nya sebesar 2,09. Dalam grafik tersebut, data *training* mengalami penurunan secara terus-menerus untuk nilai *loss* sampai dengan 0,26 pada *epoch* ke 200. Hal tersebut menunjukkan bahwa model secara efektif mengenali pola-pola dalam data *training* dan berhasil mengurangi kesalahan prediksi seiring berjalannya iterasi. Untuk data *testing*, nilai *loss* mengalami penurunan, namun pada *epoch* ke 70 hasil nilai *loss* tidak mengalami penurunan melainkan naik turun hingga pada *epoch* ke 200 yang memiliki nilai sebesar 0,55. Hal ini mengindikasikan bahwa model mungkin mengalami kesulitan dalam menerapkan pembelajaran pada data *testing* yang belum pernah dilihat sebelumnya. Dalam grafik dapat dilihat bahwa nilai *loss* pada data *training* terus mengalami penurunan, namun nilai *loss* pada data *testing* mulai naik turun. Hal ini dapat menunjukkan bahwa terjadi *overfitting*, dimana model terlalu baik dalam menyesuaikan diri dengan data *training* namun kesulitan dalam menggeneralisasi pada data baru. Terdapat beberapa penyebab hasil *loss* menjadi kurang baik. Penyebab pertama karena adanya *overfitting*. *Overfitting* dapat terjadi jika model terlalu kompleks dan mampu mempelajari data *training* dengan baik, namun tidak dapat melakukan generalisasi dengan baik pada data *testing*. Penyebab kedua karena pemilihan model dan parameter yang salah. Hal ini menyebabkan kinerja yang buruk pada data *testing*. Untuk itu diperlukan pemilihan yang baik dalam model dan juga parameter-parameter yang digunakan agar menghasikan hasil yang baik. Pada grafik *loss*, terlihat bahwa garis yang dihasilkan lebih halus dan tidak seperti grafik pada akurasi. Hal ini dikarenakan nilai *loss* diambil dari probabilitas dengan rentang nol sampai satu serta terdapat eror yang menyebabkan garis pada grafik *loss* lebih halus dan kontinyu.



Gambar 4.17 Grafik Hasil *Loss* Model 5

Berdasarkan informasi pada Gambar 4.17 yang merupakan grafik hasil *loss* model 5, sumbu X menunjukkan jumlah *epoch* yang digunakan dalam model DNN yaitu 200 kali. Sedangkan sumbu Y menunjukkan nilai *loss* yang didapatkan dalam model DNN untuk setiap *epoch*. Pada awal dimulainya iterasi, nilai *loss* untuk *training* dan *testing* bernilai tinggi karena model belum menyesuaikan dengan baik terhadap data. Pada data *training* yang ditandai dengan garis berwarna biru, nilai *loss* untuk *epoch* 0 sebesar 2,09 sedangkan pada data *testing*

yang ditandai dengan garis berwarna oranye, nilai *loss*nya sebesar 1,85. Dalam grafik tersebut, data *training* mengalami penurunan secara terus-menerus untuk nilai *loss* sampai dengan 0,13 pada *epoch* ke 200. Hal tersebut menunjukkan bahwa model secara efektif mengenali pola-pola dalam data *training* dan berhasil mengurangi kesalahan prediksi seiring berjalannya iterasi. Untuk data *testing*, nilai *loss* mengalami penurunan, namun pada *epoch* ke 60 hasil nilai *loss* tidak mengalami penurunan melainkan naik turun hingga pada *epoch* ke 200 yang memiliki nilai sebesar 0,33. Hal ini mengindikasikan bahwa model mungkin mengalami kesulitan dalam menerapkan pembelajaran pada data *testing* yang belum pernah dilihat sebelumnya. Namun, pada model 5 ini, walaupun nilai *loss* untuk *testing* naik turun, dapat dilihat dalam gambar bahwa garis untuk *training* dan *testing* hampir berdekatan. Hal ini berarti pada data *testing*, model juga melakukan prediksi dengan cukup baik. Meskipun demikian, dalam grafik dapat dilihat juga bahwa nilai *loss* pada data *training* terus mengalami penurunan, namun nilai *loss* pada data *testing* mulai naik turun walaupun hanya sedikit. Hasil nilai *loss* model 5 dapat dikatakan baik karena mendekati nol. Terdapat beberapa penyebab hasil *loss* dapat menghasilkan hasil yang baik. Penyebab pertama karena model DNN telah berhasil atau mampu menggeneralisasi dengan baik pada data *testing* dan tidak mengalami *overfitting*. Penyebab kedua karena pemilihan model yang sesuai. Model DNN yang dipilih sudah sesuai dengan karakteristik data yang diolah. Pemilihan struktur model, jumlah *layer*, fungsi aktivasi dan parameter lainnya yang tepat dapat membantu meningkatkan generalisasi model pada data *testing*. Pada grafik *loss*, terlihat bahwa garis yang dihasilkan lebih halus dan tidak seperti grafik pada akurasi. Hal ini dikarenakan nilai *loss* diambil dari probabilitas dengan rentang nol sampai satu serta terdapat eror yang menyebabkan garis pada grafik *loss* lebih halus dan kontinyu.



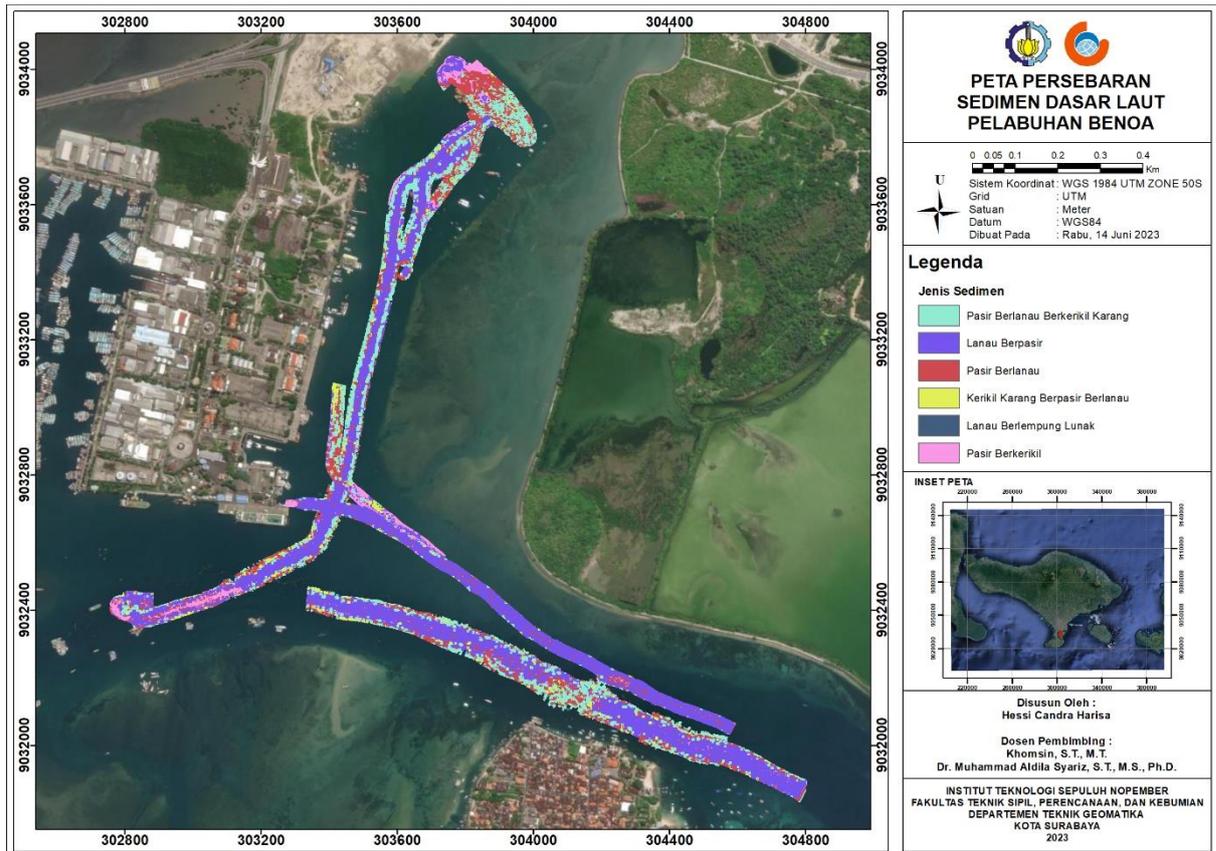
Gambar 4.18 Grafik Hasil *Loss* Model 6

Berdasarkan informasi pada Gambar 4.18 yang merupakan grafik hasil *loss* model 6, sumbu X menunjukkan jumlah *epoch* yang digunakan dalam model DNN yaitu 200 kali. Sedangkan sumbu Y menunjukkan nilai *loss* yang didapatkan dalam model DNN untuk setiap *epoch*. Pada awal dimulainya iterasi, nilai *loss* untuk *training* dan *testing* bernilai tinggi karena model belum menyesuaikan dengan baik terhadap data. Pada data *training* yang ditandai dengan garis berwarna biru, nilai *loss* untuk *epoch* 0 sebesar 1,83 sedangkan pada data *testing*

yang ditandai dengan garis berwarna oranye, nilai *loss*-nya sebesar 1,88. Dalam grafik tersebut, data *training* mengalami penurunan secara terus-menerus untuk nilai *loss* sampai dengan 0,27 pada *epoch* ke 200. Hal tersebut menunjukkan bahwa model secara efektif mengenali pola-pola dalam data *training* dan berhasil mengurangi kesalahan prediksi seiring berjalannya iterasi. Untuk data *testing*, nilai *loss* mengalami penurunan, namun pada *epoch* ke 150 hasil nilai *loss* tidak mengalami penurunan melainkan naik turun hingga pada *epoch* ke 200 yang memiliki nilai sebesar 1,27. Hal ini mengindikasikan bahwa model mungkin mengalami kesulitan dalam menerapkan pembelajaran pada data *testing* yang belum pernah dilihat sebelumnya. Dalam grafik dapat dilihat bahwa nilai *loss* pada data *training* terus mengalami penurunan, namun nilai *loss* pada data *testing* mulai naik turun. Hal ini dapat menunjukkan bahwa terjadi *overfitting*, dimana model terlalu baik dalam menyesuaikan diri dengan data *training* namun kesulitan dalam menggeneralisasi pada data baru. Terdapat beberapa penyebab hasil *loss* menjadi kurang baik. Penyebab pertama karena adanya *overfitting*. *Overfitting* dapat terjadi jika model terlalu kompleks dan mampu mempelajari data *training* dengan baik, namun tidak dapat melakukan generalisasi dengan baik pada data *testing*. Penyebab kedua karena pemilihan model dan parameter yang salah. Hal ini menyebabkan kinerja yang buruk pada data *testing*. Untuk itu diperlukan pemilihan yang baik dalam model dan juga parameter-parameter yang digunakan agar menghasilkan hasil yang baik. Pada grafik *loss*, terlihat bahwa garis yang dihasilkan lebih halus dan tidak seperti grafik pada akurasi. Hal ini dikarenakan nilai *loss* diambil dari probabilitas dengan rentang nol sampai satu serta terdapat eror yang menyebabkan garis pada grafik *loss* lebih halus dan kontinyu.

#### 4.3 Klasifikasi dan Persebaran Sedimen Dasar Laut

Partikel-partikel sedimen mempunyai ukuran yang besar sampai yang sangat halus dan mempunyai bentuk mulai dari bulat, lonjong, sampai persegi. Sesuai dengan data *insitu* yang berupa sampel sedimen sebanyak 12 titik, di sekitar Pelabuhan Benoa terindikasi terdapat 6 jenis sedimen. Jenis sedimen tersebut adalah pasir berlanau berkerikil karang, lanau berpasir, pasir berlanau, kerikil karang berpasir berlanau, lanau kelepungan lunak, dan pasir berkerikil. Klasifikasi sedimen dasar laut yang merupakan hasil dari pengolahan model DNN mengacu pada titik-titik sampel sedimen. Setelah ditentukan model DNN dengan nilai akurasi yang paling baik, dilakukan proses klasifikasi menggunakan model DNN tersebut dan juga menggunakan data nilai ekstraksi raster mosaik *backscatter* dan raster fitur mosaik *backscatter*. Dari hasil pengolahan klasifikasi didapatkan nilai dengan rentang 1 sampai 6 pada tiap titik-titik ekstraksi nilai raster yang sebelumnya telah dilakukan. Rentang nilai 1 sampai 6 tersebut sesuai dengan jumlah jenis sedimen yang diketahui. Sehingga dari hasil klasifikasi tersebut, dapat dibuat sebuah peta persebaran sedimen di Pelabuhan Benoa. Peta ini akan dapat membantu dalam berbagai hal di sekitar Pelabuhan Benoa seperti untuk perencanaan dan desain infrastruktur pelabuhan, untuk perencanaan pengerukan dan pemeliharaan dasar laut di pelabuhan, untuk navigasi dan keamanan pelayaran, untuk memberikan informasi tentang potensi polusi dan kualitas air pelabuhan, serta untuk memahami perubahan arus dan pasang surut di wilayah pelabuhan. Berikut merupakan peta sebaran sedimen yang ditunjukkan oleh Gambar 4.19.

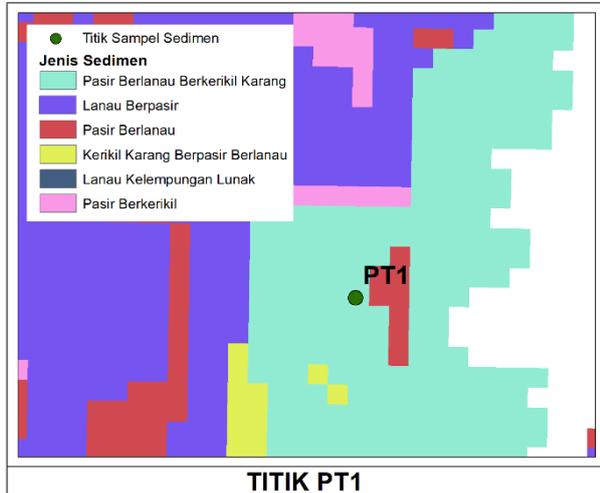


Gambar 4.19 Peta Sebaran Sedimen Dasar Laut Pelabuhan Bena

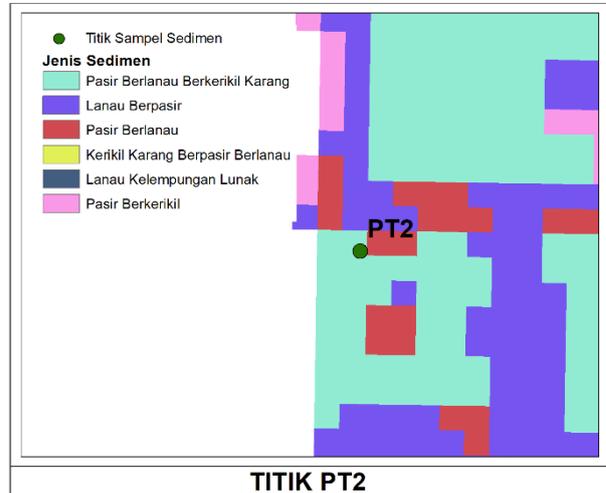
Dalam Gambar 4.19, dapat dilihat jika jenis sedimen lanau berpasir memiliki sebaran paling banyak. Jika berdasarkan jumlah titik jenis sedimen yang merupakan hasil pengolahan dari klasifikasi, terdapat 388.799 titik jenis sedimen dimana 191.659 titik dengan jenis sedimen lanau berpasir, 86.784 titik dengan jenis sedimen pasir berlanau berkerikil karang, 68.002 titik dengan jenis sedimen pasir berlanau, 22.585 titik dengan jenis sedimen pasir berkerikil, 17.725 titik dengan jenis sedimen kerikil karang berpasir berlanau, dan 2.044 titik dengan jenis sedimen lanau berlempung lunak. Jika dilihat dalam persentase, jenis sedimen lanau berpasir memiliki presentase paling besar yaitu 49,30%, lalu ada pasir berlanau berkerikil karang sebesar 22,32%, pasir berlanau sebesar 17,49%, pasir berkerikil sebesar 5,80%, kerikil karang berpasir berlanau sebesar 4,56%, dan jenis sedimen lanau berlempung lunak dengan persentase paling sedikit sebesar 0,53%.

Persebaran jenis sedimen dasar laut pada Gambar 4.19 memiliki pola dimana jenis sedimen lanau berpasir menyebar di tengah lajur dan jenis sedimen lain menyebar di pinggir lajur. Hal ini dapat dipengaruhi oleh berbagai hal seperti kurangnya variabilitas data, kurangnya data, dan kompleksitas model. Kondisi sedimen dalam lajur pelayaran memiliki variasi yang kompleks, sehingga jika variasi ini tidak diperhitungkan dengan baik dalam model, hasilnya dapat menghasilkan kesalahan dalam mengklasifikasi sedimen. Kurangnya data sampel sedimen juga dapat mempengaruhi hasil klasifikasi, dimana jika data sampel yang digunakan lebih banyak, maka kemungkinan akan dihasilkan klasifikasi yang lebih akurat. Kompleksitas model juga berpengaruh terhadap hasil klasifikasi karena jika model tidak dapat mempelajari pola dengan baik, maka hasil klasifikasinya bisa terdapat kesalahan. Hasil dari peta persebaran jenis sedimen menunjukkan hasil yang kurang baik dan kurang akurat. Hal ini mungkin disebabkan oleh beberapa hal seperti data awal yang kurang baik maupun data atau parameter

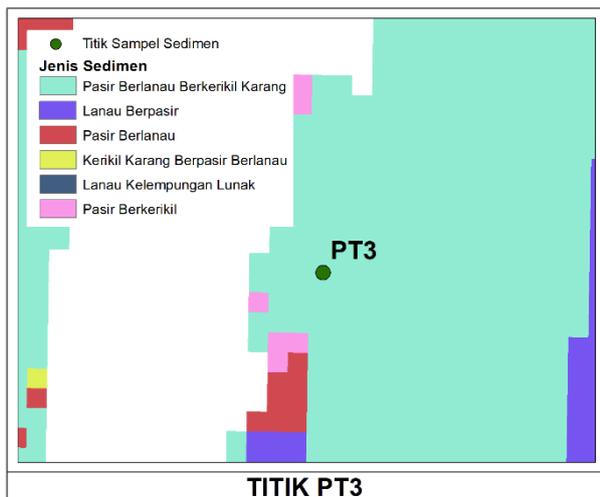
yang digunakan memang kurang untuk mengidentifikasi jenis sedimennya. Oleh karena itu, untuk menghasilkan jenis sedimen yang akurat, dapat ditambahkan parameter lain yang digunakan sehingga model DNN dapat melakukan pembelajaran dengan baik. Berikut merupakan hasil dari pengolahan DNN untuk setiap titik sampel sedimen.



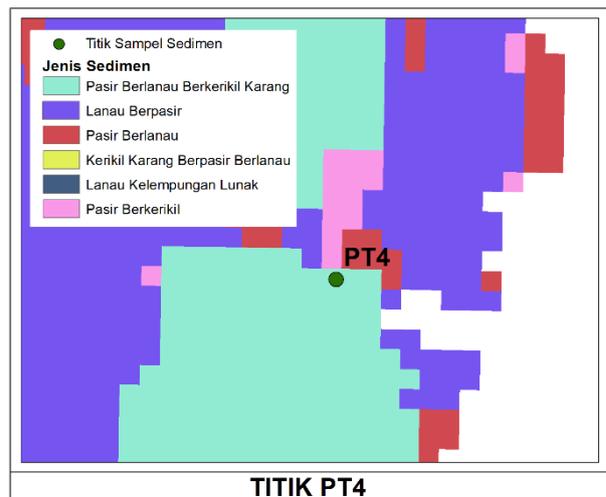
Gambar 4.20 Titik Sampel Sedimen PT1



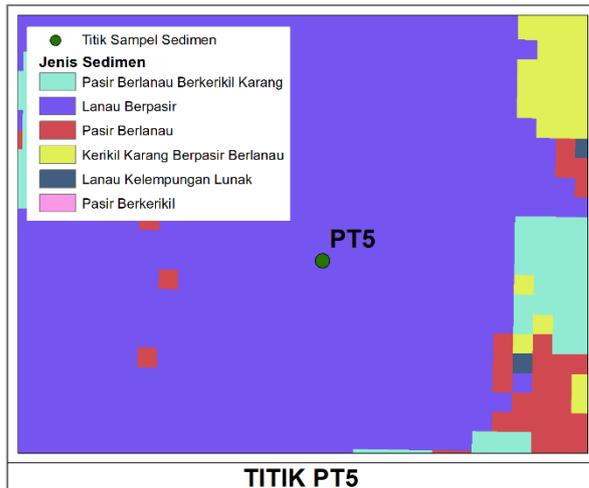
Gambar 4.21 Titik Sampel Sedimen PT2



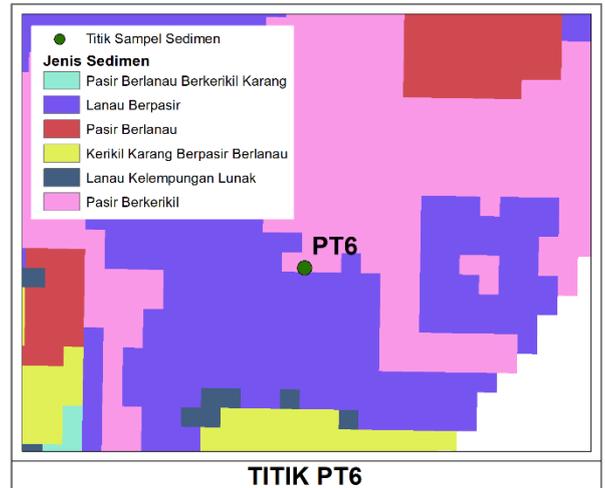
Gambar 4.22 Titik Sampel Sedimen PT3



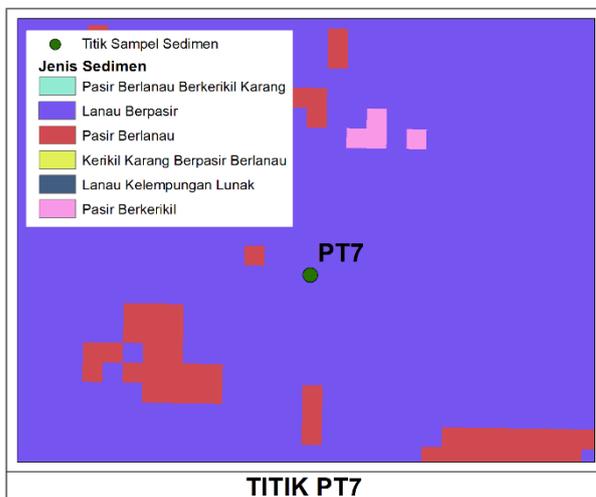
Gambar 4.23 Titik Sampel Sedimen PT4



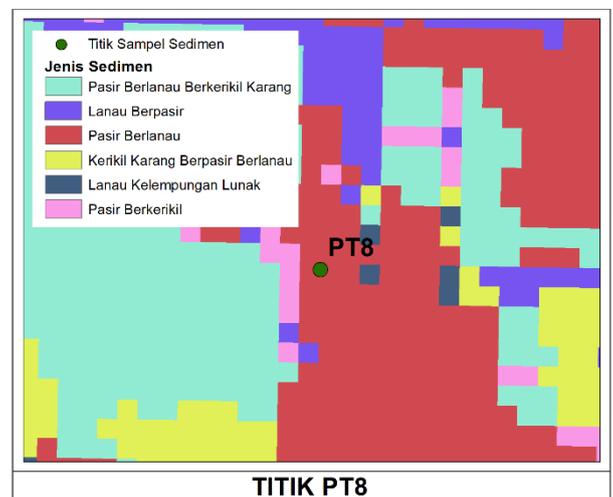
Gambar 4.24 Titik Sampel Sedimen PT5



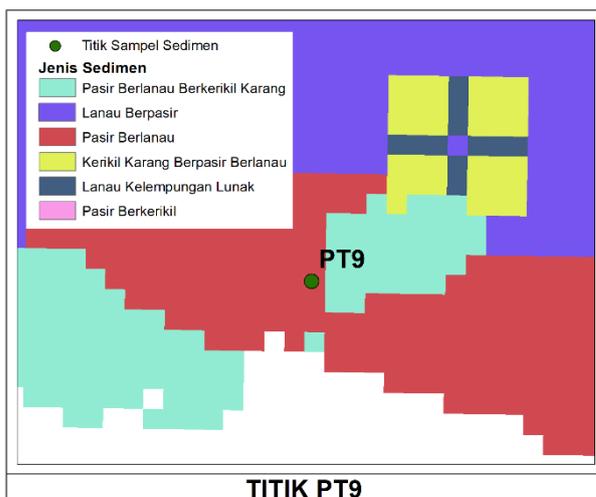
Gambar 4.25 Titik Sampel Sedimen PT6



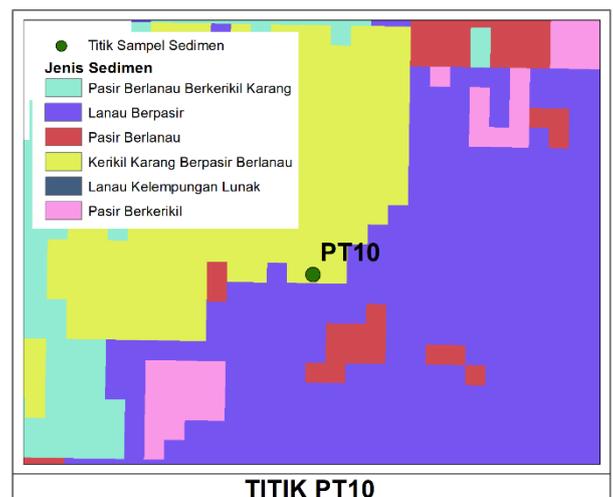
Gambar 4.26 Titik Sampel Sedimen PT7



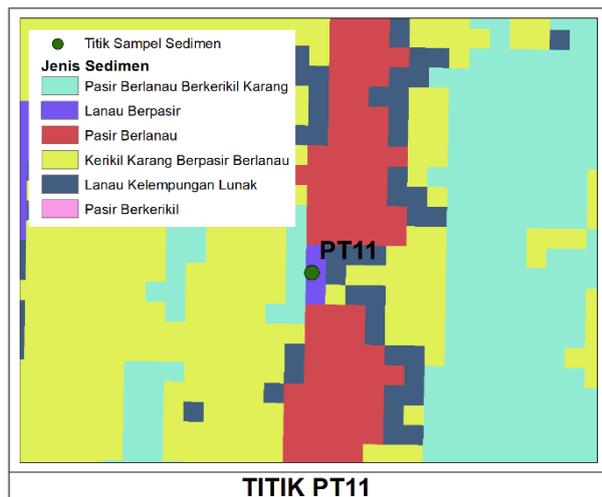
Gambar 4.27 Titik Sampel Sedimen PT8



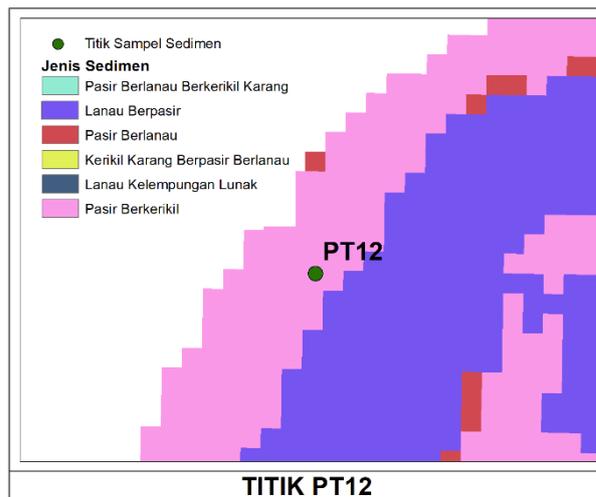
Gambar 4.28 Titik Sampel Sedimen PT9



Gambar 4.29 Titik Sampel Sedimen PT10



Gambar 4.30 Titik Sampel Sedimen PT11



Gambar 4.31 Titik Sampel Sedimen PT12

Dalam gambar-gambar di atas dapat dilihat persebaran titik sampel pada hasil pengolahan klasifikasi jenis sedimen dengan menggunakan DNN. Pada Gambar 4.20, titik PT1 berada pada jenis sedimen dengan warna *cyan*. Jika dilihat berdasarkan legenda, titik PT1 memiliki jenis sedimen pasir berlanau berkerikil karang. Pada Gambar 4.21, titik PT2 berada pada jenis sedimen dengan warna *cyan*. Jika dilihat berdasarkan legenda, titik PT2 memiliki jenis sedimen pasir berlanau berkerikil karang. Pada Gambar 4.22, titik PT3 berada pada jenis sedimen dengan warna *cyan*. Jika dilihat berdasarkan legenda, titik PT3 memiliki jenis sedimen pasir berlanau berkerikil karang. Pada Gambar 4.23, titik PT4 berada pada jenis sedimen dengan warna *cyan*. Jika dilihat berdasarkan legenda, titik PT4 memiliki jenis sedimen pasir berlanau berkerikil karang. Pada Gambar 4.24, titik PT5 berada pada jenis sedimen dengan warna ungu. Jika dilihat berdasarkan legenda, titik PT5 memiliki jenis sedimen lanau berpasir. Pada Gambar 4.25, titik PT6 berada pada jenis sedimen dengan warna merah muda. Jika dilihat berdasarkan legenda, titik PT6 memiliki jenis sedimen pasir berkerikil. Pada Gambar 4.26, titik PT7 berada pada jenis sedimen dengan warna ungu. Jika dilihat berdasarkan legenda, titik PT7 memiliki jenis sedimen lanau berpasir. Pada Gambar 4.27, titik PT8 berada pada jenis sedimen dengan warna merah. Jika dilihat berdasarkan legenda, titik PT8 memiliki jenis sedimen pasir berlanau. Pada Gambar 4.28, titik PT9 berada pada jenis sedimen dengan warna merah. Jika dilihat berdasarkan legenda, titik PT9 memiliki jenis sedimen pasir berlanau. Pada Gambar 4.29, titik PT10 berada pada jenis sedimen dengan warna kuning. Jika dilihat berdasarkan legenda, titik PT10 memiliki jenis sedimen kerikil karang berpasir berlanau. Pada Gambar 4.30, titik PT11 berada pada jenis sedimen dengan warna ungu. Jika dilihat berdasarkan legenda, titik PT11 memiliki jenis sedimen lanau berpasir. Pada Gambar 4.31, titik PT12 berada pada jenis sedimen dengan warna merah muda. Jika dilihat berdasarkan legenda, titik PT12 memiliki jenis sedimen pasir berkerikil.

Dalam memastikan jenis sedimen yang benar, dapat dilakukan validasi dengan membandingkan jenis sedimen dari sampel sedimen dan jenis sedimen dari hasil pengolahan. Dari validasi ini dapat dilihat titik mana yang sama dengan jenis sedimen di sampel sedimen dan titik mana yang tidak sama. Dari Gambar 4.20 sampai Gambar 4.31, dapat dijabarkan dalam bentuk tabel mengenai perbandingan jenis sedimen dari sampel sedimen dengan jenis sedimen dari pengolahan menggunakan metode *Dense Neural Network*.

Tabel 4.3 Perbandingan Jenis Sedimen

No	Nama Titik	Jenis Sedimen	
		Sampel Sedimen	Pengolahan DNN
1.	PT1	Pasir berlanau berkerikil karang	Pasir berlanau berkerikil karang
2.	PT2	Pasir berlanau berkerikil karang	Pasir berlanau berkerikil karang
3.	PT3	Pasir berlanau berkerikil karang	Pasir berlanau berkerikil karang
4.	PT4	Pasir berlanau berkerikil karang	Pasir berlanau berkerikil karang
5.	PT5	Lanau berpasir	Lanau berpasir
6.	PT6	Lanau berpasir	Pasir berkerikil
7.	PT7	Pasir berlanau	Lanau berpasir
8.	PT8	Pasir berlanau	Pasir berlanau
9.	PT9	Pasir berlanau	Pasir berlanau
10.	PT10	Kerikil karang berpasir berlanau	Kerikil karang berpasir berlanau
11.	PT11	Lanau kelepungan lunak	Lanau berpasir
12.	PT12	Pasir berkerikil	Pasir berkerikil

Perbandingan antara jenis sedimen dari sampel sedimen dan jenis sedimen dari pengolahan DNN menghasilkan hasil yang sama kecuali di 3 titik sedimen yaitu titik PT6, PT7, dan PT11. Perbedaan jenis sedimen ini mungkin terjadi dikarenakan beberapa alasan. Alasan pertama yang memungkinkan adanya perbedaan jenis sedimen adalah keterbatasan data *training*. Jika data *training* yang digunakan terbatas, tidak seimbang ataupun tidak mewakili jenis sedimen yang cukup maka akan mempengaruhi kemampuan model untuk mempelajari pola dengan baik. Alasan kedua yang memungkinkan perbedaan jenis sedimen adalah kesalahan dalam susunan model atau parameter model. Kesalahan ini memberikan dampak pada hasil prediksi, dimana jika susunan atau struktur model DNN tidak sesuai atau parameter *training* tidak dioptimalkan dengan baik, maka model mungkin tidak dapat menghasilkan prediksi yang akurat. Untuk itu, pada saat pengolahan model DNN harus dipastikan jika model benar-benar tepat, sehingga akan menghasilkan klasifikasi sedimen yang akurat.

*“Halaman ini sengaja dikosongkan”*

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan hasil dan analisis penelitian yang telah dilakukan dapat ditarik kesimpulan sebagai berikut:

1. Penelitian ini menghasilkan 6 model *Dense Neural Network* (DNN). Dari keseluruhan model yang dihasilkan, desain model DNN paling baik adalah model DNN 5 karena memiliki nilai akurasi untuk *training* dan akurasi *testing* paling tinggi
2. Uji akurasi pada model DNN paling tinggi (Model DNN 5) menunjukkan nilai akurasi *training* sebesar 88% dan akurasi *testing* sebesar 100%.
3. Dari hasil klasifikasi jenis sedimen terdapat 6 jenis sedimen yaitu pasir berlanau berkerikil karang, lanau berpasir, pasir berlanau, kerikil karang berpasir berlanau, lanau kelepungan lunak, dan pasir berkerikil. Jenis sedimen dari sampel sedimen dan dari pengolahan DNN menghasilkan hasil yang sama kecuali di 3 titik sedimen yaitu titik PT6, PT7, dan PT11. Dalam persentase, jenis sedimen lanau berpasir memiliki presentase paling besar yaitu 49,30%, lalu ada pasir berlanau berkerikil karang sebesar 22,32%, pasir berlanau sebesar 17,49%, pasir berkerikil sebesar 5,80%, kerikil karang berpasir berlanau sebesar 4,56%, dan jenis sedimen lanau berlempung lunak dengan persentase paling sedikit sebesar 0,53%.

#### **5.2 Saran**

Adapun saran yang diberikan penulis berdasarkan penelitian yang telah dilakukan adalah sebagai berikut :

1. Memperbanyak data sampel sedimen yang akan dilakukan klasifikasi agar hasil dari klasifikasi dapat sesuai dengan yang diinginkan.
2. Pada saat pengolahan model DNN harus dipastikan jika model benar-benar tepat, sehingga akan menghasilkan klasifikasi sedimen yang akurat.

*“Halaman ini sengaja dikosongkan”*

## DAFTAR PUSTAKA

- Ackerman, S. D., Pappal, A. L., Huntley, E. C., Blackwood, D. S., & Schwab, W. C. (2015). Geological Sampling Data and Benthic Biota Classification: Buzzards Bay and Vineyard Sound, Massachusetts O. *U.S. Geological Survey*. <https://doi.org/10.3133/ofr20141221>
- Adi, A. P., Manik, H. M., & Pujiyati, S. (2016). Integrasi Data Multibeam Batimetri dan Mosaik Backscatter untuk Klasifikasi Tipe Sedimen. Dalam *Jurnal Teknologi Perikanan dan Kelautan* (Vol. 7, Nomor 1). <https://doi.org/https://doi.org/10.24319/jtpk.7.77-84>
- Ahmad, A. (2017). *Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning*.
- Akbar, K., Pratomo, D. G., & Khomsin. (2017). Analisis Nilai Hambur Balik Sedimen Permukaan Dasar Perairan Menggunakan Data Multibeam Echosounder EM302. *Jurnal Teknik ITS*, 6(2).
- Akbar, Z. A., Pratomo, D. G., Negara, A. K., & Riyadi, N. (2022). Identifikasi Objek Berdimensi Kecil Menggunakan Sapuan Multibeam Echosounder. *Jurnal Chart Datum*, 4(2), 127–136. <https://doi.org/https://doi.org/10.37875/chartdatum.v4i2.134>
- Amherd, F., & Rodriguez, E. (2021). *Heatmap-based Object Detection and Tracking with a Fully Convolutional Neural Network*. <http://arxiv.org/abs/2101.03541>
- Aquatic BioTechnology. (2023). *Van Veen Grab*. <https://aquaticbiotechnology.com/en/sediment-sampling/van-veen-grab>
- Bayhaqi, A., & Dungga, C. M. A. (2015). Distribusi Butiran Sedimen di Pantai Dalegan, Gresik, Jawa Timur. *DEPIK*, 4(3). <https://doi.org/10.13170/depik.4.3.3054>
- Bekti. (2017, Desember 4). *Review SNAP*. SISPI (Sistem Informasi Spasial dan Pemetaan Hutan). <https://bentangalam-hutantropis.fkt.ugm.ac.id/2017/12/04/review-snap/>
- Bellec, V. K., Bøe, R., Rise, L., Lepland, A., Thorsnes, T., & Bjarnadóttir, L. R. (2017). Seabed Sediments (Grain Size) of Nordland VI, Offshore North Norway. *Journal of Maps*, 13(2), 608–620. <https://doi.org/10.1080/17445647.2017.1348307>
- Brammadi, S., Laila Nugraha, A., Sudarsono, B., & Mudita, I. (2017). Analisis Pengolahan Data Multibeam Echosounder Menggunakan Perangkat Lunak MB-System dan Caris Hips and Sips Berdasarkan Standar S-44 IHO 2008. Dalam *Jurnal Geodesi Undip Oktober* (Vol. 6, Nomor 4).
- Brennan, C.W. (2009). *Multibeam Calibration: The Patch Test*. R2Sonic LLC. USA.
- Cahyono, B. K., & Atunggal, D. (2012). *Analisis Persebaran Sedimentasi Waduk Berdasarkan Data Multi Epoch Hasil Pengukuran Bathymetry yang Terikat pada Stasiun GNSS-CORS*.
- Carmacoring. (2023). *Gravity Corer*. <https://www.carmacoring.com/products-and-solutions/gravity-corer/>
- Chen, X., Hassan, M. A., & Fu, X. (2022). Convolutional neural networks for image-based sediment detection applied to a large terrestrial and airborne dataset. *Earth Surface Dynamics*, 10(2), 349–366. <https://doi.org/10.5194/esurf-10-349-2022>
- Cui, X., Yang, F., Wang, X., Ai, B., Luo, Y., & Ma, D. (2021). Deep learning model for seabed sediment classification based on fuzzy ranking feature optimization. *Marine Geology*, 432.
- Chouinard, J.-C. (2022, Oktober 18). *How to Use Google Colab for Python (With Examples)*. <https://www.jcchouinard.com/google-colab-with-python/>
- Clarke, Hughes JE, Danforth BW, P.Valentine. (1997). Areal seabed classification using backscatter angular response at 95 kHz. NATO SACLANTCEN Conference Proceedings Series CP-45, High Frequency Acoustics in Shallow Water: pp. 243–250.

- Clinton, R. M. R., & Sengkey, R. (2019). Purwarupa Sistem Daftar Pelanggaran Lalulintas Berbasis Mini -Komputer Raspberry Pi. *Jurnal Teknik Elektro dan Komputer* , 8(3).
- de Jong, C. D. (2002). *Hydrography 2nd Edition*. DUP Blue Print.
- Colbo, K., Ross, T., Brown, C., & Weber, T. (2014). A review of oceanographic applications of water column data from multibeam echosounders. *Estuarine, Coastal and Shelf Science*, 145, 41–56. <https://doi.org/10.1016/j.ecss.2014.04.002>
- Djaja, R. (1989). Cara Perhitungan Pasang Surut Laut dengan Metode Admiralti. Dalam O. S. Ongkosongo , & Suyarso, Pasang Surut. Lembaga Ilmu Pengetahuan Indonesia, Pusat Penelitian dan Pengembangan Oseanologi: Jakarta.
- Ekojono, Rahutomo, F., & Sari, D. N. (2020). Implementasi Library Deep Learning Keras pada Sistem Ujian Essay Online. *JIP (Jurnal Informatika Polinema)*, 6(2).
- Faridatunnisa, M., Ulinnuha, H., Aji, J. S., Azam, K., & Basith, A. (2018). *Evaluasi Metode Pengukuran Batimetri dan UAV untuk Mendukung Pembangunan Infrastruktur Pelabuhan Tanjung Adikarto*.
- Fariyah, R. A., Manik, H. M., & Harsono, G. 2020. Pengukuran Dan Analisis Hambur Balik Akustik Menggunakan Teknologi Multibeam Echosounder untuk Klasifikasi Sedimen Dasar Laut Teluk Palu. *Jurnal Ilmu Dan Teknologi Kelautan Tropis*, 12(2), 439–455. <https://doi.org/10.29244/jitkt.v12i2.28465>
- Febriawan, H. K., Haryanto, D., Moefti, O., & Irfan, M. (2019). *Potensi Penggunaan Sistem Multibeam Sonar Laut Dalam untuk Pemetaan Batimetri dan Backscatter*. <https://www.researchgate.net/publication/342698795>
- Fitriana, D., Patria, M. P., & Kusratmoko, E. (2022). Karakteristik Pasang Surut Surabaya Diamati Selama 5 Tahun (2015-2020). *JGISE: Journal of Geospatial Information Science and Engineering*, 5(1), 1. <https://doi.org/10.22146/jgise.72856>
- Folk, R. L., & Ward, W. C. (1957). Brazos River Bar: A Study in the Significance of Grain Size Parameters. *Journal of Sedimentary Petrology*, 27(1).
- Geo-matching. (2023). *QPS Fledermaus Geocoder*. <https://geo-matching.com/sediment-classification-software/fledermaus-geocoder>
- Haralick, R. M., Dinstein, I., & Shanmugam, K. (1973). Textural Features for Image Classification. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-3(6), 610–621. <https://doi.org/10.1109/TSMC.1973.4309314>
- Haryanto, D., Kurnia Febriawan, H., & Haryadi, Y. (2020). Survei Dimensional untuk Pemasangan Multibeam Echosounder Laut Dalam di KR Baruna Jaya III. Dalam *14 Gedung BJ Habibie Lt. 12 Jl. MH Thamrin* (Vol. 1, Nomor 8).
- Hedrick, L. B., Anderson, J. T., Welsh, S. A., & Lin, L.-S. (2013). Sedimentation in Mountain Streams: A Review of Methods of Measurement. *Natural Resources*, 04(01), 92–104. <https://doi.org/10.4236/nr.2013.41011>
- Hellequin, L., Lurton, X., & Augustin, J. M. (1996). *Postprocessing and Signal Corrections for Multibeam Echosounder Images*. Oceans '96 MTS/IEEE Conference Committee.
- Herskind, P. (2008, Januari 1). *EIVA: Marine Software and Integrated Systems Solutions*. Hydro International. <https://www.hydro-international.com/content/article/eiva-5>
- Hovem, J. M. (2007). Underwater acoustics: Propagation, devices and systems. *Journal of Electroceramics*, 19(4), 339–347. <https://doi.org/10.1007/s10832-007-9059-9>
- Hoy, S., & Kissinger, K. (2010). Multibeam Calibration: Conducting a Patch Test. *NOAA Ship Okeanos Explorer, OER Interns*.
- Huda, M., Harahap, R., & Simbolon, R. H. T. (2022). Analisis Pengukuran Batimetri Dan Pasang Surut Untuk Menentukan Kedalaman Sungai Batang Hari Provinsi Jambi. *JTSIP*, 1(2).

- Ibrahim, D. A. F., Pratomo, D. G., Riyadi, N., & Jantarto, D. (2022). Pemanfaatan Data Kolom Air multibeam Echosounder untuk Mendeteksi Kebocoran Pipa Gas Bawah Laut. *Jurnal Chart Datum*, 4(2), 95–106. <https://doi.org/https://doi.org/10.37875/chartdatum.v4i2.131>
- IHO. (2008). *IHO Standards for Hydrographic Surveys 5th Edition Special Publication No 44*.
- Indramawan, B. S., Adi, A. P., Djunarsjah, E., & Pandoe, W. W. (2018). Analisis Nilai Hambur Balik pada Kapal Karam (Wreck) menggunakan Data Multibeam Echosounder di Perairan Belawan. <https://doi.org/https://doi.org/10.37875/chartdatum.v4i1.127>
- JavaTPoint. (2023). *Train and Test Datasets in Machine Learning*. JavaTPoint. <https://www.javatpoint.com/train-and-test-datasets-in-machine-learning>
- Jayanto, H., Manik, H. M., & Adi, A. P. (2021). Analisis Akustik Backscatter untuk Pemprofilan Dasar Laut Guna Penentuan Lokasi Duduk Kapal Selam (Studi Kasus di Perairan Laut Jawa Utara Segmen 8). <https://doi.org/https://doi.org/10.37875/chartdatum.v7i2.217>
- Ji, X., Yang, B., & Tang, Q. 2020. Seabed sediment classification using multibeam backscatter data based on the selecting optimal random forest model. *Applied Acoustics*, 167. <https://doi.org/10.1016/j.apacoust.2020.107387>
- Josep, A. A. (2019). Analisis Manfaat dalam Proyek Pengerukan Studi Kasus: Alur Pelayaran Surabaya Timur. *Jurnal Penelitian Transportasi Laut*, 21(1), 35–40. <https://doi.org/10.25104/transla.v21i1.1168>
- Julpan, Budhiarti Nababan, E., & Zarlis, M. (2015). Analisis Fungsi Aktivasi Sigmoid Biner dan Sigmoid Bipolar dalam Algoritma Backpropagation pada Prediksi Kemampuan Siswa. Dalam *Jurnal Teknovasi* (Vol. 02, Nomor 1).
- Keras. (2023). *Probabilistic Losses*. [https://keras.io/api/losses/probabilistic\\_losses/#categorical\\_crossentropy-function](https://keras.io/api/losses/probabilistic_losses/#categorical_crossentropy-function)
- Khomsin, Mukhtasor, Pratomo, D. G., & Suntoyo. (2021). The Development of Seabed Sediment Mapping Methods: The Opportunity Application in the Coastal Waters. *IOP Conference Series: Earth and Environmental Science*, 731(1). <https://doi.org/10.1088/1755-1315/731/1/012039>
- Khomsin, Pratomo, D. G., & Saputro, I. (2021). Comparative analysis of singlebeam and multibeam echosounder bathymetric data. *IOP Conference Series: Materials Science and Engineering*, 1052(1), 012015. <https://doi.org/10.1088/1757-899x/1052/1/012015>
- Kingma, D. P., & Ba, J. (2014). *Adam: A Method for Stochastic Optimization*. <http://arxiv.org/abs/1412.6980>
- Le Bas, T. P., & Huvenne, V. A. I. (2009). Acquisition and Processing of Backscatter Data for Habitat Mapping - Comparison of Multibeam and Sidescan Systems. *Applied Acoustics*, 70(10), 1248–1257. <https://doi.org/10.1016/j.apacoust.2008.07.010>
- Lesmana, N. T., & Haykal, F. (2021). Pemetaan Batimetri dalam Perencanaan Pembangunan Pesisir. *Journal of Empowerment Community and Education*, 1(1).
- Lubis, M. S. Y. (2021). Implementasi Artificial Intelligence pada Sistem Manufaktur Terpadu.
- Manik, H. M., Hartoyo, D., & Rohman, S. (2014). Underwater Multiple Objects Detection and Tracking using Multibeam and Side Scan Sonar. *International Journal of Applied Information Systems (IJ AIS)*, 7(4). [www.ijais.org](http://www.ijais.org)
- Manik, H. M., Junaedi, L., & Harsono, G. (2016). Pemrosesan Citra Side Scan Sonar untuk Pemetaan Dasar Laut Pelabuhan Benoa. *JNTETI*, 5(2).
- Marine Matcher. (2023). *Hydrographic Surveying*. <https://marinematcher.com/service/hydrographic-surveying/>
- MathWorks. (2023). *What Is Reinforcement Learning?* <https://www.mathworks.com/discovery/reinforcement-learning.html>

- Muljawan, D., Haryanto, D., & Ilyas, M. (2020). Pacht Test Calibration of Deep Water Multibeam Echosounder In RV. Baruna Jaya-I. *Jurnal Riset dan Rekayasa Kelautan*, 1(1). <https://doi.org/https://doi.org/10.29122/oseanika.v1i1.4055>
- Musu, W., & Ibrahim, A. (2021). *Pengaruh Komposisi Data Training dan Testing terhadap Akurasi Algoritma C4.5*.
- Nikawanti, G., & Aca, R. (2021). Ecoliteracy: Membangun Ketahanan Pangan dari Kekayaan Maritim Indonesia. Dalam *Jurnal Kemaritiman: Indonesian Journal of Maritime* (Vol. 2, Nomor 2).
- Nitriansyah, R., & Cahyono, B. K. (2022). Seabed Classification Using Multibeam Echosounder Measurement Data. *IOP Conference Series: Earth and Environmental Science*, 1039(1). <https://doi.org/10.1088/1755-1315/1039/1/012045>
- Nugroho, C., Manik, H., Gultom, D., & Firdaus, M. (2022). Implementasi Multibeam Echosounder untuk Pengukuran dan Analisis Data Kedalaman Perairan Teluk Jakarta Berdasarkan Standar International Hydrographic Organization. *POSITRON*, 12(1), 60. <https://doi.org/10.26418/positron.v12i1.51833>
- NIWA. (2023). *New Zealand Bathymetry*. NIWA . <https://niwa.co.nz/our-science/oceans/bathymetry/further-information>
- Nugroho, C., Manik, H., Gultom, D., & Firdaus, M. (2022). Implementasi Multibeam Echosounder untuk Pengukuran dan Analisis Data Kedalaman Perairan Teluk Jakarta Berdasarkan Standar International Hydrographic Organization. *POSITRON*, 12(1), 60. <https://doi.org/10.26418/positron.v12i1.51833>
- Nugroho, S. H., & Basit, A. (2014). Sebaran Sedimen Berdasarkan Analisis Ukuran Butir di Teluk Weda, Maluku Utara. Dalam *Jurnal Ilmu dan Teknologi Kelautan Tropis* (Vol. 6, Nomor 1). [http://itk.fpik.ipb.ac.id/ej\\_itkt61](http://itk.fpik.ipb.ac.id/ej_itkt61)
- Pamungkas, I., & Alam, S. (2022). Studi Komparasi Fungsi Aktivasi Sigmoid Biner, Sigmoid Bipolar dan Linear pada Jaringan Saraf Tiruan dalam Menentukan Warna RGB Menggunakan Matlab. *Serambi Engineering*, VII(4).
- Pettit, R. W., Fullem, R., Cheng, C., & Amos, C. I. 2021. Artificial intelligence, machine learning, and deep learning for clinical outcome prediction. In *Emerging Topics in Life Sciences* (Vol. 5, Issue 6, pp. 729–745). Portland Press Ltd. <https://doi.org/10.1042/ETLS20210246>
- Ponce, V.M., 1989, *Engineering Hydrology, Principles and Practice*, Prentice-Hall Inc., New Jersey
- Prananda, A. R. A., Merici, A., Huda, A. N., Amalia, A., Nastiti, A., Wijayanto, G. N., Alfi, H. N., Yuda, L. D., Kartika, M. R., & Wibowo, T. W. (2017). *Pembuatan Peta Batimetri dengan Menggunakan Metode Hidroakustik Studi Kasus Sebagian Sungai Cijulang Kabupaten Pangandaran, Jawa Barat*.
- Purnawan, S., Haridhi, H. A., Setiawan, I., & Merwantim. (2015). Parameter Statistik Ukuran Butiran pada Sedimen Berpasir di Muara Kuala Gigieng, Kabupaten Aceh Besar.
- Ramadhan, M. L., Sasmito, B., & Hadi, F. (2021). Analisis Pengaruh Nilai Kekeruhan Air Terhadap Akurasi Satellite Derived Bathymetry dengan Algoritma STUMPF (Studi Kasus: Pantai Kartini, Jawa Tengah). *Jurnal Geodesi Undip*, 10(2).
- Raut, A., Mayur, P. A., Kamble, S. D., & Dhondrikar, C. P. (2016). Texture Parameters Extraction of Satellite Image. Dalam *IJSTE-International Journal of Science Technology & Engineering* | (Vol. 2, Nomor 11). [www.ijste.org](http://www.ijste.org)
- Rompas, N.F., Jasin, M.I., & Tawas, H.J. 2022. Analisis Pasang Surut di Pantai Mahembang Kecamatan Kakas Kabupaten Minahasa Provinsi Sulawesi Utara. *Jurnal Sipil Statik* Vol. 10 No. 1 Januari 2022.
- Rifardi. (2012). *Ekologi Sedimen Laut Modern*.

- Rosmala, D., & Dwipa, G. (2012). Pembangunan Website Content Monitoring System Menggunakan Difflib Python. *Jurnal Informatika ITN Bandung*, 3(3).
- Ruder, S. (2016). *An Overview of Gradient Descent Optimization Algorithms*. <http://arxiv.org/abs/1609.04747>
- Rustam, I. (2016). Tantangan ALKI dalam Mewujudkan Cita-cita Indonesia sebagai Poros Maritim Dunia (Vol. 1, Nomor 1). <https://doi.org/https://doi.org/10.14710/ip.v1i1.10426>
- Sarono, S., & Basith, A. (2022). Uji Kualitas Data Pengukuran Batimetri Singlebeam Echosounder Berdasarkan SNI-7647 Tahun 2010 (Studi Kasus Survei Batimetri Menggunakan Hi-Target HD 370 di Laguna Pantai Glagah, Kulon Progo). *JGISE: Journal of Geospatial Information Science and Engineering*, 5(1), 21. <https://doi.org/10.22146/jgise.69749>
- Sasmito, B. (2020). Analisis Perubahan Garis Pantai Akibat Kenaikan Muka Air Laut Pantai Kabupaten Demak (Vol. 3, Nomor 2).
- Sayyid, D. N., Ningsih, E. N., & Diansyah, G. (2020). Survei Batimetri Sungai Banyuasin Menggunakan Single Beam Echosounder. *MASPARI Journal*, 12(2), 37–44.
- Sharma, S., Sharma, S., & Athaiya, A. (2020). Activation Functions in Neural Networks. Dalam *International Journal of Engineering Applied Sciences and Technology* (Vol. 4). <http://www.ijeast.com>
- Singh, S. (2018). *Cousins of Artificial Intelligence*. Toward Data Science. <https://towardsdatascience.com/cousins-of-artificial-intelligence-dda4edc27b55>
- Stephanie, Widada, S., & Rahardiawan, R. (2014). Interpretasi Lapisan Sedimen Dasar Laut di Perairan Utara Jawa (Lembar Peta 1310) Menggunakan Sub Bottom Profile. *Jurnal Oseanografi*, 3(4), 504–510. <http://ejournal-s1.undip.ac.id/index.php/jose>
- Suhaemi, Raharjo, S., & Marhan. (2018). Penentuan Tipe Pasang Surut Perairan pada Alur Pelayaran Manokwari dengan Menggunakan Metode Admiralty (Vol. 2, Nomor 1). [www.ejournalfpikunipa.ac.id](http://www.ejournalfpikunipa.ac.id)
- Sun, W., Kolappal, A. Z., & Gong, P. (2005). Two Computation Methods for Detecting Anisotropy in Image Texture. *Geographic Information Sciences*, 11(2), 87–96. <https://doi.org/10.1080/10824000509480604>
- Usman, K. O. (2014). Analisis Sedimentasi pada Muara Sungai Komerling Kota Palembang. Valeport. (2023). *miniSVP - Sound Velocity Profiler*. <https://www.valeport.co.uk/products/minisvp-sound-velocity-profiler/>
- Widodo, R., Widodo, A. W., & Supriyanto, A. (2018). Pemanfaatan Ciri Gray Level Co-Occurrence Matrix (GLCM) Citra Buah Jeruk Keprok (*Citrus reticulata* Blanco) untuk Klasifikasi Mutu. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(11), 5769–5776. <http://j-ptiik.ub.ac.id>
- Wiranda, N., Purba, H. S., & Sukmawati, R. A. (2020). Survei Penggunaan Tensorflow pada Machine Learning untuk Identifikasi Ikan Kawasan Lahan Basah. *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, 10(2), 179. <https://doi.org/10.22146/ijeis.58315>
- Zhou, P., Chen, G., Wang, M., Chen, J., & Li, Y. 2020. Sediment classification of acoustic backscatter image based on stacked denoising autoencoder and modified extreme learning machine. *Remote Sensing*, 12(22), 1–18. <https://doi.org/10.3390/rs12223762>
- Zhang, S., Xu, X., Xu, D., Long, K., Shen, C., & Tian, C. (2022). The Design and Calibration of a Low-cost Underwater Sound Velocity Profiler. *Frontiers in Marine Science*, 9. <https://doi.org/10.3389/fmars.2022.996299>
- Zhu, Z., Cui, X., Zhang, K., Ai, B., Shi, B., & Yang, F. (2021). DNN-based Seabed Classification using Differently Weighted MBES Multifeatures. *Marine Geology*, 438. <https://doi.org/10.1016/j.margeo.2021.106519>

*“Halaman ini sengaja dikosongkan”*

## LAMPIRAN

### Lampiran 1. Script Pengolahan *Dense Neural Network* Model 1

```
# 1. Load data
import numpy as np
import pandas as pd
from google.colab import drive
from keras.utils.np_utils import to_categorical
drive.mount('/content/drive')
data = pd.read_csv(r"/content/drive/MyDrive/Tugas
Akhir/FileDNN/DNN_Sedimenbaru.csv")
print(data)

# Indeks baris yang akan dijadikan data testing
testing_indices = [3, 5, 7]

# Memisahkan data training dan data testing
training_data = data.drop(testing_indices) #data training (kodingan untuk menghilangkan data
testing/ memisahkan dari data training)
testing_data = data.loc[testing_indices] #data testing

# Cetak hasil
print("Data training:")
print(training_data)
print("Data testing:")
print(testing_data)

# 2. Training-testing division
yTrain = training_data.iloc[:,0].values #titik dua sebelum koma itu baris semuanya
yTrain = to_categorical(yTrain)
xTrain = training_data.iloc[:,4:30].values

yTest = testing_data.iloc[:,0].values #titik dua sebelum koma itu baris semuanya
yTest = to_categorical(yTest,num_classes=6)
xTest = testing_data.iloc[:,4:30].values

#Model DNN 1
import tensorflow as tf
tf.keras.backend.clear_session()
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import Sequential

model = Sequential()
model.add(Dense(40, input_shape=(25,), activation='sigmoid'))
model.add(Dense(30, activation='relu'))
model.add(Dense(20, activation='relu'))
model.add(Dense(10, activation='relu'))
model.add(Dense(6, activation='sigmoid'))
```

```

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

model.summary()

from tensorflow.keras.callbacks import ModelCheckpoint
import matplotlib.pyplot as plt

# 2. Training
checkPointer = ModelCheckpoint(filepath="weightsFE.hdf5",
                                monitor='accuracy',
                                verbose=1,
                                save_best_only=True)

history = model.fit(xTrain, yTrain,
                    validation_data=(xTest, yTest),
                    epochs=200,
                    batch_size=8,
                    callbacks=[checkPointer]) #masalah metric accuracy

model.load_weights('weightsFE.hdf5')

#4. Testing
loss, accuracy = model.evaluate(xTrain, yTrain)
print("Training Loss:", loss)
print("Training Accuracy:", accuracy)
lossTest, accuracyTest = model.evaluate(xTest, yTest)
print("Testing Loss:", lossTest)
print("Testing Accuracy:", accuracyTest)

# 3. Evalaute the model
fig, ax = plt.subplots(figsize=(10, 8))
print(history.history.keys())
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper right')
plt.show()

fig, ax = plt.subplots(figsize=(10, 8))
print(history.history.keys())
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper right')
plt.show()

model.save('model.hdf5')

```

## Lampiran 2. Script Pengolahan *Dense Neural Network* Model 2

```
# 1. Load data
import numpy as np
import pandas as pd
from google.colab import drive
from keras.utils.np_utils import to_categorical
drive.mount('/content/drive')
data = pd.read_csv(r"/content/drive/MyDrive/Tugas
Akhir/FileDNN/DNN_Sedimenbaru.csv")
print(data)

# Indeks baris yang akan dijadikan data testing
testing_indices = [3, 5, 7]

# Memisahkan data training dan data testing
training_data = data.drop(testing_indices) #data training (kodingan untuk menghilangkan data
testing/ memisahkan dari data training)
testing_data = data.loc[testing_indices] #data testing

# Cetak hasil
print("Data training:")
print(training_data)
print("Data testing:")
print(testing_data)

# 2. Training-testing division
yTrain = training_data.iloc[:,0].values #titik dua sebelum koma itu baris semuanya
yTrain = to_categorical(yTrain)
xTrain = training_data.iloc[:,4:30].values

yTest = testing_data.iloc[:,0].values #titik dua sebelum koma itu baris semuanya
yTest = to_categorical(yTest,num_classes=6)
xTest = testing_data.iloc[:,4:30].values

#Model DNN 2
import tensorflow as tf
tf.keras.backend.clear_session()
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import Sequential

model = Sequential()
model.add(Dense(10, input_shape=(25,), activation='sigmoid'))
model.add(Dense(25, activation='relu'))
model.add(Dense(30, activation='relu'))
model.add(Dense(45, activation='relu'))
model.add(Dense(6, activation='sigmoid'))
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

```

model.summary()

from tensorflow.keras.callbacks import ModelCheckpoint
import matplotlib.pyplot as plt

# 2. Training
checkPointer = ModelCheckpoint(filepath="weightsFE.hdf5",
                               monitor='accuracy',
                               verbose=1,
                               save_best_only=True)

history = model.fit(xTrain, yTrain,
                   validation_data=(xTest, yTest),
                   epochs=200,
                   batch_size=8,
                   callbacks=[checkPointer]) #masalah metric accuracy

model.load_weights('weightsFE.hdf5')

#4. Testing
loss, accuracy = model.evaluate(xTrain, yTrain)
print("Training Loss:", loss)
print("Training Accuracy:", accuracy)
lossTest, accuracyTest = model.evaluate(xTest, yTest)
print("Testing Loss:", lossTest)
print("Testing Accuracy:", accuracyTest)

# 3. Evalaute the model
fig, ax = plt.subplots(figsize=(10, 8))
print(history.history.keys())
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper right')
plt.show()

fig, ax = plt.subplots(figsize=(10, 8))
print(history.history.keys())
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper right')
plt.show()

model.save('model.hdf5')

```

### Lampiran 3. Script Pengolahan *Dense Neural Network* Model 3

```
# 1. Load data
import numpy as np
import pandas as pd
from google.colab import drive
from keras.utils.np_utils import to_categorical
drive.mount('/content/drive')
data = pd.read_csv(r"/content/drive/MyDrive/Tugas
Akhir/FileDNN/DNN_Sedimenbaru.csv")
print(data)

# Indeks baris yang akan dijadikan data testing
testing_indices = [3, 5, 7]

# Memisahkan data training dan data testing
training_data = data.drop(testing_indices) #data training (kodingan untuk menghilangkan data
testing/ memisahkan dari data training)
testing_data = data.loc[testing_indices] #data testing

# Cetak hasil
print("Data training:")
print(training_data)
print("Data testing:")
print(testing_data)

# 2. Training-testing division
yTrain = training_data.iloc[:,0].values #titik dua sebelum koma itu baris semuanya
yTrain = to_categorical(yTrain)
xTrain = training_data.iloc[:,4:30].values

yTest = testing_data.iloc[:,0].values #titik dua sebelum koma itu baris semuanya
yTest = to_categorical(yTest,num_classes=6)
xTest = testing_data.iloc[:,4:30].values

#Model DNN 3
import tensorflow as tf
tf.keras.backend.clear_session()
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import Sequential

model = Sequential()
model.add(Dense(25, input_shape=(25,), activation='sigmoid'))
model.add(Dense(25, activation='relu'))
model.add(Dense(25, activation='relu'))
model.add(Dense(25, activation='relu'))
model.add(Dense(6, activation='sigmoid'))
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

```

model.summary()

from tensorflow.keras.callbacks import ModelCheckpoint
import matplotlib.pyplot as plt

# 2. Training
checkPointer = ModelCheckpoint(filepath="weightsFE.hdf5",
                               monitor='accuracy',
                               verbose=1,
                               save_best_only=True)

history = model.fit(xTrain, yTrain,
                   validation_data=(xTest, yTest),
                   epochs=200,
                   batch_size=8,
                   callbacks=[checkPointer]) #masalah metric accuracy

model.load_weights('weightsFE.hdf5')

#4. Testing
loss, accuracy = model.evaluate(xTrain, yTrain)
print("Training Loss:", loss)
print("Training Accuracy:", accuracy)
lossTest, accuracyTest = model.evaluate(xTest, yTest)
print("Testing Loss:", lossTest)
print("Testing Accuracy:", accuracyTest)

# 3. Evalaute the model
fig, ax = plt.subplots(figsize=(10, 8))
print(history.history.keys())
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper right')
plt.show()

fig, ax = plt.subplots(figsize=(10, 8))
print(history.history.keys())
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper right')
plt.show()

model.save('model.hdf5')

```

#### Lampiran 4. Script Pengolahan *Dense Neural Network* Model 4

```
# 1. Load data
import numpy as np
import pandas as pd
from google.colab import drive
from keras.utils.np_utils import to_categorical
drive.mount('/content/drive')
data = pd.read_csv(r"/content/drive/MyDrive/Tugas
Akhir/FileDNN/DNN_Sedimenbaru.csv")
print(data)

# Indeks baris yang akan dijadikan data testing
testing_indices = [3, 5, 7]

# Memisahkan data training dan data testing
training_data = data.drop(testing_indices) #data training (kodingan untuk menghilangkan data
testing/ memisahkan dari data training)
testing_data = data.loc[testing_indices] #data testing

# Cetak hasil
print("Data training:")
print(training_data)
print("Data testing:")
print(testing_data)

# 2. Training-testing division
yTrain = training_data.iloc[:,0].values #titik dua sebelum koma itu baris semuanya
yTrain = to_categorical(yTrain)
xTrain = training_data.iloc[:,4:30].values

yTest = testing_data.iloc[:,0].values #titik dua sebelum koma itu baris semuanya
yTest = to_categorical(yTest,num_classes=6)
xTest = testing_data.iloc[:,4:30].values

#Model DNN 4
import tensorflow as tf
tf.keras.backend.clear_session()
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import Sequential

model = Sequential()
model.add(Dense(20, input_shape=(25,), activation='sigmoid'))
model.add(Dense(30, activation='relu'))
model.add(Dense(30, activation='relu'))
model.add(Dense(20, activation='relu'))
model.add(Dense(6, activation='sigmoid'))
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

```

model.summary()

from tensorflow.keras.callbacks import ModelCheckpoint
import matplotlib.pyplot as plt

# 2. Training
checkPointer = ModelCheckpoint(filepath="weightsFE.hdf5",
                               monitor='accuracy',
                               verbose=1,
                               save_best_only=True)

history = model.fit(xTrain, yTrain,
                   validation_data=(xTest, yTest),
                   epochs=200,
                   batch_size=8,
                   callbacks=[checkPointer]) #masalah metric accuracy

model.load_weights('weightsFE.hdf5')

#4. Testing
loss, accuracy = model.evaluate(xTrain, yTrain)
print("Training Loss:", loss)
print("Training Accuracy:", accuracy)
lossTest, accuracyTest = model.evaluate(xTest, yTest)
print("Testing Loss:", lossTest)
print("Testing Accuracy:", accuracyTest)

# 3. Evalaute the model
fig, ax = plt.subplots(figsize=(10, 8))
print(history.history.keys())
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper right')
plt.show()

fig, ax = plt.subplots(figsize=(10, 8))
print(history.history.keys())
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper right')
plt.show()

model.save('model.hdf5')

```

## Lampiran 5. Script Pengolahan *Dense Neural Network* Model 5

```
# 1. Load data
import numpy as np
import pandas as pd
from google.colab import drive
from keras.utils.np_utils import to_categorical
drive.mount('/content/drive')
data = pd.read_csv(r"/content/drive/MyDrive/Tugas
Akhir/FileDNN/DNN_Sedimenbaru.csv")
print(data)

# Indeks baris yang akan dijadikan data testing
testing_indices = [3, 5, 7]

# Memisahkan data training dan data testing
training_data = data.drop(testing_indices) #data training (kodingan untuk menghilangkan data
testing/ memisahkan dari data training)
testing_data = data.loc[testing_indices] #data testing

# Cetak hasil
print("Data training:")
print(training_data)
print("Data testing:")
print(testing_data)

# 2. Training-testing division
yTrain = training_data.iloc[:,0].values #titik dua sebelum koma itu baris semuanya
yTrain = to_categorical(yTrain)
xTrain = training_data.iloc[:,4:30].values

yTest = testing_data.iloc[:,0].values #titik dua sebelum koma itu baris semuanya
yTest = to_categorical(yTest,num_classes=6)
xTest = testing_data.iloc[:,4:30].values

#Model DNN 5
import tensorflow as tf
tf.keras.backend.clear_session()
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import Sequential

model = Sequential()
model.add(Dense(35, input_shape=(25,), activation='sigmoid'))
model.add(Dense(20, activation='relu'))
model.add(Dense(20, activation='relu'))
model.add(Dense(35, activation='relu'))
model.add(Dense(6, activation='sigmoid'))
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

```

model.summary()

from tensorflow.keras.callbacks import ModelCheckpoint
import matplotlib.pyplot as plt

# 2. Training
checkPointer = ModelCheckpoint(filepath="weightsFE.hdf5",
                               monitor='accuracy',
                               verbose=1,
                               save_best_only=True)

history = model.fit(xTrain, yTrain,
                   validation_data=(xTest, yTest),
                   epochs=200,
                   batch_size=8,
                   callbacks=[checkPointer]) #masalah metric accuracy

model.load_weights('weightsFE.hdf5')

#4. Testing
loss, accuracy = model.evaluate(xTrain, yTrain)
print("Training Loss:", loss)
print("Training Accuracy:", accuracy)
lossTest, accuracyTest = model.evaluate(xTest, yTest)
print("Testing Loss:", lossTest)
print("Testing Accuracy:", accuracyTest)

# 3. Evalaute the model
fig, ax = plt.subplots(figsize=(10, 8))
print(history.history.keys())
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper right')
plt.show()

fig, ax = plt.subplots(figsize=(10, 8))
print(history.history.keys())
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper right')
plt.show()

model.save('model.hdf5')

```

## Lampiran 6. Script Pengolahan *Dense Neural Network* Model 6

```
# 1. Load data
import numpy as np
import pandas as pd
from google.colab import drive
from keras.utils.np_utils import to_categorical
drive.mount('/content/drive')
data = pd.read_csv(r"/content/drive/MyDrive/Tugas
Akhir/FileDNN/DNN_Sedimenbaru.csv")
print(data)

# Indeks baris yang akan dijadikan data testing
testing_indices = [3, 5, 7]

# Memisahkan data training dan data testing
training_data = data.drop(testing_indices) #data training (kodingan untuk menghilangkan data
testing/ memisahkan dari data training)
testing_data = data.loc[testing_indices] #data testing

# Cetak hasil
print("Data training:")
print(training_data)
print("Data testing:")
print(testing_data)

# 2. Training-testing division
yTrain = training_data.iloc[:,0].values #titik dua sebelum koma itu baris semuanya
yTrain = to_categorical(yTrain)
xTrain = training_data.iloc[:,4:30].values

yTest = testing_data.iloc[:,0].values #titik dua sebelum koma itu baris semuanya
yTest = to_categorical(yTest,num_classes=6)
xTest = testing_data.iloc[:,4:30].values

#Model DNN 6
import tensorflow as tf
tf.keras.backend.clear_session()
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import Sequential

model = Sequential()
model.add(Dense(35, input_shape=(25,), activation='sigmoid'))
model.add(Dense(20, activation='relu'))
model.add(Dense(40, activation='relu'))
model.add(Dense(15, activation='relu'))
model.add(Dense(6, activation='sigmoid'))
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

```

model.summary()

from tensorflow.keras.callbacks import ModelCheckpoint
import matplotlib.pyplot as plt

# 2. Training
checkPointer = ModelCheckpoint(filepath="weightsFE.hdf5",
                               monitor='accuracy',
                               verbose=1,
                               save_best_only=True)

history = model.fit(xTrain, yTrain,
                   validation_data=(xTest, yTest),
                   epochs=200,
                   batch_size=8,
                   callbacks=[checkPointer]) #masalah metric accuracy

model.load_weights('weightsFE.hdf5')

#4. Testing
loss, accuracy = model.evaluate(xTrain, yTrain)
print("Training Loss:", loss)
print("Training Accuracy:", accuracy)
lossTest, accuracyTest = model.evaluate(xTest, yTest)
print("Testing Loss:", lossTest)
print("Testing Accuracy:", accuracyTest)

# 3. Evalaute the model
fig, ax = plt.subplots(figsize=(10, 8))
print(history.history.keys())
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper right')
plt.show()

fig, ax = plt.subplots(figsize=(10, 8))
print(history.history.keys())
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper right')
plt.show()

model.save('model.hdf5')

```

## Lampiran 7. Script Pengolahan Klasifikasi dan *Mapping*

#Klasifikasi dan Mapping

```
from google.colab import drive
drive.mount('/content/drive')

import pandas as pd
import numpy as np
from keras.models import load_model
from keras.utils import to_categorical

# Load model yang telah dilatih sebelumnya
model = load_model(r"/content/drive/MyDrive/Colab Notebooks/weightsFE.hdf5")

# Menyiapkan data baru untuk diprediksi
new_data = pd.read_csv(r"/content/drive/MyDrive/Tugas Akhir/Klasifikasi/Valuep.csv")

# Melakukan prediksi menggunakan data baru
predictions = model.predict(new_data)
predicted_labels = np.argmax(predictions, axis=1)

# Mengubah hasil prediksi menjadi kategori
predicted_labels_categorical = to_categorical(predicted_labels)

# Menampilkan hasil prediksi
print(predicted_labels_categorical)

import pandas as pd

# Membuat DataFrame dari ndarray
df = pd.DataFrame(data=predicted_labels_categorical)

# Menyimpan DataFrame ke file Excel
output_path = '/content/drive/MyDrive/Tugas Akhir/Klasifikasi/hasil1.xlsx'
df.to_excel(output_path, index=False)
```

**Lampiran 8. Data Pasang Surut**

<b>Hari</b>	<b>Tanggal</b>	<b>Tahun</b>	<b>Jam</b>	<b>Menit</b>	<b>Detik</b>	<b>Bacaan</b>
31	1	2022	8	57	0	1.606
31	1	2022	9	7	0	1.594
31	1	2022	9	17	0	1.598
31	1	2022	9	27	0	1.589
31	1	2022	9	37	0	1.565
31	1	2022	9	47	0	1.549
31	1	2022	9	57	0	1.542
31	1	2022	10	7	0	1.521
31	1	2022	10	17	0	1.496
31	1	2022	10	27	0	1.453
31	1	2022	10	37	0	1.426
31	1	2022	10	47	0	1.395
31	1	2022	10	57	0	1.347
31	1	2022	11	7	0	1.322
31	1	2022	11	17	0	1.275
31	1	2022	11	27	0	1.256
31	1	2022	11	37	0	1.213
31	1	2022	11	47	0	1.169
31	1	2022	11	57	0	1.134
31	1	2022	12	7	0	1.107
31	1	2022	12	17	0	1.089
31	1	2022	12	27	0	1.042
31	1	2022	12	37	0	1.013
31	1	2022	12	47	0	0.988
31	1	2022	12	57	0	0.969
31	1	2022	13	7	0	0.933
31	1	2022	13	17	0	0.939
31	1	2022	13	27	0	0.92
31	1	2022	13	37	0	0.915
31	1	2022	13	47	0	0.899
31	1	2022	13	57	0	0.898
31	1	2022	14	7	0	0.898
31	1	2022	14	17	0	0.914
31	1	2022	14	27	0	0.926
31	1	2022	14	37	0	0.955
31	1	2022	14	47	0	0.967
31	1	2022	14	57	0	0.99
31	1	2022	15	7	0	1.017
31	1	2022	15	17	0	1.069
31	1	2022	15	27	0	1.094

Hari	Tanggal	Tahun	Jam	Menit	Detik	Bacaan
31	1	2022	15	37	0	1.132
31	1	2022	15	47	0	1.187
31	1	2022	15	57	0	1.233
31	1	2022	16	7	0	1.27
31	1	2022	16	17	0	1.338

**Lampiran 9.** Data *Sound Velocity Profiler*

No	<i>Sound Velocity</i>	Tekanan	Kedalaman	Waktu	Date
1	1541.212	0.03	0.03	5:55:53	1/31/2022
2	1542.067	0.01	0.01	5:55:53	1/31/2022
3	1541.968	0.01	0.01	5:55:53	1/31/2022
4	1541.439	0.02	0.02	5:55:53	1/31/2022
5	1541.521	0.03	0.03	5:55:53	1/31/2022
6	1541.484	0.07	0.07	5:55:53	1/31/2022
7	1541.435	0.07	0.07	5:55:53	1/31/2022
8	1541.893	0.08	0.08	5:55:53	1/31/2022
9	1541.879	0.11	0.11	5:55:53	1/31/2022
10	1541.853	0.09	0.09	5:55:53	1/31/2022
11	1541.843	0.11	0.11	5:55:53	1/31/2022
12	1541.831	0.11	0.11	5:55:53	1/31/2022
13	1541.839	0.09	0.09	5:55:53	1/31/2022
14	1541.837	0.12	0.11	5:55:53	1/31/2022
15	1541.841	0.11	0.11	5:55:53	1/31/2022
16	1541.838	0.1	0.09	5:55:53	1/31/2022
17	1541.835	0.08	0.08	5:55:54	1/31/2022
18	1541.834	0.12	0.12	5:55:54	1/31/2022
19	1541.83	0.11	0.11	5:55:54	1/31/2022
20	1541.827	0.11	0.11	5:55:54	1/31/2022
21	1541.828	0.08	0.08	5:55:54	1/31/2022
22	1541.828	0.09	0.09	5:55:54	1/31/2022
23	1541.828	0.09	0.09	5:55:54	1/31/2022
24	1541.827	0.09	0.09	5:55:54	1/31/2022
25	1541.827	0.1	0.1	5:55:54	1/31/2022
26	1541.828	0.14	0.14	5:55:54	1/31/2022
27	1541.827	0.14	0.14	5:55:54	1/31/2022
28	1541.828	0.13	0.13	5:55:54	1/31/2022
29	1541.827	0.19	0.19	5:55:54	1/31/2022
30	1541.827	0.18	0.18	5:55:54	1/31/2022
31	1541.824	0.17	0.17	5:55:54	1/31/2022
32	1541.818	0.18	0.18	5:55:54	1/31/2022

<b>No</b>	<b><i>Sound Velocity</i></b>	<b>Tekanan</b>	<b>Kedalaman</b>	<b>Waktu</b>	<b>Date</b>
33	1541.815	0.18	0.18	5:55:54	1/31/2022
34	1541.81	0.19	0.19	5:55:54	1/31/2022
35	1541.805	0.21	0.21	5:55:54	1/31/2022
36	1541.799	0.21	0.21	5:55:54	1/31/2022
37	1541.797	0.2	0.2	5:55:55	1/31/2022
38	1541.796	0.2	0.2	5:55:55	1/31/2022
39	1541.795	0.22	0.21	5:55:55	1/31/2022
40	1541.79	0.18	0.18	5:55:55	1/31/2022
41	1541.787	0.2	0.2	5:55:55	1/31/2022
42	1541.789	0.21	0.2	5:55:55	1/31/2022
43	1541.792	0.19	0.19	5:55:55	1/31/2022
44	1541.801	0.19	0.19	5:55:55	1/31/2022
45	1541.802	0.21	0.2	5:55:55	1/31/2022
46	1541.808	0.23	0.23	5:55:55	1/31/2022
47	1541.814	0.2	0.2	5:55:55	1/31/2022
48	1541.822	0.21	0.21	5:55:55	1/31/2022
49	1541.838	0.21	0.2	5:55:55	1/31/2022
50	1541.838	0.18	0.18	5:55:55	1/31/2022
51	1541.837	0.22	0.22	5:55:55	1/31/2022
52	1541.849	0.2	0.2	5:55:55	1/31/2022
53	1541.881	0.21	0.21	5:55:55	1/31/2022
54	1541.895	0.23	0.23	5:55:55	1/31/2022
55	1541.898	0.2	0.2	5:55:55	1/31/2022
56	1541.892	0.22	0.22	5:55:55	1/31/2022
57	1541.89	0.2	0.2	5:55:56	1/31/2022
58	1541.886	0.21	0.21	5:55:56	1/31/2022
59	1541.886	0.22	0.22	5:55:56	1/31/2022
60	1541.887	0.18	0.18	5:55:56	1/31/2022
61	1541.884	0.23	0.23	5:55:56	1/31/2022
62	1541.878	0.23	0.23	5:55:56	1/31/2022
63	1541.867	0.2	0.2	5:55:56	1/31/2022
64	1541.867	0.19	0.19	5:55:56	1/31/2022
65	1541.873	0.26	0.26	5:55:56	1/31/2022
66	1541.875	0.21	0.21	5:55:56	1/31/2022
67	1541.874	0.24	0.24	5:55:56	1/31/2022
68	1541.872	0.23	0.23	5:55:56	1/31/2022
69	1541.869	0.27	0.26	5:55:56	1/31/2022
70	1541.872	0.3	0.3	5:55:56	1/31/2022
71	1541.878	0.28	0.28	5:55:56	1/31/2022
72	1541.886	0.29	0.29	5:55:56	1/31/2022

No	<i>Sound Velocity</i>	Tekanan	Kedalaman	Waktu	Date
73	1541.89	0.29	0.29	5:55:56	1/31/2022
74	1541.89	0.32	0.31	5:55:56	1/31/2022
75	1541.894	0.31	0.31	5:55:56	1/31/2022
76	1541.895	0.33	0.33	5:55:56	1/31/2022
77	1541.896	0.31	0.31	5:55:57	1/31/2022
78	1541.894	0.35	0.34	5:55:57	1/31/2022
79	1541.897	0.36	0.36	5:55:57	1/31/2022
80	1541.894	0.38	0.38	5:55:57	1/31/2022
81	1541.892	0.35	0.35	5:55:57	1/31/2022
82	1541.895	0.37	0.37	5:55:57	1/31/2022
83	1541.901	0.36	0.36	5:55:57	1/31/2022
84	1541.91	0.37	0.37	5:55:57	1/31/2022
85	1541.92	0.39	0.39	5:55:57	1/31/2022
86	1541.919	0.37	0.37	5:55:57	1/31/2022
87	1541.916	0.36	0.36	5:55:57	1/31/2022
88	1541.91	0.34	0.34	5:55:57	1/31/2022
89	1541.904	0.36	0.36	5:55:57	1/31/2022
90	1541.902	0.34	0.34	5:55:57	1/31/2022
91	1541.902	0.38	0.38	5:55:57	1/31/2022
92	1541.903	0.35	0.35	5:55:57	1/31/2022
93	1541.91	0.35	0.35	5:55:57	1/31/2022
94	1541.908	0.36	0.35	5:55:57	1/31/2022
95	1541.914	0.37	0.37	5:55:57	1/31/2022
96	1541.915	0.37	0.37	5:55:57	1/31/2022
97	1541.916	0.37	0.37	5:55:58	1/31/2022
98	1541.919	0.36	0.36	5:55:58	1/31/2022
99	1541.918	0.39	0.39	5:55:58	1/31/2022
100	1541.919	0.39	0.39	5:55:58	1/31/2022
...	...	...	...	...	...
4678	1541.664	0.1	0.1	5:59:47	1/31/2022
4679	1541.666	0.07	0.07	5:59:47	1/31/2022
4680	1541.667	0	0	5:59:47	1/31/2022
4681	1541.659	-0.01	-0.01	5:59:47	1/31/2022
4682	1541.658	0	0	5:59:47	1/31/2022
4683	0	-0.01	-0.01	5:59:47	1/31/2022
4684	0	-0.05	-0.05	5:59:47	1/31/2022
4685	0	-0.01	-0.01	5:59:47	1/31/2022
4686	0	-0.02	-0.02	5:59:47	1/31/2022
4687	0	0	0	5:59:47	1/31/2022
4688	0	-0.02	-0.02	5:59:47	1/31/2022

**Lampiran 10.** Data Intensitas Tiap Titik

No	EASTING	NORTHING	ASM 170	CON 170	COR 170	ENER 170	DB 170	ASM 220	CON 220
1	302799.21	9032367.92	4.00	0.00	1.00	2.00	-56.67	4.00	0.00
2	302800.21	9032367.92	2.91	0.31	1.00	1.71	-24.92	4.00	0.00
3	302801.21	9032367.92	3.09	0.25	1.00	1.76	-23.00	4.00	0.00
4	302802.21	9032367.92	3.19	0.22	1.00	1.78	-25.47	2.69	0.38
5	302803.21	9032367.92	3.68	0.08	1.00	1.92	-33.81	2.47	0.46
6	302804.21	9032367.92	2.50	0.45	1.00	1.58	-26.42	2.60	0.41
7	302805.21	9032367.92	1.70	3.16	1.00	1.31	-28.34	3.12	0.24
8	302806.21	9032367.92	1.83	2.93	1.00	1.35	-29.30	1.91	0.58
9	302807.21	9032367.92	2.59	2.44	1.00	1.61	-29.57	1.73	0.64
10	302808.21	9032367.92	2.63	1.14	1.00	1.62	-30.53	1.82	0.60
11	302809.21	9032367.92	1.86	2.49	1.00	1.36	-33.40	2.82	0.29
12	302810.21	9032367.92	1.36	3.40	1.00	1.17	-32.45	2.19	0.75
13	302811.21	9032367.92	1.12	4.30	1.00	1.06	-34.77	1.89	0.74
14	302812.21	9032367.92	1.61	2.33	1.00	1.27	-34.77	1.30	2.10
15	302813.21	9032367.92	1.83	1.70	1.00	1.35	-31.21	1.42	2.77
16	302814.21	9032367.92	1.98	1.63	1.00	1.41	-29.57	1.20	4.10
17	302815.21	9032367.92	1.52	2.30	1.00	1.23	-29.30	1.03	3.80
18	302816.21	9032367.92	1.36	2.67	1.00	1.17	-29.57	0.92	3.90
19	302817.21	9032367.92	1.94	1.63	1.00	1.39	-32.17	0.89	3.53
20	302818.21	9032367.92	3.02	0.57	1.00	1.74	-33.40	0.89	4.33
21	302819.21	9032367.92	3.37	0.27	1.00	1.84	-35.32	1.06	3.50
22	302820.21	9032367.92	2.51	0.63	1.00	1.59	-33.40	1.50	2.63
23	302821.21	9032367.92	1.81	0.93	1.00	1.34	-33.40	1.88	1.00
24	302822.21	9032367.92	1.69	1.00	1.00	1.30	-34.77	1.85	0.77
25	302823.21	9032367.92	1.40	1.70	1.00	1.18	-34.36	1.73	0.83
26	302824.21	9032367.92	0.98	3.13	1.00	0.99	-34.36	1.81	1.07
27	302825.21	9032367.92	0.65	4.60	1.00	0.80	-34.36	2.60	0.70
28	302826.21	9032367.92	0.51	5.80	1.00	0.72	-36.28	2.40	0.97
29	302827.21	9032367.92	0.44	5.40	1.00	0.66	-35.73	2.15	0.87
30	302828.21	9032367.92	0.30	6.47	1.00	0.55	-33.81	1.63	1.20
31	302829.21	9032367.92	0.27	6.60	1.00	0.52	-34.09	1.98	1.13
32	302830.21	9032367.92	0.30	7.50	1.00	0.55	-34.09	1.87	1.67
33	302831.21	9032367.92	0.30	6.37	1.00	0.55	-38.88	2.03	1.37
34	302832.21	9032367.92	0.30	6.17	1.00	0.55	-38.47	1.72	1.27
35	302833.21	9032367.92	0.29	5.57	1.00	0.54	-37.92	2.05	1.03
36	302834.21	9032367.92	0.33	5.87	1.00	0.57	-34.09	1.87	1.43
37	302835.21	9032367.92	0.30	7.30	1.00	0.55	-34.09	2.12	1.33
38	302836.21	9032367.92	0.38	6.63	1.00	0.62	-34.77	1.60	1.23
39	302837.21	9032367.92	0.36	5.60	1.00	0.60	-39.84	1.45	0.87

No	EASTING	NORTHING	ASM 170	CON 170	COR 170	ENER 170	DB 170	ASM 220	CON 220
40	302838.21	9032367.92	0.56	3.80	1.00	0.75	-38.88	1.00	2.70
41	302839.21	9032367.92	0.63	4.00	1.00	0.79	-35.05	0.96	2.80
42	302840.21	9032367.92	0.80	3.60	1.00	0.89	-35.32	0.89	2.70
43	302841.21	9032367.92	0.84	3.17	1.00	0.92	-35.32	0.90	1.87
44	302842.21	9032367.92	1.00	1.40	1.00	1.00	-36.28	0.76	2.87
45	302843.21	9032367.92	1.05	1.23	1.00	1.02	-34.36	0.65	2.97
46	302844.21	9032367.92	1.18	0.90	1.00	1.08	-33.40	0.56	3.63
47	302845.21	9032367.92	1.13	0.93	1.00	1.06	-32.17	0.54	3.37
48	302846.21	9032367.92	0.92	0.97	1.00	0.96	-32.86	0.48	4.30
49	302847.21	9032367.92	0.80	1.33	1.00	0.89	-33.13	0.41	4.03
50	302848.21	9032367.92	0.68	1.60	1.00	0.83	-35.05	0.46	3.27
51	302849.21	9032367.92	0.62	1.60	1.00	0.79	-36.69	0.44	3.57
52	302850.21	9032367.92	0.90	1.37	1.00	0.95	-37.24	0.44	4.20
53	302851.21	9032367.92	0.71	1.73	1.00	0.85	-37.24	0.55	3.70
54	302852.21	9032367.92	0.64	1.97	1.00	0.80	-36.69	0.67	2.57
55	302853.21	9032367.92	0.51	2.67	1.00	0.71	-36.69	0.84	1.57
56	302854.21	9032367.92	0.59	2.20	1.00	0.77	-36.69	0.83	1.47
57	302855.21	9032367.92	0.62	2.00	1.00	0.79	-35.05	0.78	1.67
58	302856.21	9032367.92	0.67	2.00	1.00	0.82	-33.13	0.79	1.83
59	302857.21	9032367.92	0.54	2.50	1.00	0.73	-33.13	0.71	2.17
60	302858.21	9032367.92	0.64	2.33	1.00	0.80	-32.86	0.75	2.10
61	302859.21	9032367.92	0.65	1.80	1.00	0.80	-33.40	0.75	2.03
62	302860.21	9032367.92	0.60	1.93	1.00	0.78	-34.77	0.96	1.67
63	302861.21	9032367.92	0.58	2.17	1.00	0.76	-34.77	0.86	1.63
64	302862.21	9032367.92	0.62	2.30	1.00	0.79	-34.36	1.02	1.30
65	302863.21	9032367.92	0.78	1.83	1.00	0.88	-35.73	1.08	1.37
66	302864.21	9032367.92	0.83	2.00	1.00	0.91	-35.05	1.27	0.93
67	302865.21	9032367.92	0.87	2.00	1.00	0.93	-33.40	0.99	1.37
68	302866.21	9032367.92	0.91	2.03	1.00	0.95	-34.09	0.82	1.70
69	302867.21	9032367.92	0.99	1.23	1.00	1.00	-33.40	0.73	2.00
70	302868.21	9032367.92	1.07	1.47	1.00	1.03	-34.77	0.87	1.87
71	302869.21	9032367.92	1.07	1.47	1.00	1.03	-31.49	1.01	2.03
72	302870.21	9032367.92	1.22	1.30	1.00	1.11	-33.40	1.09	2.07
73	302871.21	9032367.92	1.35	1.27	1.00	1.16	-31.90	1.20	2.13
74	302872.21	9032367.92	1.83	1.20	1.00	1.35	-32.86	1.83	1.20
75	302873.21	9032367.92	1.98	1.13	1.00	1.41	-33.13	2.92	0.60
76	302874.21	9032367.92	2.17	0.87	1.00	1.47	-33.40	3.62	0.10
77	302875.21	9032367.92	2.21	0.57	1.00	1.49	-34.36	3.62	0.10
78	302876.21	9032367.92	1.71	0.73	1.00	1.31	-34.77	3.26	0.20
79	302877.21	9032367.92	1.15	1.40	1.00	1.07	-35.05	3.15	0.23

No	EASTING	NORTHING	ASM 170	CON 170	COR 170	ENER 170	DB 170	ASM 220	CON 220
80	302878.21	9032367.92	0.85	1.93	1.00	0.92	-33.40	2.64	0.40
81	302879.21	9032367.92	0.86	2.20	1.00	0.93	-33.81	2.83	0.33
82	302880.21	9032367.92	1.00	1.80	1.00	1.00	-33.81	2.26	0.43
83	302881.21	9032367.92	0.98	2.10	1.00	0.99	-36.69	2.09	0.47
84	302882.21	9032367.92	0.85	2.10	1.00	0.92	-38.19	1.16	2.43
85	302883.21	9032367.92	1.15	2.13	1.00	1.07	-29.30	1.20	2.77
86	302884.21	9032367.92	2.04	1.23	1.00	1.43	-29.98	0.86	4.23
87	302885.21	9032367.92	3.25	0.60	1.00	1.80	-30.94	0.79	3.33
88	302886.21	9032367.92	2.63	0.37	1.00	1.62	-32.17	0.55	4.67
89	302887.21	9032367.92	1.75	2.30	1.00	1.32	-32.17	0.63	4.17
90	302888.21	9032367.92	1.34	2.80	1.00	1.16	-33.13	0.56	4.80
91	302889.21	9032367.92	2.03	2.50	1.00	1.42	-34.77	0.63	3.33
92	302890.21	9032367.92	2.23	1.37	1.00	1.49	-34.77	0.61	3.27
93	302891.21	9032367.92	1.65	2.23	1.00	1.28	-29.02	0.73	1.73
94	302892.21	9032367.92	1.14	2.63	1.00	1.07	-27.66	0.75	2.10
95	302893.21	9032367.92	1.32	2.50	1.00	1.15	-30.26	0.67	2.23
96	302894.21	9032367.92	2.24	0.83	1.00	1.50	-30.94	0.57	2.73
97	302895.21	9032367.92	2.71	0.67	1.00	1.65	-29.57	0.61	2.53
98	302896.21	9032367.92	2.12	1.47	1.00	1.46	-28.34	0.48	5.03
99	302897.21	9032367.92	1.59	1.67	1.00	1.26	-29.57	0.39	42.30
100	302898.21	9032367.92	1.59	1.67	1.00	1.26	-29.98	0.32	188.57
...	...	...	...	...	...	...	...	...	...
388797	303888.21	9033979.92	0.19	1795.33	0.96	0.43	-49.01	0.34	347.50
388798	303889.21	9033979.92	0.30	1905.78	0.96	0.54	-56.67	0.44	240.00
388799	303890.21	9033979.92	1.33	261.33	1.00	1.15	-56.67	1.33	341.33

Lanjutan Lampiran 10

No	COR 220	ENER 220	DB 220	ASM 270	CON 270	COR 270	ENER 270	DB 270	ASM 350
1	1.00	2.00	-27.44	4.00	0.00	1.00	2.00	-55.43	4.00
2	1.00	2.00	-30.54	4.00	0.00	1.00	2.00	-55.43	4.00
3	1.00	2.00	-31.18	4.00	0.00	1.00	2.00	-55.43	4.00
4	1.00	1.64	-32.78	4.00	0.00	1.00	2.00	-28.05	2.71
5	1.00	1.57	-31.18	4.00	0.00	1.00	2.00	-32.83	1.59
6	1.00	1.61	-31.82	4.00	0.00	1.00	2.00	-35.27	0.61
7	1.00	1.77	-32.78	4.00	0.00	1.00	2.00	-35.62	0.55
8	1.00	1.38	-35.99	4.00	0.00	1.00	2.00	-34.69	0.51
9	1.00	1.31	-36.31	4.00	0.00	1.00	2.00	-30.50	0.53
10	1.00	1.35	-36.31	4.00	0.00	1.00	2.00	-33.06	0.39
11	1.00	1.68	-35.03	3.13	0.95	1.00	1.77	-35.62	0.43

No	COR 220	ENER 220	DB 220	ASM 270	CON 270	COR 270	ENER 270	DB 270	ASM 350
12	1.00	1.48	-34.07	2.39	1.06	1.00	1.55	-33.06	0.59
13	1.00	1.37	-32.46	1.69	1.92	1.00	1.30	-36.32	0.87
14	1.00	1.14	-36.31	1.82	1.48	1.00	1.35	-33.76	1.03
15	1.00	1.19	-33.42	1.57	1.96	1.00	1.25	-32.83	1.27
16	1.00	1.10	-33.42	1.60	1.63	1.00	1.27	-33.41	1.63
17	1.00	1.01	-32.78	1.29	2.33	1.00	1.14	-33.99	1.71
18	1.00	0.96	-33.42	1.44	1.67	1.00	1.20	-36.90	1.30
19	1.00	0.94	-34.39	1.58	1.53	1.00	1.26	-35.04	1.25
20	1.00	0.94	-36.31	1.88	0.60	1.00	1.37	-37.60	1.63
21	1.00	1.03	-35.99	2.35	0.43	1.00	1.53	-38.19	2.14
22	1.00	1.22	-36.31	1.98	0.63	1.00	1.41	-38.19	1.81
23	1.00	1.37	-36.31	1.68	0.80	1.00	1.30	-35.27	1.81
24	1.00	1.36	-36.63	1.35	1.03	1.00	1.16	-35.27	2.04
25	1.00	1.31	-37.59	1.74	0.77	1.00	1.32	-36.56	2.82
26	1.00	1.35	-35.35	1.83	0.67	1.00	1.35	-36.32	2.50
27	1.00	1.61	-38.55	1.47	1.00	1.00	1.21	-36.32	2.15
28	1.00	1.55	-38.55	0.75	1.93	1.00	0.86	-37.60	1.81
29	1.00	1.47	-37.27	0.54	3.60	1.00	0.74	-35.97	2.08
30	1.00	1.28	-37.27	0.46	5.83	1.00	0.67	-37.25	1.67
31	1.00	1.41	-34.39	0.46	7.20	1.00	0.68	-38.19	1.54
32	1.00	1.37	-34.71	0.32	6.87	1.00	0.57	-39.12	1.35
33	1.00	1.43	-35.67	0.29	6.00	1.00	0.53	-39.82	1.62
34	1.00	1.31	-35.99	0.31	5.53	1.00	0.56	-41.68	1.70
35	1.00	1.43	-34.07	0.33	6.40	1.00	0.58	-40.40	1.31
36	1.00	1.37	-34.71	0.36	6.37	1.00	0.60	-36.90	0.89
37	1.00	1.46	-35.35	0.37	5.23	1.00	0.61	-36.56	0.82
38	1.00	1.27	-35.35	0.46	3.30	1.00	0.68	-36.90	1.01
39	1.00	1.21	-35.67	0.50	2.70	1.00	0.71	-36.56	0.88
40	1.00	1.00	-36.95	0.52	2.37	1.00	0.72	-37.25	0.72
41	1.00	0.98	-36.95	0.54	2.53	1.00	0.74	-36.32	0.71
42	1.00	0.94	-39.19	0.55	2.37	1.00	0.74	-34.69	0.84
43	1.00	0.95	-40.05	0.54	2.27	1.00	0.74	-33.06	0.93
44	1.00	0.87	-37.59	0.62	1.83	1.00	0.79	-33.41	0.75
45	1.00	0.80	-36.63	0.62	1.87	1.00	0.79	-33.99	0.80
46	1.00	0.75	-35.35	0.73	1.43	1.00	0.85	-35.62	0.82
47	1.00	0.74	-36.95	0.83	1.37	1.00	0.91	-36.56	0.85
48	1.00	0.69	-37.91	1.05	1.00	1.00	1.02	-40.05	0.71
49	1.00	0.64	-41.01	0.94	1.30	1.00	0.97	-37.60	0.71
50	1.00	0.68	-43.90	0.91	1.30	1.00	0.96	-36.56	0.78
51	1.00	0.66	-41.97	1.01	1.43	1.00	1.00	-36.56	0.82

<b>No</b>	<b>COR 220</b>	<b>ENER 220</b>	<b>DB 220</b>	<b>ASM 270</b>	<b>CON 270</b>	<b>COR 270</b>	<b>ENER 270</b>	<b>DB 270</b>	<b>ASM 350</b>
52	1.00	0.66	-40.37	1.27	1.00	1.00	1.13	-35.97	0.75
53	1.00	0.74	-38.23	1.23	0.93	1.00	1.11	-34.34	0.64
54	1.00	0.82	-35.99	1.27	0.83	1.00	1.13	-33.99	0.64
55	1.00	0.91	-37.27	1.50	0.70	1.00	1.22	-34.69	0.65
56	1.00	0.91	-36.95	1.52	1.30	1.00	1.23	-35.27	0.73
57	1.00	0.88	-35.67	1.09	1.87	1.00	1.04	-35.97	0.89
58	1.00	0.89	-37.91	1.01	1.93	1.00	1.01	-36.56	0.84
59	1.00	0.84	-38.23	0.95	1.57	1.00	0.98	-36.90	0.88
60	1.00	0.86	-38.55	1.11	1.77	1.00	1.05	-37.60	1.12
61	1.00	0.87	-35.35	0.91	2.37	1.00	0.96	-35.62	1.54
62	1.00	0.98	-35.03	0.80	2.93	1.00	0.89	-35.97	1.58
63	1.00	0.92	-34.71	0.75	2.83	1.00	0.86	-36.56	1.15
64	1.00	1.01	-34.71	0.77	2.40	1.00	0.88	-37.60	1.04
65	1.00	1.04	-35.99	0.78	2.00	1.00	0.88	-38.54	1.02
66	1.00	1.13	-37.27	0.74	1.93	1.00	0.86	-37.60	1.08
67	1.00	0.99	-37.27	0.81	2.20	1.00	0.90	-37.60	1.05
68	1.00	0.90	-37.27	0.84	2.33	1.00	0.92	-37.84	1.17
69	1.00	0.85	-37.91	1.12	2.03	1.00	1.06	-39.82	1.15
70	1.00	0.93	-38.55	1.24	1.67	1.00	1.11	-35.27	1.32
71	1.00	1.00	-38.23	1.59	1.20	1.00	1.26	-33.06	1.32
72	1.00	1.04	-36.63	2.16	0.97	1.00	1.47	-33.41	1.99
73	1.00	1.10	-34.71	2.83	0.33	1.00	1.68	-35.97	2.70
74	1.00	1.35	-34.71	1.99	0.87	1.00	1.41	-35.62	3.37
75	1.00	1.71	-34.39	0.98	2.07	1.00	0.99	-34.69	2.81
76	1.00	1.90	-35.35	0.75	2.73	1.00	0.86	-36.32	2.61
77	1.00	1.90	-35.67	0.83	2.50	1.00	0.91	-37.25	2.34
78	1.00	1.81	-34.71	1.11	1.93	1.00	1.06	-36.56	3.15
79	1.00	1.77	-34.71	0.88	2.67	1.00	0.94	-35.97	3.26
80	1.00	1.62	-35.03	0.83	3.07	1.00	0.91	-34.34	3.49
81	1.00	1.68	-36.63	0.90	2.67	1.00	0.95	-35.04	2.71
82	1.00	1.50	-37.27	1.60	1.23	1.00	1.27	-34.69	1.94
83	1.00	1.44	-36.95	2.26	0.47	1.00	1.50	-35.04	1.47
84	1.00	1.08	-32.78	3.38	0.17	1.00	1.84	-33.99	1.47
85	1.00	1.10	-35.03	2.93	0.27	1.00	1.71	-33.99	1.61
86	1.00	0.93	-34.39	2.72	0.33	1.00	1.65	-34.69	1.35
87	1.00	0.89	-38.55	2.72	0.33	1.00	1.65	-35.97	1.34
88	1.00	0.74	-35.03	3.49	0.13	1.00	1.87	-35.97	1.40
89	1.00	0.79	-37.27	3.37	0.13	1.00	1.84	-32.48	2.11
90	1.00	0.75	-39.73	2.45	0.43	1.00	1.56	-33.06	2.70
91	1.00	0.79	-35.03	1.55	2.93	1.00	1.24	-33.41	3.49
92	1.00	0.78	-37.27	1.22	4.73	1.00	1.11	-34.34	3.14

No	COR 220	ENER 220	DB 220	ASM 270	CON 270	COR 270	ENER 270	DB 270	ASM 350
93	1.00	0.85	-38.23	0.95	7.13	1.00	0.98	-33.76	2.71
94	1.00	0.86	-37.91	0.89	6.93	1.00	0.94	-32.83	2.71
95	1.00	0.82	-37.91	0.73	7.30	1.00	0.86	-34.34	3.37
96	1.00	0.75	-36.95	0.80	6.87	1.00	0.90	-37.60	3.37
97	1.00	0.78	-36.31	1.00	7.60	1.00	1.00	-35.62	3.02
98	1.00	0.69	-34.71	1.27	6.90	1.00	1.13	-33.99	2.71
99	1.00	0.62	-35.99	1.44	5.47	1.00	1.20	-37.25	2.83
100	1.00	0.56	-36.31	1.61	3.27	1.00	1.27	-33.41	2.59
...	...	...	...	...	...	...	...	...	...
388797	0.99	0.58	-50.63	0.16	1881.25	0.96	0.41	-49.60	0.17
388798	1.00	0.67	-54.69	0.19	2217.55	0.95	0.43	-49.95	0.17
388799	0.99	1.15	-54.69	0.19	3282.92	0.93	0.44	-55.43	0.35

Lanjutan Lampiran 10

No	CON 350	COR 350	ENER 350	DB 350	ASM 400	CON 400	COR 400	ENER 400	DB 400
1	0.00	1.00	2.00	-31.86	4.00	0.00	1.00	2.00	-58.22
2	0.00	1.00	2.00	-31.86	4.00	0.00	1.00	2.00	-29.85
3	0.00	1.00	2.00	-34.96	1.86	7.00	1.00	1.36	-31.16
4	1.50	1.00	1.65	-33.48	1.01	7.60	1.00	1.01	-30.50
5	6.00	1.00	1.26	-29.58	0.68	8.60	1.00	0.82	-35.06
6	84.38	1.00	0.78	-31.46	0.65	9.65	1.00	0.81	-46.12
7	74.46	1.00	0.74	-39.79	0.48	11.29	1.00	0.69	-30.24
8	63.94	1.00	0.72	-44.90	0.28	553.31	0.99	0.52	-46.12
9	31.08	1.00	0.73	-47.05	0.22	465.95	0.99	0.47	-31.55
10	65.67	1.00	0.63	-41.00	0.36	412.79	0.99	0.60	-43.65
11	59.96	1.00	0.66	-38.18	0.39	187.25	1.00	0.62	-41.70
12	54.94	1.00	0.77	-35.63	0.40	516.68	0.99	0.63	-40.78
13	6.89	1.00	0.93	-38.58	0.40	504.78	0.99	0.63	-39.09
14	5.23	1.00	1.01	-40.73	1.13	474.22	0.99	1.07	-37.53
15	4.00	1.00	1.13	-39.53	1.42	3.05	1.00	1.19	-37.92
16	3.73	1.00	1.28	-37.24	1.61	2.47	1.00	1.27	-37.92
17	3.33	1.00	1.31	-32.40	1.33	3.67	1.00	1.16	-36.62
18	4.50	1.00	1.14	-30.52	2.05	3.00	1.00	1.43	-32.46
19	4.33	1.00	1.12	-30.52	2.02	2.37	1.00	1.42	-30.90
20	3.17	1.00	1.28	-34.02	2.00	0.80	1.00	1.41	-32.72
21	1.10	1.00	1.46	-35.36	1.61	1.00	1.00	1.27	-34.41
22	1.77	1.00	1.35	-36.30	1.54	1.23	1.00	1.24	-37.27
23	1.77	1.00	1.35	-34.42	1.22	1.50	1.00	1.10	-35.32

No	CON 350	COR 350	ENER 350	DB 350	ASM 400	CON 400	COR 400	ENER 400	DB 400
24	1.67	1.00	1.43	-34.02	0.88	2.20	1.00	0.94	-35.58
25	0.43	1.00	1.68	-34.42	0.72	2.57	1.00	0.85	-35.58
26	0.50	1.00	1.58	-34.42	0.66	2.57	1.00	0.81	-35.97
27	0.63	1.00	1.47	-34.42	0.70	2.20	1.00	0.83	-35.32
28	1.07	1.00	1.35	-34.42	0.77	2.13	1.00	0.88	-36.23
29	0.87	1.00	1.44	-37.24	1.09	2.10	1.00	1.04	-37.92
30	1.07	1.00	1.29	-38.85	1.00	2.63	1.00	1.00	-37.92
31	1.13	1.00	1.24	-37.51	0.97	2.03	1.00	0.99	-36.62
32	1.43	1.00	1.16	-34.96	1.25	1.37	1.00	1.12	-35.58
33	1.27	1.00	1.27	-34.96	1.77	0.83	1.00	1.33	-36.23
34	1.10	1.00	1.30	-34.96	1.63	1.20	1.00	1.28	-36.88
35	1.63	1.00	1.14	-34.69	0.91	2.23	1.00	0.96	-36.88
36	2.20	1.00	0.94	-37.24	0.60	3.37	1.00	0.77	-36.88
37	2.30	1.00	0.90	-38.85	0.52	3.40	1.00	0.72	-37.27
38	1.70	1.00	1.01	-38.85	0.48	2.97	1.00	0.70	-38.57
39	1.97	1.00	0.94	-38.58	0.48	2.50	1.00	0.69	-38.57
40	2.07	1.00	0.85	-35.63	0.52	2.73	1.00	0.72	-37.27
41	2.03	1.00	0.84	-35.63	0.52	3.10	1.00	0.72	-36.88
42	1.57	1.00	0.92	-36.57	0.53	2.90	1.00	0.73	-38.18
43	1.77	1.00	0.96	-36.97	0.54	2.60	1.00	0.73	-39.74
44	2.20	1.00	0.86	-36.97	0.62	2.23	1.00	0.79	-41.04
45	2.10	1.00	0.90	-36.57	0.74	2.27	1.00	0.86	-38.57
46	2.03	1.00	0.91	-37.51	0.83	2.07	1.00	0.91	-36.88
47	1.83	1.00	0.92	-37.91	1.22	1.73	1.00	1.11	-37.92
48	1.97	1.00	0.84	-38.58	1.54	1.27	1.00	1.24	-36.88
49	1.87	1.00	0.84	-37.51	2.34	0.80	1.00	1.53	-38.57
50	1.67	1.00	0.88	-38.18	1.77	0.97	1.00	1.33	-37.53
51	1.83	1.00	0.91	-37.24	1.30	1.47	1.00	1.14	-36.62
52	2.37	1.00	0.87	-36.30	1.08	1.60	1.00	1.04	-37.27
53	2.67	1.00	0.80	-37.51	1.19	1.33	1.00	1.09	-37.53
54	2.53	1.00	0.80	-39.53	1.30	1.27	1.00	1.14	-39.74
55	2.60	1.00	0.81	-39.12	1.06	1.53	1.00	1.03	-38.83
56	3.20	1.00	0.86	-38.85	1.08	1.43	1.00	1.04	-38.83
57	3.63	1.00	0.94	-37.91	1.10	1.03	1.00	1.05	-37.27
58	3.90	1.00	0.92	-34.96	1.38	0.63	1.00	1.17	-38.18
59	2.80	1.00	0.94	-34.96	1.25	0.70	1.00	1.12	-37.27
60	2.40	1.00	1.06	-34.69	1.35	0.77	1.00	1.16	-38.57
61	1.67	1.00	1.24	-36.03	1.35	0.77	1.00	1.16	-38.57
62	2.17	1.00	1.25	-36.57	1.43	0.83	1.00	1.20	-38.18
63	2.13	1.00	1.07	-36.97	1.34	0.87	1.00	1.16	-37.53
64	2.00	1.00	1.02	-36.97	1.35	0.83	1.00	1.16	-36.62

<b>No</b>	<b>CON 350</b>	<b>COR 350</b>	<b>ENER 350</b>	<b>DB 350</b>	<b>ASM 400</b>	<b>CON 400</b>	<b>COR 400</b>	<b>ENER 400</b>	<b>DB 400</b>
65	1.23	1.00	1.01	-37.51	1.80	0.63	1.00	1.34	-37.53
66	1.07	1.00	1.04	-39.53	2.25	0.47	1.00	1.50	-37.53
67	1.17	1.00	1.02	-37.91	2.20	0.53	1.00	1.48	-40.13
68	1.07	1.00	1.08	-37.24	1.75	0.80	1.00	1.32	-37.53
69	1.17	1.00	1.07	-36.97	1.89	0.77	1.00	1.38	-37.53
70	1.00	1.00	1.15	-36.57	2.35	0.57	1.00	1.53	-35.97
71	1.17	1.00	1.15	-36.30	2.93	0.30	1.00	1.71	-37.27
72	0.87	1.00	1.41	-36.03	2.83	0.33	1.00	1.68	-35.97
73	0.63	1.00	1.64	-35.63	2.64	0.40	1.00	1.62	-36.88
74	0.27	1.00	1.84	-36.03	2.64	0.40	1.00	1.62	-35.58
75	0.63	1.00	1.68	-36.03	2.64	0.40	1.00	1.62	-34.67
76	0.70	1.00	1.62	-36.03	2.73	0.37	1.00	1.65	-35.97
77	0.80	1.00	1.53	-34.96	2.36	0.47	1.00	1.54	-34.67
78	0.23	1.00	1.77	-34.02	2.18	0.50	1.00	1.48	-35.32
79	0.20	1.00	1.81	-33.75	1.96	0.60	1.00	1.40	-35.97
80	0.13	1.00	1.87	-33.48	1.91	0.67	1.00	1.38	-35.97
81	0.57	1.00	1.65	-32.81	1.69	1.00	1.00	1.30	-35.06
82	1.83	1.00	1.39	-32.40	1.39	1.20	1.00	1.18	-35.06
83	3.13	1.00	1.21	-34.42	1.54	1.07	1.00	1.24	-35.32
84	3.33	1.00	1.21	-35.63	1.37	1.43	1.00	1.17	-35.32
85	2.63	1.00	1.27	-36.30	1.32	1.57	1.00	1.15	-38.18
86	2.77	1.00	1.16	-34.96	1.05	2.00	1.00	1.02	-35.58
87	3.30	1.00	1.16	-33.75	1.25	1.63	1.00	1.12	-37.27
88	3.23	1.00	1.18	-34.02	1.52	1.60	1.00	1.23	-35.32
89	1.93	1.00	1.45	-35.36	1.88	1.20	1.00	1.37	-35.97
90	0.63	1.00	1.64	-34.42	1.80	1.57	1.00	1.34	-35.32
91	0.13	1.00	1.87	-33.07	1.75	1.20	1.00	1.32	-33.76
92	0.53	1.00	1.77	-33.75	1.73	1.83	1.00	1.31	-33.37
93	0.67	1.00	1.65	-34.42	1.79	1.57	1.00	1.34	-34.41
94	0.67	1.00	1.65	-34.42	2.13	1.53	1.00	1.46	-33.76
95	0.27	1.00	1.84	-38.58	2.21	1.30	1.00	1.49	-37.53
96	0.47	1.00	1.84	-37.51	2.15	1.77	1.00	1.47	-32.72
97	0.57	1.00	1.74	-36.30	2.22	1.43	1.00	1.49	-33.37
98	0.67	1.00	1.65	-35.36	2.81	0.93	1.00	1.68	-33.37
99	0.33	1.00	1.68	-32.81	3.62	0.10	1.00	1.90	-34.67
100	1.50	1.00	1.61	-33.48	4.00	0.00	1.00	2.00	-34.67
...	...	...	...	...	...	...	...	...	...
388797	751.70	0.98	0.41	-53.10	0.13	2202.59	0.93	0.36	-54.45
388798	917.80	0.98	0.41	-52.56	0.18	3643.00	0.88	0.42	-53.80
388799	822.18	0.98	0.59	-55.11	0.26	5563.27	0.82	0.51	-58.22

*“Halaman ini sengaja dikosongkan”*

## BIODATA PENULIS



Penulis dilahirkan di Kabupaten Magetan, Jawa Timur pada tanggal 8 Februari 2001 dan bertempat tinggal di Desa Mojopurno, Kecamatan Ngariboyo, Kabupaten Magetan. Penulis telah menempuh Pendidikan formal di SDN Mojopurno 2 (2007-2013), SMPN 1 Magetan (2013-2016), dan SMAN 1 Magetan (2016-2019). Setelah lulus dari SMA tahun 2019, penulis mengikuti SNMPTN dan diterima di S-1 Departemen Teknik Geomatika Fakultas Teknik Sipil, Perencanaan, dan Kebumihan Institut Teknologi Sepuluh Nopember Surabaya dan terdaftar dengan NRP 03311940000014.

Selama berkuliah penulis aktif mengikuti berbagai pelatihan dan organisasi. Penulis pernah mengikuti LKMM Pra-TD, LKMM TD, *Basic Media School*, dan menjadi pemateri kegiatan himpunan.

Dalam berorganisasi penulis pernah diamanahi sebagai Sekretaris II HIMAGE-ITS pada tahun 2020/2021 dan Sekretaris I HIMAGE-ITS pada tahun 2021/2021. Selain itu, penulis pernah melaksanakan magang di PT Gesosolution Pratama Nusantara selama 1 tahun serta kerja praktek di tempat yang sama selama 3 bulan. Dalam menyelesaikan syarat studi S-1, penulis mengambil bidang hidrografi dengan judul “Penerapan Algoritma Dense Neural Network untuk Klasifikasi Jenis Sedimen Dasar Laut Menggunakan Data Mosaik dan Pola Fitur Backscatter”.