

**IMPLEMENTASI JARINGAN *MULTI-LAYER*
PERCEPTRON UNTUK KLASIFIKASI GANGGUAN
SALURAN TRANSMISI TEGANGAN TINGGI
BERBASIS ANALISIS SINYAL DIGITAL**

LAPORAN TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana S1 Pada
Prodi Teknik Elektro Universitas Islam Sultan Agung



Oleh :
IRFAN KHOIRUDIN
30602300143

**PROGRAM STUDI S1 TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG
2025**

IMPLEMENTATION OF MULTI-LAYER PERCEPTRON NETWORK FOR HIGH VOLTAGE TRANSMISSION LINE FAULT CLASSIFICATION BASED ON DIGITAL SIGNAL ANALYSIS

*Suggested As One of the Requirements to Get a Bachelor's Degree in a Study Program
Electrical Engineering at Universitas Islam Sultan Agung.*



**PROGRAM STUDI S1 TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG
2025**

LEMBAR PENGESAHAN PEMBIMBING

LEMBAR PENGESAHAN PEMBIMBING

Laporan Tugas Akhir dengan judul "**IMPLEMENTASI JARINGAN MULTI-LAYER PERCEPTRON UNTUK KLASIFIKASI GANGGUAN SALURAN TRANSMISI TEGANGAN TINGGI BERBASIS ANALISIS SINYAL DIGITAL**" ini disusun oleh:

Nama : Irfan Khoirudin
NIM : 30602300143
Program Studi : Teknik Elektro

Telah disahkan dan disetujui oleh dosen pembimbing pada:

Hari : Kamis
Tanggal : 27 November 2025

Pembimbing I


Rehim
Prof. Dr. Ir. Sri Arttini Dwi Prasetyowati, M.Si.
NIDN : 0620026501

Mengetahui,

Ka. Program Studi Teknik Elektro


28/11/2025
Munai Ismail, S.T., M.T.
NIDN : 0613127302

LEMBAR PENGESAHAN PENGUJI

LEMBAR PENGESAHAN PENGUJI

Laporan Tugas Akhir dengan judul "**IMPLEMENTASI JARINGAN MULTI-LAYER PERCEPTRON UNTUK KLASIFIKASI GANGGUAN SALURAN TRANSMISI TEGANGAN TINGGI BERBASIS ANALISIS SINYAL DIGITAL**" ini telah dipertahankan di depan Penguji sidang Tugas Akhir pada:

Hari

: Jumat

Tanggal

: 21 November 2025

Tim Penguji

Tanda Tangan

Munaf Ismail, S.T., M.T.
NIDN : 0613127302

Ketua



Dr. Bustanul Arifin, S.T., M.T.
NIDN : 0614117701
Penguji I



Prof. Dr. Ir. Sri Arttini Dwi
Prasetyowati, M.Si.
NIDN : 0620026501
Penguji II



PERNYATAAN KEASLIAN

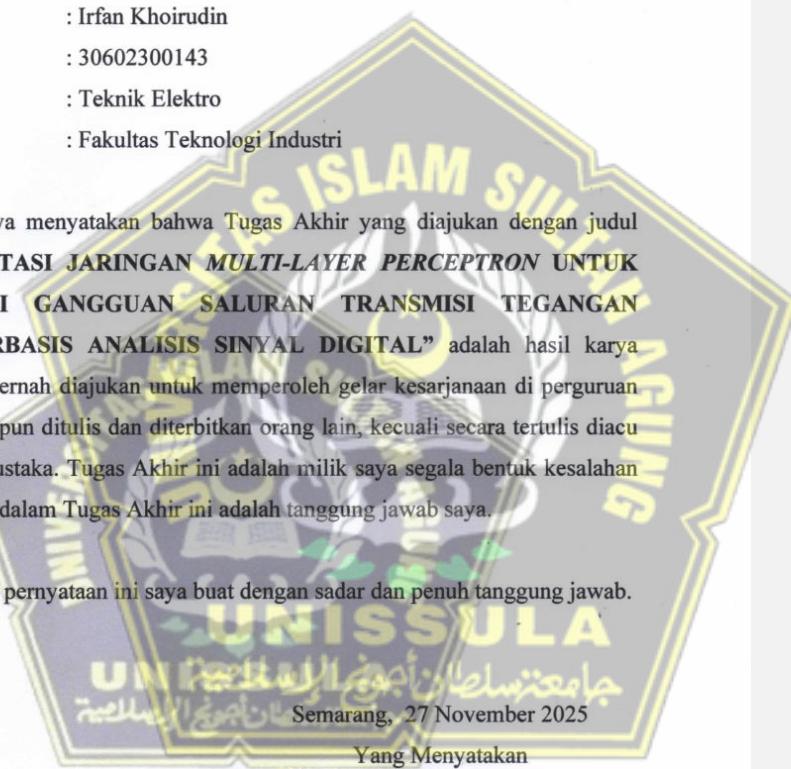
SURAT PERNYATAAN

Yang bertanda tangan dibawah ini:

Nama : Irfan Khoirudin
NIM : 30602300143
Jurusan : Teknik Elektro
Fakultas : Fakultas Teknologi Industri

Dengan ini saya menyatakan bahwa Tugas Akhir yang diajukan dengan judul **“IMPLEMENTASI JARINGAN MULTI-LAYER PERCEPTRON UNTUK KLASIFIKASI GANGGUAN SALURAN TRANSMISI TEGANGAN TINGGI BERBASIS ANALISIS SINYAL DIGITAL”** adalah hasil karya sendiri, tidak pernah diajukan untuk memperoleh gelar kesarjanaan di perguruan tinggi lain maupun ditulis dan diterbitkan orang lain, kecuali secara tertulis diacu dalam daftar pustaka. Tugas Akhir ini adalah milik saya segala bentuk kesalahan dan kekeliruan dalam Tugas Akhir ini adalah tanggung jawab saya.

Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.



Yang Menyatakan



Irfan Khoirudin

PERNYATAAN PERSETUJUAN PUBLIKASI ILMIAH

PERNYATAAN PERSETUJUAN UNGGAH KARYA ILMIAH

Saya yang bertanda tangan di bawah ini :

Nama	:	Irfan Khoirudin
NIM	:	30602300143
Program Studi	:	Teknik Elektro
Fakultas	:	Fakultas Teknologi Industri

Dengan ini menyerahkan karya ilmiah berupa Tugas Akhir/Skripsi/Tesis/Disertasi* dengan judul :

IMPLEMENTASI JARINGAN MULTI-LAYER PERCEPTRON UNTUK KLASIFIKASI GANGGUAN SALURAN TRANSMISI TEGANGAN TINGGI BERBASIS ANALISIS SINYAL DIGITAL

dan menyetujuinya menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak Bebas Royalti Non-ekslusif untuk disimpan, dialihmediakan, dikelola dalam pangkalan data, dan dipublikasikannya di internet atau media lain untuk kepentingan akademis selama tetap mencantumkan nama penulis sebagai pemilik Hak Cipta.

Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan pihak Universitas Islam Sultan Agung.

UNISSULA
Semarang, 28 November 2025
Yang menyatakan,

(Irfan Khoirudin)

*Coret yang tidak perlu

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Dengan nama Allah yang Maha Pengasih lagi Maha Penyayang. Segala puji bagi Allah yang mulia dan pemilik ‘Arsy yang agung. Kami memohon kepada Nya agar diberi pertolongan di dunia dan akhirat, juga agar diberikan keberkahan dimanapun berada, serta agar kami dijadikan di antara hamba-hamba Nya yang apabila diberi ia bersyukur, apabila diuji ia bersabar, dan apabila berbuat dosa ia memohon ampun. Selawat dan salam senantiasa tercurah kepada Nabi Muhammad *shalallahu ‘alaihi wa sallam*, keluarga, sahabat, serta orang-orang yang mengikuti mereka hingga akhir zaman.

Penulis menyadari bahwa penyusunan tugas akhir ini tidak akan bisa diwujudkan jika bukan karena Allah kemudian bimbingan, dukungan, serta doa dari berbagai pihak. Oleh karenanya rasa syukur kepada Allah *Ta’ala* tidak luput untuk dipanjatkan serta ucapan terima kasih penulis haturkan kepada:

1. Orang tua yang telah menjadi dukungan penulis untuk segera menyelesaikan tugas akhir ini.
2. Istri dan putri tercinta yang senantiasa mendampingi dan mendukung penulis.
3. Ibu Dr. Ir. Novi Marlyana, ST., MT., IPU., ASEAN, Eng. selaku Dekan Fakultas Teknologi Industri, Universitas Islam Sultan Agung Semarang.
4. Bapak Munaf Ismail, ST., MT. selaku Ketua Prodi Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Sultan Agung Semarang.
5. Prof. Dr. Ir. Sri Arttini Dwi Prasetyowati, M.Si. selaku Dosen Pembimbing tugas akhir.
6. Seluruh Dosen jurusan Teknik Elektro, Universitas Islam Sultan Agung Semarang yang telah memberikan ilmu, wawasan, dan arahan selama masa perkuliahan. Serta seluruh Staff Teknik Elektro.
7. Serta seluruh rekan, kolega dan semua pihak yang telah memberikan bantuan dan kesempatan yang tidak dapat disebutkan satu per satu.

ABSTRAK

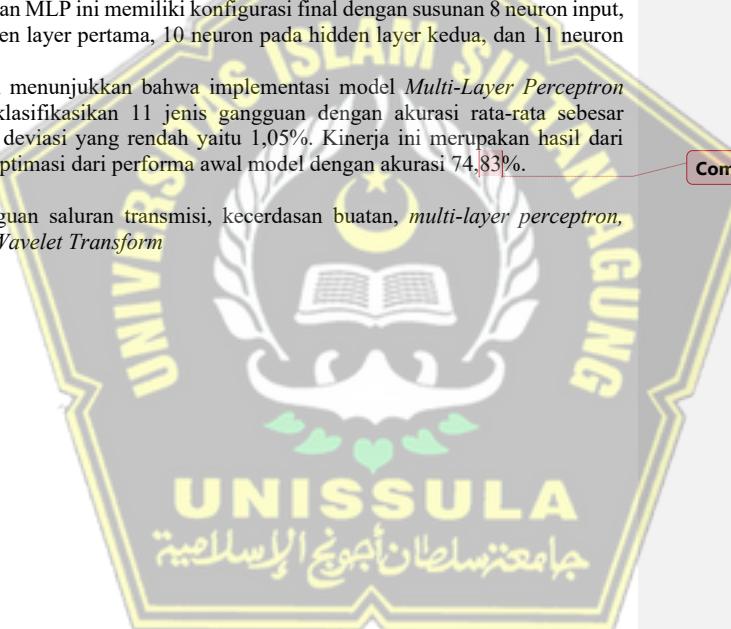
Saluran transmisi tegangan tinggi tidak lepas akan potensi terjadinya berbagai jenis gangguan. Proses mengenali karakteristik gangguan tersebut seringkali masih memerlukan interpretasi manual yang membutuhkan waktu lama. Guna meningkatkan efisiensi dalam proses pemulihan sistem serta mempercepat waktu yang dibutuhkan untuk mengetahui jenis gangguan yang terjadi, maka pada tugas akhir ini diusulkan implementasi dari jaringan *Multi-Layer Perceptron (MLP)*. Sistem kecerdasan buatan ini dirancang untuk membaca karakteristik arus, tegangan, dan karakteristik lainnya ketika gangguan terjadi sehingga dapat melakukan klasifikasi jenis gangguan secara efisien.

Prinsip dari penelitian ini diawali dengan membangkitkan 1188 sampel data untuk 11 jenis gangguan melalui simulasi *MATLAB*. Kemudian sinyal gangguan mentah diproses dan dilakukan ekstraksi fitur menggunakan pendekatan *Discrete Wavelet Transform (DWT)* untuk menghasilkan delapan fitur yang akan digunakan sebagai input *MLP*. Arsitektur dari jaringan *MLP* ini memiliki konfigurasi final dengan susunan 8 neuron input, 22 neuron pada hidden layer pertama, 10 neuron pada hidden layer kedua, dan 11 neuron output.

Hasil penelitian menunjukkan bahwa implementasi model *Multi-Layer Perceptron* final mampu mengklasifikasikan 11 jenis gangguan dengan akurasi rata-rata sebesar 96,18% dan standar deviasi yang rendah yaitu 1,05%. Kinerja ini merupakan hasil dari serangkaian proses optimasi dari performa awal model dengan akurasi 74,83%.

Commented [ik1]: Kata kunci

Kata Kunci: Gangguan saluran transmisi, kecerdasan buatan, *multi-layer perceptron*, *MATLAB*, *Discrete Wavelet Transform*



DAFTAR ISI

IMPLEMENTASI JARINGAN <i>MULTI-LAYER PERCEPTRON</i> UNTUK KLASIFIKASI GANGGUAN SALURAN TRANSMISI TEGANGAN TINGGI BERBASIS ANALISIS SINYAL DIGITAL.....	i
IMPLEMENTATION OF MULTI-LAYER PERCEPTRON NETWORK FOR HIGH VOLTAGE TRANSMISSION LINE FAULT CLASSIFICATION BASED ON DIGITAL SIGNAL ANALYSIS	ii
LEMBAR PENGESAHAN PEMBIMBING	iii
LEMBAR PENGESAHAN PENGUJI.....	iv
PERNYATAAN KEASLIAN.....	v
PERNYATAAN PERSETUJUAN PUBLIKASI ILMIAH	vi
KATA PENGANTAR.....	vii
ABSTRAK	viii
DAFTAR ISI.....	ix
DAFTAR TABEL	xiii
DAFTAR GAMBAR.....	xiv
BAB I PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Sistematika Penelitian	3
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI.....	5
2.1 Tinjauan Pustaka	5
2.2 Dasar Teori.....	7
2.2.1 Sistem Tenaga Listrik	7
1. Saluran Transmisi Tegangan Tinggi	8
2. Gangguan Pada Saluran Transmisi.....	9
3. DC Offset Transient.....	10

4.	Analisis Transien.....	11
5.	Model Saluran Transmisi Pendek.....	12
2.2.2	Analisis Sinyal Digital	13
2.2.3	Transformasi <i>Wavelet</i>	13
1.	<i>Discrete Wavelet Transform (DWT)</i> dan Dekomposisi Sinyal	13
2.	Koefisien Aproksimasi (A) dan Detail (D)	14
3.	Aplikasi <i>Discrete Wavelet Transform (DWT)</i> untuk Ekstraksi Fitur Gangguan.....	15
4.	Jaringan Saraf Tiruan	16
2.2.5	<i>Multi-Layer Perceptron</i>	16
1.	Arsitektur <i>Multi-Layer Perceptron</i>	17
2.	<i>Neuron</i> dan Fungsi Aktivasi.....	18
2.2.6	Pelatihan Jaringan Saraf Tiruan	20
1.	<i>Backpropagation</i>	20
2.	Epoch.....	20
3.	Overfitting dan Underfitting.....	21
4.	<i>Confusion Matrix</i> dan Evaluasi Model.....	22
2.2.7	<i>MATLAB</i>	
1.	<i>Simulink</i>	25
2.	<i>Deep Learning Toolbox</i>	25
3.	<i>Signal Processing Toolbox</i>	25
4.	Pemrograman <i>MATLAB</i>	25
BAB III METODOLOGI PENELITIAN		28
3.1	Model Penelitian	28
3.2	Alat dan Bahan	28
3.3	Prosedur Penelitian.....	30
3.3.1	Perancangan dan Simulasi Sistem di Simulink	30
1.	Blok <i>Simulink</i> yang digunakan.....	31
2.	Rangkaian dan Arsitektur Simulasi	36
3.3.1	Menjalankan Simulasi dan Generasi Dataset	38
1.	Parameter untuk Generasi Dataset	38
2.	Penjelasan Skrip Generasi Dataset	41

3.	Menjalankan Skrip	45
3.3.3	Ekstraksi Fitur dan Pembentukan Dataset.....	46
1.	Analisis Data Mentah Hasil Simulasi.....	46
2.	Ekstraksi Fitur Energi Menggunakan <i>Discrete Wavelet Transform (DWT)</i>	48
3.	Ekstraksi Fitur Tambahan Arus Urutan Nol (I0)....	49
4.	Pembentukan Dataset Final	51
3.3.4	Persiapan Data untuk Pelatihan Model	53
3.3.5	Pelatihan Model <i>Multi-Layer Perceptron</i>	54
1.	Pembagian Dataset (Latih, Validasi, dan Uji).....	54
2.	Konfigurasi Arsitektur Jaringan dan Parameter Pelatihan54	
3.	Proses Pelatihan dan Validasi Model	56
4.	Evaluasi Kinerja Model.....	59
BAB IV HASIL DAN ANALISIS	64	
4.1	Hasil Simulasi	64
4.1.1	Sinyal Kondisi Sistem Normal.....	64
4.1.2	Sinyal Gangguan Satu Fasa ke Tanah.....	65
4.1.3	Sinyal Gangguan Dua Fasa ke Tanah	69
4.1.4	Sinyal Gangguan Tiga Fasa ke Tanah (A-B-C-G)...	73
4.1.5	Sinyal Gangguan Fasa ke Fasa.....	74
4.2	Hasil Dataset dan Pemrosesan Data.....	79
4.2.1	Tujuh Fitur Energi Hasil Dekomposisi <i>DWT</i>	79
4.2.2	Satu Fitur Arus Tambahan Hasil dari Magnitudo Total I0	80
4.2.3	Dataset Fitur dan Dataset Label Final	81
4.2.4	Matriks Input (<i>X</i>) dan Target (<i>Y</i>) untuk Pelatihan....	83
4.3	Kinerja Model dan Optimasi	85
4.3.1	Model Dasar	85
4.3.2	Optimasi Data Set	88
4.3.3	Analisis Kebingungan pada Gangguan Simetris (Kelas 10 dan 11)	91
4.3.4	Rekapitulasi Optimasi Kinerja Model.....	96

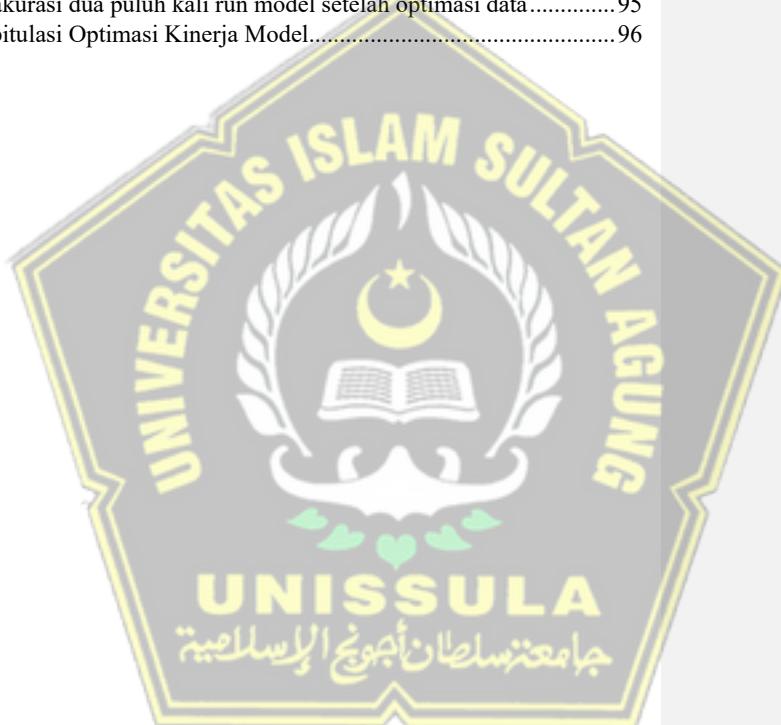
BAB V PENUTUP.....	98
5.1	Kesimpulan
5.2	Saran.....
DAFTAR PUSTAKA	100
LAMPIRAN.....	1

Commented [ik2]: Perbaikan daftar isi



DAFTAR TABEL

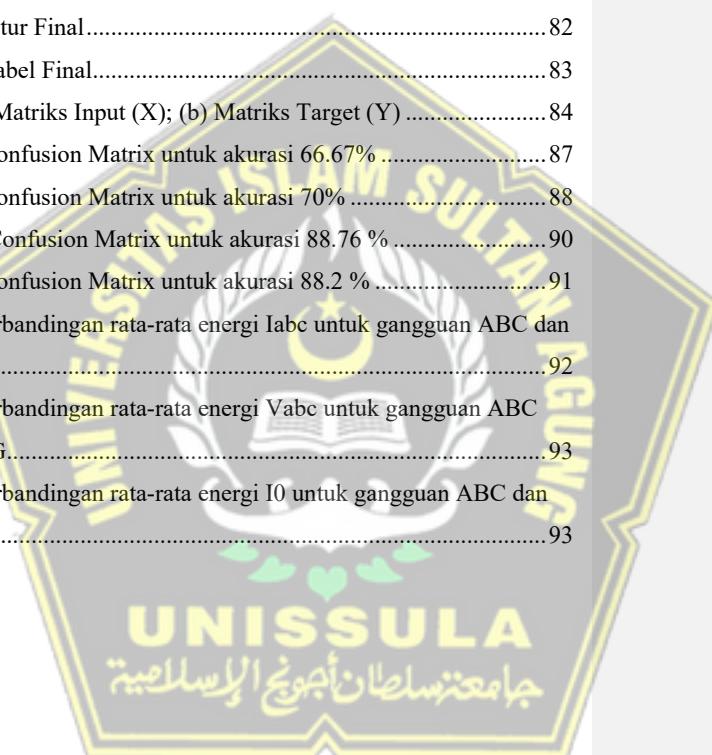
Tabel 4.1 Hasil akurasi dua puluh kali run model dasar	86
Tabel 4.2 Hasil akurasi dua puluh kali <i>run</i> model setelah optimasi data.....	89
Tabel 4.3 Percobaan arsitektur hidden layer	94
Tabel 4.4 Hasil akurasi dua puluh kali run model setelah optimasi data.....	95
Tabel 4. 5 Rekapitulasi Optimasi Kinerja Model.....	96



DAFTAR GAMBAR

Gambar 2.1 Model Saluran Transmisi yang digunakan oleh peneliti Rao, Kumar, dan Kesava Rao (2017)	6
Gambar 2.2 Dasar Konfigurasi Sistem Tenaga Listrik	8
Gambar 2.3 Menunjukkan contoh kondisi sistem ketika terjadi gangguan Fasa a ke tanah. Sistem yang sebelumnya seimbang kini terdapat arus yang mengalir ke tanah melalui impedansi gangguan Zf	10
Gambar 2.4 Model Saluran Transmisi Pendek	12
Gambar 2.5 Arsitektur Multi-layer Perceptron	17
Gambar 2.6 Model non-linear dari sebuah neuron dengan label k	19
Gambar 2.7 Contoh Confusion Matrix di mana blok diagonal berwarna biru adalah kelas yang berhasil diprediksi dengan benar	23
Gambar 2.8 Tampilan awal MATLAB R2025a	25
Gambar 3.1 Simulasi Saluran Transmisi Menggunakan Simulink29	
Gambar 3.2 Blok <i>Three-Phase Source</i>	31
Gambar 3.3 Blok Three-Phase Series RLC Load	32
Gambar 3.4 Blok <i>Three-Phase Pi Section Line</i>	32
Gambar 3.5 Blok Three-Phase Fault.....	33
Gambar 3.6 Blok Three-Phase V-I Measurement.....	34
Gambar 3.7 Blok Sequence Analyzer.....	34
Gambar 3.8 Blok To <i>Workspace</i>	35
Gambar 3.9 Blok Powergui.....	35
Gambar 3.10 Rangkaian Simulasi Transmisi di Simulink.....	36
Gambar 3.11 Nilai Tipikal Beban Puncak di Gardu Induk 150kV Batang	Er
rror! Bookmark not defined.	
Gambar 3.12 Skrip Generating Data sedang berjalan.....	45
Gambar 3.13 Skrip selesai dijalankan.....	45
Gambar 3.14 Item yang muncul di <i>Workspace</i>	46
Gambar 3.15 Contoh cuplikan data mentah hasil simulasi	47

Gambar 3.16 Pembacaan Arus I ₀ pada gangguan AG, BG, ABG.....	50
Gambar 3.17 Ilustrasi pembentukan dataset fitur	51
Gambar 3.18 Hasil dataset_fitur di <i>Workspace MATLAB</i>	51
Gambar 3.19 Hasil dataset_label di <i>Workspace MATLAB</i>	52
Gambar 3.20 Beberapa parameter, fungsi dan algoritma yang digunakan jaringan	56
Gambar 3.21 Proses pelatihan model sedang berjalan.....	57
Gambar 3.22 Hasil dari Jaringan yang telah dilatih dengan bias dan weight yang optimal	58
Gambar 3.23 Contoh Confusion Matrix	59
Gambar 4.1 Panduan representasi warna untuk gelombang tiga fasa ABC	64
Gambar 4.2 Sinyal Kondisi Sistem Normal (a) Sinyal Arus; (b) Sinyal Tegangan; (c) Sinyal Arus I ₀	64
Gambar 4.3 Sinyal Gangguan Fasa A-G (a) Sinyal Arus; (b) Sinyal Tegangan; (c) Sinyal Arus I ₀	65
Gambar 4.4 Sinyal Gangguan Fasa B-G (a) Sinyal Arus; (b) Sinyal Tegangan; (c) Sinyal Arus I ₀	67
Gambar 4.5 Sinyal Gangguan Fasa B-C (a) Sinyal Arus; (b) Sinyal Tegangan; (c) Sinyal Arus I ₀	68
Gambar 4.6 Sinyal Gangguan Fasa A-B-G (a) Sinyal Arus; (b) Sinyal Tegangan; (c) Sinyal Arus I ₀	69
Gambar 4.7 Sinyal Gangguan Fasa A-C-G (a) Sinyal Arus; (b) Sinyal Tegangan; (c) Sinyal Arus I ₀	70
Gambar 4.8 Sinyal Gangguan Fasa B-C-G (a) Sinyal Arus; (b) Sinyal Tegangan; (c) Sinyal Arus I ₀	72
Gambar 4.9 Sinyal Gangguan Fasa A-B-C-G (a) Sinyal Arus; (b) Sinyal Tegangan; (c) Sinyal Arus I ₀	73
Gambar 4.10 Sinyal Gangguan Fasa A-B (a) Sinyal Arus; (b) Sinyal Tegangan; (c) Sinyal Arus I ₀	74
Gambar 4.11 Sinyal Gangguan Fasa A-C (a) Sinyal Arus; (b) Sinyal Tegangan; (c) Sinyal Arus I ₀	75



BAB I

PENDAHULUAN

1.1 Latar belakang

Sistem transmisi yang andal merupakan pondasi vital bagi keberlangsungan penyaluran energi listrik ke seluruh pusat-pusat beban di Indonesia. Stabilitas Saluran Udara Tegangan Tinggi dan Ekstra Tinggi (SUTT/SUTET) menjadi salah satu infrastruktur utama yang memegang peranan krusial meskipun dalam operasionalnya tidak terlepas dari potensi terjadinya gangguan (*fault*). Gangguan ini merupakan suatu kondisi abnormal pada saluran yang dapat menghentikan aliran daya, merusak peralatan, bahkan membuka potensi bahaya pada area di sekitar jaringan transmisi.

Dalam menangani gangguan, sistem proteksi modern seperti relai digital telah digunakan secara luas dan mampu mengisolasi gangguan dengan cepat. Akan tetapi, proses analisis pasca-gangguan untuk menentukan secara pasti jenis dan karakteristik gangguan masih memerlukan interpretasi manual oleh *engineer proteksi* [1]. Analisis manual ini membutuhkan waktu dan keahlian khusus, sehingga berpotensi mengurangi efisiensi dalam proses pemulihan sistem.

Seiring dengan pesatnya perkembangan teknologi, pemanfaatan kecerdasan buatan (*Artificial Intelligence*) membuka peluang untuk meningkatkan efisiensi pada proses analisis tersebut. Tugas akhir ini mengusulkan implementasi Jaringan *Multi-Layer Perceptron* (MLP) yang merupakan salah satu bentuk jaringan saraf tiruan untuk melakukan klasifikasi gangguan secara otomatis berbasis analisis sinyal digital. Proses perancangan model sistem transmisi dan data gangguan untuk pelatihan model akan direalisasikan menggunakan simulasi *MATLAB* dan *toolbox Simulink*. Sistem klasifikasi otomatis ini diharapkan dapat mengidentifikasi jenis gangguan dengan cepat dan akurat, sehingga mampu menjadi alat pendukung dalam proses pemulihan sistem. Dengan demikian, efektivitas dan keandalan jaringan transmisi dapat ditingkatkan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan diatas, maka dapat diambil perumusan masalah sebagai berikut:

1. Bagaimana merancang model sistem saluran transmisi pendek untuk merepresentasikan karakteristik data sinyal gangguan yang umum terjadi pada saluran transmisi tegangan tinggi?
2. Bagaimana penentuan dan ekstraksi fitur-fitur sinyal yang paling relevan untuk digunakan sebagai input *Multi-Layer Perceptron* dalam klasifikasi gangguan transmisi?
3. Bagaimana merancang arsitektur *Multi-Layer Perceptron* yang memadai untuk melakukan klasifikasi jenis gangguan transmisi tegangan tinggi?
4. Bagaimana performa dan akurasi model *Multi-Layer Perceptron* yang telah dilatih dalam klasifikasi berbagai jenis gangguan?

1.3 Batasan Masalah

Batasan masalah dalam tugas akhir ini adalah sebagai berikut:

1. Pembahasan hanya berfokus pada penentuan jenis gangguan, tanpa menyertakan analisis lokasi maupun penyebab gangguan.
2. Sumber data utama untuk pelatihan dan pengujian model adalah hasil dari simulasi menggunakan *MATLAB* Simulink.
3. Software yang digunakan adalah *MATLAB* R2025a, Simulink, Deep Learning Toolbox, Signal Processing.
4. Model saluran transmisi yang disimulasikan adalah model saluran transmisi pendek (*short transmission line*) untuk level tegangan 150kV dengan parameter yang disesuaikan.
5. Jenis gangguan yang akan diklasifikasikan mencakup: gangguan satu fasa ke tanah (AG, BG, CG), antar fasa (AB, BC, CA), dua fasa ke tanah (ABG, BCG, CAG), dan tiga fasa (ABC, ABCG).
6. Untuk parameter yang tidak secara spesifik tersedia akan diestimasi menggunakan nilai tipikal dan pendekatan teoritis. Pendekatan ini

mengacu pada rentang nilai yang umum digunakan dalam literatur akademis dan penelitian relevan.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah:

1. Merancang dan membangun model simulasi sistem saluran transmisi 150 kV yang dapat membangkitkan data sinyal tegangan dan arus untuk berbagai skenario gangguan.
2. Mengimplementasikan arsitektur Jaringan *Multi-Layer Perceptron* (MLP) dan melatihnya menggunakan data fitur yang telah diekstraksi.
3. Mengevaluasi kinerja model MLP yang dihasilkan dalam mengklasifikasikan jenis-jenis gangguan.

1.5 Manfaat Penelitian

Manfaat penelitian ini adalah sebagai berikut:

1. Menambah pemahaman dan wawasan mengenai sistem saluran transmisi tegangan tinggi, sistem proteksi dan gangguan pada saluran transmisi, serta teknologi kecerdasan buatan khususnya *Multi-Layer Perceptron*.
2. Menyajikan pengetahuan mengenai pemodelan gangguan pada saluran transmisi tegangan tinggi menggunakan *MATLAB Simulink*.
3. Memberikan kontribusi dalam penerapan kecerdasan buatan dalam domain sistem tenaga listrik.
4. Menghasilkan sebuah prototipe sistem cerdas yang berpotensi untuk dikembangkan dan digunakan pada aplikasi industrial dan aktual sebagai instrumen pendukung bagi *engineer* untuk efisiensi analisis gangguan.

1.6 Sistematika Penelitian

Sistematika penulisan dari penelitian ini adalah sebagai berikut:

BAB I : PENDAHULUAN

Bab ini membahas tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat, metode penelitian dan penulisan sistematis penelitian yang akan dilakukan.

BAB II : TINJAUAN PUSTAKA DAN LANDASAN TEORI

Bab ini membahas secara umum tentang sistem transmisi tenaga listrik, gangguan transmisi, Neural Network serta dasar teori dan prinsip yang melandasi pembuatan Tugas Akhir ini.

BAB III : METODOLOGI PENELITIAN

Berisi tentang model penelitian, alat dan bahan yang digunakan berupa *software* maupun *hardware* sebagai media pendukung, menjelaskan prosedur penelitian, melakukan pemodelan serta simulasi untuk mendapatkan hasil penelitian yang dibutuhkan.

BAB IV : HASIL DAN ANALISIS

Bab ini membahas data simulasi gangguan, pelatihan model *Multi-Layer Perceptron* serta analisis evaluasi kinerja model dalam menentukan berbagai jenis gangguan.

BAB V : PENUTUP

Bab ini berisi kesimpulan dan saran dari hasil penelitian Tugas Akhir ini.



BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

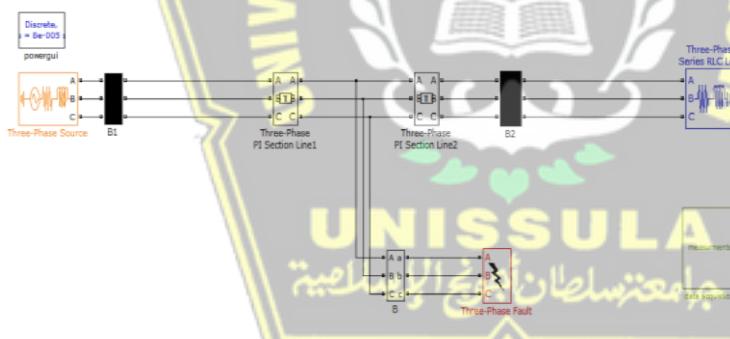
Menurut Raj Jain dalam *The Art of Computer Systems Performance Analysis*, terdapat tiga teknik utama untuk evaluasi kinerja sistem: pemodelan analitis, simulasi, dan pengukuran. Salah satu pertimbangan kunci yang penulis tekankan adalah pada tahap apa sistem tersebut berada (*life-cycle stage*). Dengan kata lain, pemilihan teknik sangat bergantung pada tujuan dan konteks dari evaluasi yang dilakukan. Ketika tujuannya adalah perancangan atau konsep baru yang belum tersedia secara fisik, maka pemodelan analitis dan simulasi menjadi satu-satunya teknik yang dapat dipilih. Lebih jauh lagi, simulasi seringkali mampu memberikan hasil yang lebih mendekati kondisi nyata dibandingkan pemodelan analitis karena simulasi dapat mengakomodasi lebih banyak detail dan memerlukan lebih sedikit asumsi penyederhanaan [2].

Prinsip yang dipaparkan Raj Jain menjadi sangat relevan dalam konteks penelitian ini, di mana tujuannya adalah untuk melatih sebuah model Jaringan *Multi-Layer Perceptron* untuk melakukan klasifikasi gangguan pada saluran transmisi tegangan tinggi. Keberhasilan model tersebut sangat bergantung pada ketersediaan dataset yang besar dan beragam, yang mencakup berbagai jenis dan lokasi gangguan. Membangkitkan skenario-skenario gangguan secara fisik pada sistem transmisi yang sesungguhnya bukan hanya tidak praktis dan memerlukan biaya tinggi, tetapi juga berbahaya. Oleh karena itu, penggunaan simulasi untuk membangkitkan data (*data generating*) menjadi metode yang valid dan efektif untuk menyediakan bahan penelitian yang dibutuhkan dalam tugas akhir ini.

Pemanfaatan Jaringan Saraf Tiruan (JST) untuk klasifikasi gangguan pada sistem transmisi merupakan bidang riset yang telah banyak dieksplorasi. Salah satu penelitian yang relevan adalah tesis Md. Chayon Ali (2023) yang meneliti metode deteksi dan klasifikasi gangguan menggunakan kombinasi *Wavelet Transform* dan Jaringan Saraf Tiruan (JST). Untuk keperluan penelitian ini, sebuah model sistem transmisi 230 kV disimulasikan secara menyeluruh menggunakan

MATLAB/Simulink. Proses pembangkitan data dilakukan dengan menciptakan berbagai skenario gangguan yang meliputi variasi pada jenis gangguan, lokasi gangguan di sepanjang saluran, dan nilai resistansi gangguan. Sinyal arus dari setiap simulasi kemudian dianalisis menggunakan *Discrete Wavelet Transform (DWT)* yang selanjutnya digunakan sebagai masukan untuk melatih model ANN [3].

Dalam penelitian yang dilakukan oleh Rao, Kumar, dan Kesava Rao (2017), diusulkan sebuah pendekatan untuk klasifikasi gangguan pada saluran transmisi menggunakan kombinasi antara *Wavelet Multi-Resolution Analysis* dan Jaringan Saraf Tiruan. Untuk mendapatkan data penelitian, mereka merancang dan mensimulasikan sebuah model sistem tenaga 220 kV sepanjang 300 km menggunakan *MATLAB Simulink*. Melalui simulasi ini dibangkitkan berbagai skenario gangguan untuk menghasilkan sinyal arus tiga fasa. Sinyal-sinyal ini kemudian dianalisis menggunakan *Discrete Wavelet Transform (DWT)*, yang selanjutnya digunakan sebagai masukan untuk melatih Jaringan Saraf Tiruan dalam mengklasifikasikan jenis gangguan [4].



Gambar 2.1 Model Saluran Transmisi yang digunakan oleh peneliti Rao, Kumar, dan Kesava Rao (2017)

Penelitian lain yang relevan dilakukan oleh Li et al. (2021) mengembangkan sebuah metode klasifikasi gangguan hubung singkat ke tanah pada jaringan transmisi menggunakan *Convolutional Neural Network (CNN)*. Selanjutnya suatu simulasi terukur dilakukan pada perangkat lunak PSCAD/EMTDC menggunakan model *grid new England 10 machine 39 nodes*. Proses pembangkitan data

dilakukan dengan menciptakan 5000 set sampel gangguan yang berbeda. Data mentah berupa bentuk gelombang arus, tegangan, sudut phasa dan daya beban kemudian diproses dengan metode *Z-score standardization*. Setelah data diproses kemudian data dibagi menjadi *training set* dan *test set* untuk jaringan CNN yang memiliki lima *convolutional layers*, tiga *max-pooling layers*, satu *concatenate layer*, satu *dropout layer*, satu *fully connected layer*, dan satu *Softmax classifier* [5].

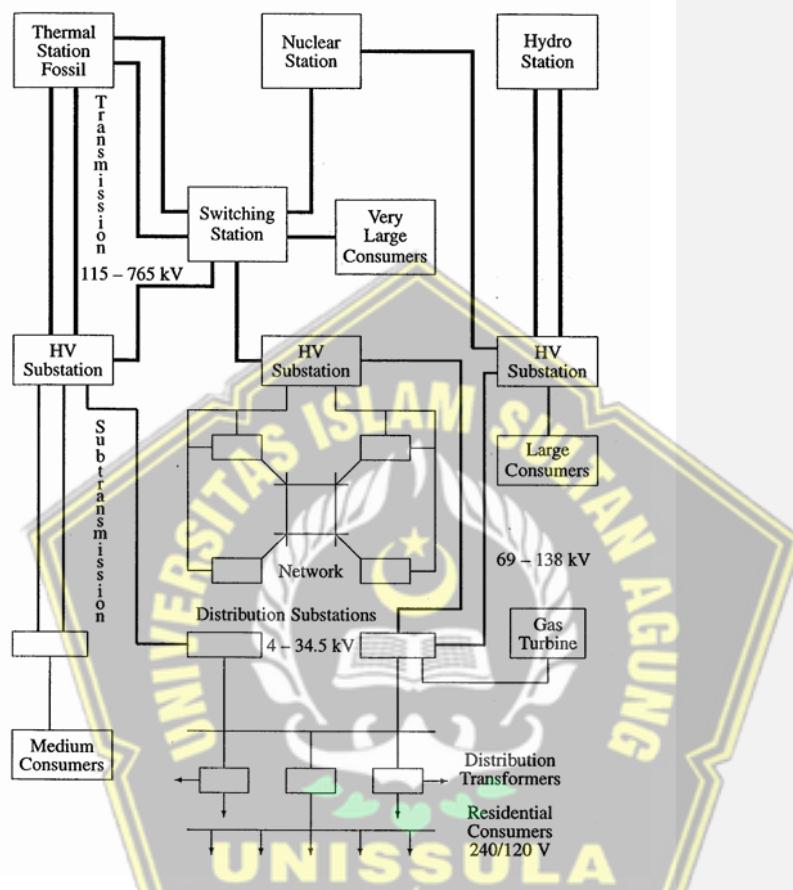
Selanjutnya, Tunio et al. (2024) menyajikan sebuah metode deteksi dan klasifikasi gangguan menggunakan *Discrete Wavelet Transform (DWT)* yang dikombinasikan dengan *Temporal Convolutional Neural Network (TCN)*. Penelitian ini juga bergantung pada simulasi yang dibangun menggunakan *MATLAB* untuk membuat model sistem transmisi 500 kV *Jamshoro-New Karachi* di *Sindh, Pakistan* [6].

Menindaklanjuti keberhasilan konsep tersebut, maka pada tugas akhir ini akan dilakukan penelitian serupa. Sebuah model *Multi-Layer Perceptron (MLP)* akan dirancang untuk menjadi sistem klasifikasi gangguan di mana *input* untuk model akan diperoleh dari hasil analisis sinyal digital yang dibangkitkan melalui simulasi komputasi.

2.2 Dasar Teori

2.2.1 Sistem Tenaga Listrik

Sistem tenaga listrik merupakan sebuah jaringan kompleks dan saling terhubung. Sistem ini terdiri dari pusat-pusat pembangkit, saluran transmisi, dan jaringan distribusi yang mengalirkan listrik sampai ke konsumen [1]. Keandalan dan stabilitas sistem tenaga listrik menjadi faktor krusial, karena setiap gangguan yang terjadi dapat menyebabkan pemadaman listrik yang merugikan secara ekonomi dan sosial.



Gambar 2.2 Dasar Konfigurasi Sistem Tenaga Listrik

Gambar 2.2 menunjukkan konfigurasi umum dari sistem tenaga listrik yang secara sederhana terdiri dari pembangkit kemudian disalurkan melalui transmisi dengan berbagai level tegangan sebelum berakhir di konsumen.

1. Saluran Transmisi Tegangan Tinggi

Saluran transmisi tegangan tinggi merupakan komponen vital dan menjadi tulang punggung dalam penyaluran energi listrik dari pusat pembangkit ke pusat beban. Di Indonesia, sistem transmisi ini secara umum diklasifikasikan berdasarkan level tegangannya, yaitu Saluran Udara Tegangan Tinggi (SUTT) yang beroperasi pada tegangan 70 kV-150 kV dan Saluran Udara Tegangan Ekstra Tinggi (SUTET)

pada tegangan 275 kV-500 kV. Pemilihan level tegangan yang tinggi ini ditujukan untuk mengurangi rugi-rugi daya selama transmisi jarak jauh. Hal ini didasarkan pada prinsip bahwa besarnya arus yang mempengaruhi rugi daya akan berbanding terbalik dengan besarnya tegangan. dinyatakan sebagai berikut:

$$P_{losses} = 3.I^2.R \quad (2.1)$$

dan

$$P = \sqrt{3}.V.I.\cos(\phi) \quad (2.2)$$

Di mana,

P_{losses} = rugi-rugi daya (Watt)

V = Tegangan (Volt)

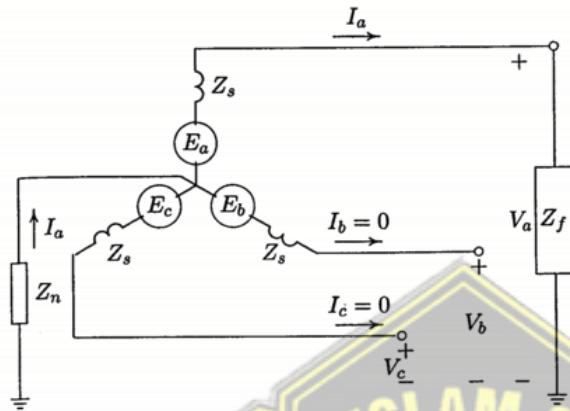
I = Arus (A)

R = Tahanan konduktor (Ω)

$\cos(\phi)$ = Faktor Daya

2. Gangguan Pada Saluran Transmisi

Meskipun dirancang untuk tingkat keandalan tinggi, saluran transmisi masih rentan terhadap berbagai jenis gangguan (*faults*). Gangguan adalah suatu kondisi abnormal yang mengganggu kestabilan sistem dan menyebabkan aliran arus besar ke peralatan listrik [7]. Secara umum, gangguan ini dapat diklasifikasikan menjadi dua kategori utama: gangguan simetris berupa hubung singkat tiga fasa dan gangguan tidak simetris [1]. Gangguan tidak simetris merupakan jenis yang paling sering terjadi, meliputi gangguan hubung singkat satu fasa ke tanah (*single line-to-ground*), hubung singkat antar fasa (*line-to-line*), dan hubung singkat dua fasa ke tanah (*double line-to-ground*). Oleh karena itu, deteksi dan klasifikasi jenis gangguan secara cepat dan akurat sangat penting untuk mengisolasi area yang terganggu dan mempercepat proses pemulihian sistem.



Gambar 2.3 Contoh kondisi sistem ketika gangguan fasa a ke ground

Gambar 2.3 Menunjukkan contoh kondisi sistem ketika terjadi gangguan Fasa a ke tanah. Sistem yang sebelumnya seimbang kini terdapat arus yang mengalir ke tanah melalui impedansi gangguan Z_f .

3. DC Offset Transient

Pada rangkaian daya yang didominasi induktansi (R–L), arus gangguan sesaat setelah *fault* terdiri dari dua komponen: komponen AC kondisi tunak (simetris) dan komponen DC offset (asimetri) yang memiliki *decay rate* proporsional terhadap R/L. Nilai awal komponen DC ini ditentukan oleh sudut penutupan/terjadinya gangguan (α) berdasarkan persamaan,

$$i_{dc} = -I\sqrt{2} \sin(\alpha - \varphi) e^{-\frac{R}{L}t} \quad (2.3)$$

dengan,

i_{dc} = Komponen arus DC offset (asimetris)

I = Arus RMS

α = Sudut terjadinya gangguan (fault inception angle)

φ = Sudut fasa arus relatif terhadap tegangan sumber

R = Resistansi ekuivalen dari sistem dari titik gangguan.

L = Induktansi ekuivalen dari sistem dari titik gangguan.

t = Waktu setelah terjadinya gangguan [8].

Secara spesifik, besarnya offset ini bervariasi dari nilai minimum 0 ketika gangguan terjadi pada sudut $\alpha = \theta$, hingga mencapai nilai maksimum $\sqrt{2} * i_{ac}$ yang setara dengan nilai puncak arus AC ketika gangguan terjadi pada sudut $\alpha = (\theta \pm \pi/2)$. Oleh karena itu, tergantung pada waktu terjadinya gangguan bentuk gelombang arus gangguan yang dihasilkan dapat bersifat simetris (dengan DC offset minimal) atau sangat asimetris (dengan DC offset maksimal). Fenomena ini secara fundamental mempengaruhi karakteristik sinyal transien secara keseluruhan [9].

4. Analisis Transien

Analisis transien adalah metode investigasi fenomena kelistrikan yang diambil ketika suatu sistem mengalami gangguan seperti hubung singkat. Ketika gangguan terjadi, perubahan drastis pada tegangan dan arus di titik gangguan akan membangkitkan gelombang elektromagnetik berfrekuensi tinggi yang dikenal sebagai *traveling wave*. Gelombang ini merambat di sepanjang saluran transmisi ke kedua arah dari titik gangguan dengan kecepatan mendekati kecepatan cahaya [10]. Analisis transien berfokus pada pendekripsi dan interpretasi "ledakan" sinyal-sinyal gangguan non-stasioner. Transien terjadi dalam beberapa milidetik pertama sebelum sistem mencapai kondisi stabil baru atau kembali normal.

Pendekatan investigasi transien ini secara fundamental berbeda dengan analisis *steady-state* konvensional. Analisis *steady-state* mengamati perilaku sistem setelah semua komponen transien mereda dan sinyal mencapai bentuk stabil yang baru (*steady state*). Sebaliknya, analisis transien justru memanfaatkan "kekacauan" awal pasca-gangguan. Analisis pada sinyal yang kaya informasi ini selaras dengan kebutuhan untuk deteksi gangguan yang cepat di sistem tenaga modern saat ini [11].

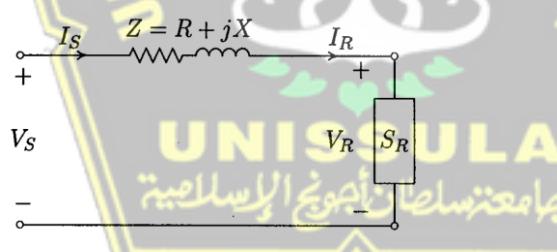
Validitas penggunaan analisis transien untuk diagnosis gangguan terletak pada fakta bahwa sinyal transien yang dihasilkan oleh gangguan bersifat unik dan informatif. Sinyal-sinyal ini dapat memberikan informasi ekstensif mengenai tipe gangguan, deteksi, lokasi, arah, dan durasinya [10]. Kehadiran komponen frekuensi tinggi yang tiba-tiba muncul pada sinyal tegangan dan arus merupakan indikator yang sangat jelas bahwa sebuah anomali telah terjadi. Karena sinyal ini muncul

seketika dan dapat dibedakan dengan jelas dari kondisi operasi normal, analisisnya menjadi dasar yang kuat untuk merancang sistem proteksi yang cepat dan akurat.

Oleh karena itu, pemanfaatan analisis transien menjadi landasan penting dalam pengembangan skema proteksi modern yang menuntut kecepatan dan keandalan. Dengan menganalisis sinyal pada jendela waktu yang sangat singkat sesaat setelah gangguan terdeteksi, dimungkinkan untuk mengekstraksi fitur-fitur kunci sebelum sinyal tersebut terdistorsi oleh respons sistem lainnya. Pendekatan ini tidak hanya mempercepat waktu deteksi secara drastis tetapi juga meningkatkan akurasi klasifikasi gangguan. Analisis ini menjadi metode yang sangat relevan dan fundamental untuk penelitian ini.

5. Model Saluran Transmisi Pendek

Untuk keperluan analisis dan simulasi, saluran transmisi dapat dimodelkan secara matematis. Pada tugas akhir ini, digunakan model saluran transmisi pendek (*short transmission line*) yang relevan untuk saluran dengan panjang kurang dari 80 km [1]. Rangkaian ekuivalen satu fasa untuk model ini diilustrasikan pada Gambar 2.4.



Gambar 2.4 Model Saluran Transmisi Pendek

Pada model pada Gambar 2.5, Vs dan Is merepresentasikan tegangan dan arus pada sisi pengirim (*sending end*). Adapun Vr dan Ir adalah tegangan dan arus pada sisi penerima (*receiving end*). Keseluruhan saluran transmisi direpresentasikan oleh sebuah impedansi seri tunggal Z, yang terdiri dari resistansi total R dan reaktansi induktif X [1].

Nilai reaktansi induktif (X) muncul akibat medan magnet di sekitar konduktor yang nilainya bergantung pada frekuensi sis tem (f) serta induktansi (L) saluran sesuai dengan persamaan:

$$X = 2\pi f L \quad (2.4)$$

Model ini menyederhanakan representasi saluran dengan hanya memperhitungkan impedansi seri (resistansi dan induktansi) dan mengabaikan kapasitansi *shunt* yang tidak signifikan. Penyederhanaan ini dapat digunakan karena efek kapasitansi pada saluran pendek sangat kecil dan tidak signifikan memengaruhi hasil analisis gangguan hubung singkat.

2.2.2 Analisis Sinyal Digital

Ketika gangguan terjadi pada saluran transmisi, bentuk gelombang tegangan dan arus akan mengalami perubahan drastis dari kondisi normalnya. Sinyal analog ini dapat diubah menjadi data digital melalui proses sampling untuk dianalisis lebih lanjut. Analisis sinyal digital (*Digital Signal Processing/DSP*) adalah teknik yang digunakan untuk mengekstraksi informasi atau fitur-fitur penting dari sinyal digital tersebut. Dalam konteks klasifikasi gangguan, fitur-fitur ini dapat berupa amplitudo puncak, komponen frekuensi, atau parameter statistik lainnya yang unik untuk setiap jenis gangguan. Ekstraksi fitur yang tepat merupakan langkah fundamental sebelum data dimasukkan ke dalam model *neural network* klasifikasi gangguan.

2.2.3 Transformasi Wavelet

Transformasi *Wavelet* merupakan metode dekomposisi data yang mampu mengidentifikasi komponen frekuensi yang berbeda dari sebuah sinyal. Kemampuan identifikasi ini menjadikan transformasi wavelet sangat efektif untuk menganalisis sinyal non-stasioner, seperti gangguan pada sistem tenaga.

1. *Discrete Wavelet Transform (DWT) dan Dekomposisi Sinyal*

Discrete Wavelet Transform (DWT) adalah teknik analisis sinyal yang sering digunakan dalam analisis gangguan sistem tenaga, khususnya untuk mendeteksi transisi atau perubahan mendadak pada suatu sinyal. Berbeda dengan metode lain

yang memiliki resolusi frekuensi yang sama di semua pita, *DWT* memungkinkan untuk menganalisis sinyal dengan lebih detail pada berbagai skala. Di antara mother wavelet yang paling sering digunakan adalah *Morlet* dan *Daubechies*. *Daubechies* memiliki bentuk yang menyerupai segitiga, sehingga meskipun interpretasinya lebih sulit, tetapi memiliki beberapa sifat matematis yang sangat berguna [12].

Proses inti dari *DWT* adalah dekomposisi sinyal secara multiresolusi. Proses ini dimulai dengan memecah sinyal masukan menjadi dua komponen: komponen berfrekuensi rendah dan komponen berfrekuensi tinggi. Pemecahan ini dilakukan dengan melewatkannya melalui sepasang filter digital, yaitu low-pass filter dan high-pass filter. Selanjutnya, komponen berfrekuensi rendah yang dihasilkan kemudian dipecah lebih lanjut dengan melewatkannya kembali ke pasangan filter yang sama. Proses ini diulang beberapa kali sesuai dengan jumlah level dekomposisi yang diinginkan untuk menghasilkan komponen sinyal yang dibutuhkan [12].

Secara matematis, proses dekomposisi ini didasarkan pada dua fungsi fundamental: *scaling function* $\phi(t)$ dan *wavelet function* $w(t)$. *Scaling function* $\phi(t)$ berasosiasi dengan *low-pass filter* ($h_0(k)$), sementara *wavelet function* $w(t)$ berasosiasi dengan *high-pass filter* ($h_1(k)$). Hubungan antara fungsi-fungsi ini dengan koefisien filter dijelaskan dalam persamaan berikut:

$$\phi(t) = \sum_k 2h_0(k)\phi(2t - k) \quad (2.5)$$

$$\phi(t) = \sum_k 2h_1(k)\phi(2t - k) \quad (2.6)$$

[12]

dengan,

$\phi(t)$: *Fungsi Skala (Scaling Function)*.

$w(t)$: *Fungsi Wavelet (Mother Wavelet)*.

$h_0(k)$: *Koefisien filter low-pass*.

$h_1(k)$: *Koefisien filter high-pass*.

t : *Variabel waktu kontinu*.

k : *Indeks pergeseran waktu (integer)*.

2. Koefisien Aproksimasi (A) dan Detail (D)

Hasil dari proses dekomposisi sinyal menggunakan *Discrete Wavelet Transform (DWT)* pada setiap levelnya adalah dua set koefisien. Koefisien yang dihasilkan oleh low-pass filter disebut sebagai Koefisien Aproksimasi (A). Koefisien ini merepresentasikan komponen sinyal berfrekuensi rendah, atau bisa dianggap sebagai versi sinyal yang lebih "kasar" (*coarse*). Koefisien Aproksimasi dari satu level dekomposisi kemudian menjadi sinyal masukan untuk proses dekomposisi di level selanjutnya.

Di sisi lain, koefisien yang dihasilkan oleh high-pass filter disebut sebagai Koefisien Detail (D). Koefisien inilah yang paling penting dalam konteks deteksi gangguan, karena komponen transien dari sebuah sinyal diekstraksi sebagai keluaran dari high-pass filter. Koefisien Detail menangkap informasi berfrekuensi tinggi, seperti lonjakan tajam, osilasi, dan diskontinuitas lainnya yang menjadi ciri khas dari sebuah sinyal gangguan. Oleh karena itu, analisis terhadap Koefisien Detail menjadi kunci untuk mengidentifikasi gangguan pada saluran transmisi.

3. Aplikasi *Discrete Wavelet Transform (DWT)* untuk Ekstraksi Fitur Gangguan

Kemampuan DWT dalam memisahkan komponen sinyal menjadikannya sebagai alat yang sangat efektif dan telah umum digunakan untuk tahap ekstraksi fitur dalam berbagai penelitian klasifikasi gangguan [6,7,9]. Dengan menganalisis koefisien-koefisien hasil dekomposisi, ciri atau karakteristik dari setiap sinyal gangguan dapat diekstraksi secara kuantitatif. Nilai ini kemudian dapat dijadikan masukan untuk model klasifikasi cerdas.

Dalam analisis sinyal, penghitungan energi merupakan salah satu metode yang sudah umum digunakan. Secara formal, energi dari sebuah sinyal waktu diskrit didefinisikan sebagai jumlah dari kuadrat magnitudo setiap sampelnya [13]. Prinsip ini kemudian diterapkan secara konsisten dalam analisis wavelet. Dalam konteks ini, energi wavelet pada skala tertentu diestimasi dengan menjumlahkan kuadrat magnitudo dari koefisien-koefisien wavelet seperti pada persamaan:

$$Ex = \sum |x[n]|^2 \quad (2.7)$$

[13]

di mana,

Ex: Energi total dari sebuah sinyal diskrit

x[n]: Properti dari suatu sinyal

Nilai energi ini dapat berfungsi sebagai fitur tunggal yang ringkas dan informatif untuk merepresentasikan karakteristik sinyal gangguan.

4. Jaringan Saraf Tiruan

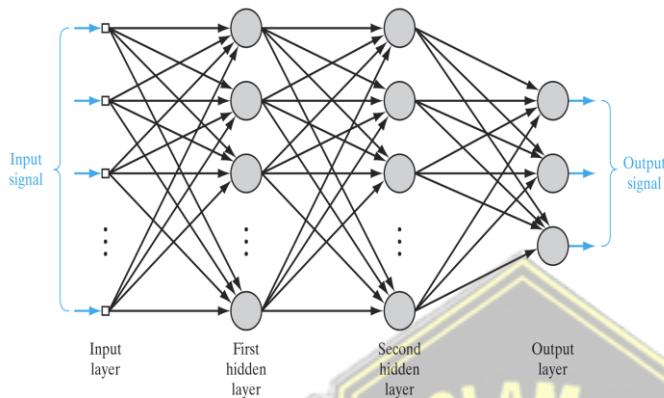
Jaringan Saraf Tiruan (JST) atau *Artificial Neural Network (ANN)* adalah sebuah model komputasi yang terinspirasi dari struktur dan cara kerja jaringan saraf biologis di otak. Penggunaan metode JST dalam rekayasa dipilih karena kemampuannya yang unggul dalam mempelajari pola-pola kompleks dan non-linear langsung dari data, tanpa memerlukan model matematis eksplisit dari sistem yang dianalisis. Kemampuan ini dijelaskan oleh Haykin sebagai berikut:

Neural Network adalah sekumpulan prosesor yang tersebar secara luas, tersusun dari unit-unit pemrosesan sederhana, dan memiliki kecenderungan alami untuk menyimpan informasi dari pengalaman. *Neural Network* menyerupai cara kerja otak manusia dalam dua hal:

1. Pengetahuan diperoleh dari lingkungan melalui proses belajar.
2. Kekuatan dari *Neuron* yang saling terhubung (*synaptic weights*) digunakan untuk menyimpan pengetahuan yang telah didapat [14].

Beberapa kelebihan utama dari JST adalah kemampuannya untuk belajar dari contoh (*training data*), melakukan generalisasi terhadap data baru yang belum pernah dilihat, serta memiliki toleransi terhadap data yang tidak lengkap atau mengandung noise.

2.2.5 *Multi-Layer Perceptron*



Gambar 2.5 Arsitektur Multi-layer Perceptron

Multi-Layer Perceptron (MLP) adalah salah satu arsitektur JST jenis *feedforward* yang paling umum digunakan. Struktur MLP dasar seperti pada Gambar 2.6 terdiri dari setidaknya tiga lapisan neuron: satu lapisan masukan (*input layer*), satu atau lebih lapisan tersembunyi (*hidden layers*), dan satu lapisan keluaran (*output layer*). Setiap neuron pada satu lapisan terhubung dengan semua neuron di lapisan berikutnya. Lapisan masukan menerima data fitur hasil ekstraksi sinyal, lapisan tersembunyi bertugas untuk memproses dan mentransformasi data tersebut, dan lapisan keluaran menghasilkan hasil klasifikasi akhir misalnya, jenis gangguan.

1. Arsitektur *Multi-Layer Perceptron*

Arsitektur *Multi-Layer Perceptron* secara klasik terdiri dari tiga jenis lapisan (*layer*):

a. Lapisan Input (Input Layer):

Lapisan ini berfungsi sebagai pintu gerbang atau titik masuk bagi data ke dalam jaringan. Input layer terdiri dari sejumlah neuron yang disebut sebagai *source nodes*. Jumlah neuron pada lapisan ini akan sama dengan jumlah fitur dalam dataset. Sebagai contoh, untuk dataset dengan 8 fitur, maka input layer akan memiliki 8 neuron.

Setiap neuron pada input layer *MLP* dirancang untuk menerima satu nilai fitur tunggal (skalar). Data input untuk *MLP* bisa berupa sebuah vektor datar (misalnya, 1×8), di mana setiap elemen dari vektor itu akan menjadi masukan untuk satu neuron. Hal ini berbeda dengan arsitektur JST yang lebih modern (seperti *Convolutional Neural Network*) yang mampu memproses data dalam bentuk matriks atau tensor secara langsung pada unit inputnya [15].

b. Lapisan Tersembunyi (*Hidden Layer*):

Hidden Layer merupakan lapisan komputasi yang terletak di antara lapisan input dan output. Suatu jaringan *MLP* bisa memiliki satu atau lebih lapisan tersembunyi. Di lapisan ini proses "belajar" pola-pola kompleks terjadi. Setiap neuron di lapisan ini terhubung sepenuhnya (*fully connected*) ke semua neuron di lapisan sebelumnya [15].

c. Lapisan Output (*Output Layer*):

Bagian ini merupakan lapisan terakhir yang menghasilkan respon akhir dari jaringan terhadap pola aktivasi yang diberikan oleh *source nodes* pada *input layer* pertama. Jumlah neuron pada lapisan ini ditentukan oleh jenis masalah yang akan diprediksi [15]. Sebagai contoh, untuk mengklasifikasikan 11 jenis gangguan, maka output layer akan memiliki 11 neuron yang mewakili setiap jenis gangguan.

2. Neuron dan Fungsi Aktivasi

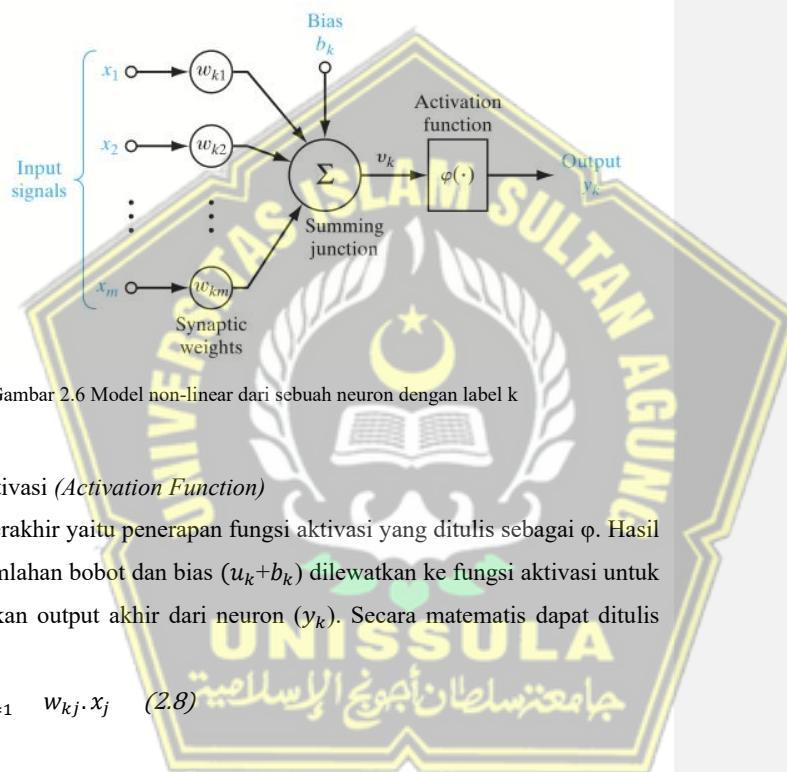
Lapisan-lapisan dalam arsitektur *Multi-layer Perceptron* tersusun dari unit-unit komputasi dasar yang disebut neuron. Sebagaimana dijelaskan oleh Haykin (2009), sebuah neuron adalah unit pemrosesan informasi fundamental yang menjadi dasar dari cara kerja sebuah jaringan saraf. Operasi yang terjadi di dalam sebuah neuron dapat dipecah menjadi beberapa elemen dasar:

a. Penjumlahan Linear (*Linear Combiner*)

Setiap sinyal masukan (x) yang diterima neuron terhubung melalui sebuah *synapse* yang memiliki bobot (*weight*) atau "kekuatan" koneksi. Langkah pertama adalah mengalikan setiap sinyal masukan dengan bobotnya masing-masing. Selanjutnya semua hasil perkalian itu akan dijumlahkan.

b. Penambahan Bias

Setelah hasil dari penjumlahan didapatkan, sebuah parameter yang disebut “bias” (b_k) ditambahkan. Sesuai dengan Gambar 2.7 fungsi dari bias adalah untuk menaikkan atau menurunkan total sinyal masukan sebelum diproses lebih lanjut.



c. Fungsi Aktivasi (*Activation Function*)

Langkah terakhir yaitu penerapan fungsi aktivasi yang ditulis sebagai φ . Hasil dari penjumlahan bobot dan bias ($u_k + b_k$) dilewatkan ke fungsi aktivasi untuk menghasilkan output akhir dari neuron (y_k). Secara matematis dapat ditulis sebagai:

$$u_k = \sum_{j=1}^m w_{kj} \cdot x_j \quad (2.8)$$

dan

$$y_k = \varphi(u_k + b_k) \quad (2.9)$$

y_k = Sinyal keluaran dari neuron ke- k .

φ = Fungsi aktivasi.

u_k = Hasil penjumlahan terbobot untuk neuron ke- k .

b_k = Nilai bias untuk neuron ke- k .

w_{kj} = Bobot koneksi dari neuron ke- j (lapisan sebelumnya) ke neuron ke- k .

x_j = Sinyal masukan ke- j .

m = Jumlah total sinyal masukan.

k = Indeks untuk neuron pada lapisan saat ini.

j = Indeks untuk sinyal masukan dari lapisan sebelumnya.

Fungsi aktivasi bertujuan untuk membatasi amplitudo dari output neuron dengan "menekan" (*squashing*) nilainya ke dalam rentang tertentu. Keberadaan fungsi ini memungkinkan jaringan untuk mempelajari pola-pola yang kompleks [14].

2.2.6 Pelatihan Jaringan Saraf Tiruan

1. Backpropagation

Algoritma *Backpropagation* adalah mekanisme standar untuk melatih model *Multi-layer Perceptron* dalam *supervised learning*. Algoritma ini memungkinkan informasi dari fungsi *cost* yang mengukur besaran *error* antara prediksi dan target sebenarnya untuk berjalan mundur melalui jaringan guna menghitung gradien [15]. Secara sederhana, algoritma ini bekerja dalam dua tahap: *forward pass* untuk menghasilkan prediksi dan menghitung *error*, kemudian dilanjutkan dengan *backward pass* di mana *error* tersebut digunakan untuk memperbarui seluruh bobot secara iteratif. Tujuan akhirnya adalah untuk meminimalkan *error* hingga model mencapai tingkat akurasi yang diinginkan [15].

2. Epoch

Pada proses pelatihan sebuah jaringan saraf, pembaruan bobot-bobot sinaptik dilakukan secara iteratif untuk meminimalkan *error* dan meningkatkan akurasi model. Akan tetapi, proses pembelajaran ini tidak berjalan tanpa henti, melainkan diorganisir ke dalam unit-unit terstruktur yang disebut sebagai epoch.

Satu epoch didefinisikan sebagai satu siklus lengkap di mana algoritma pembelajaran telah "melihat" atau dipresentasikan dengan seluruh sampel yang ada di dalam set data latih. Dengan kata lain, penyesuaian terhadap bobot-bobot sinaptik pada *Multi-layer Perceptron* dilakukan secara *epoch-by-epoch*. Proses pelatihan dilakukan hingga model mencapai tingkat konvergensi yang memuaskan. Pada

setiap epoch, urutan data latih dapat diacak (*shuffled*) untuk meningkatkan kemampuan generalisasi model [15].

3. Overfitting dan Underfitting

Tujuan dari proses pelatihan sebuah jaringan saraf tidak hanya untuk memetakan relasi input-output dari data yang ada, melainkan juga memastikan model tersebut mampu melakukan generalisasi dengan baik. Akan tetapi, dalam mencapai kemampuan generalisasi ini, terdapat dua tantangan praktis yang seringkali muncul, yaitu *overfitting* dan *underfitting*.

Overfitting, atau terkadang disebut *overtraining*, adalah sebuah fenomena yang terjadi ketika model yang dilatih menjadi terlalu bergantung pada data latih. Dalam kondisi ini, model tidak hanya mempelajari pola-pola fundamental dari data, tetapi juga mulai "menghafal" detail-detail yang tidak relevan, termasuk noise atau anomali yang hanya ada pada sampel data latih. Akibat dari fenomena ini adalah model akan menunjukkan performa yang sangat tinggi pada data latih, namun performanya akan menurun drastis ketika dihadapkan pada data baru yang belum pernah dilihat sebelumnya. Pada dasarnya, ketika sebuah jaringan mengalami *overfitting*, ia kehilangan kemampuan esensialnya dalam melakukan generalisasi [15].

Sebaliknya, *underfitting* terjadi ketika proses pelatihan belum cukup untuk model dapat menangkap pola fundamental dari data. Akibatnya, model akan menunjukkan performa yang buruk baik pada data latih maupun data baru. Hal ini merupakan indikasi bahwa model belum cukup "belajar" dan belum mampu mengambil pola dari masalah yang ada.

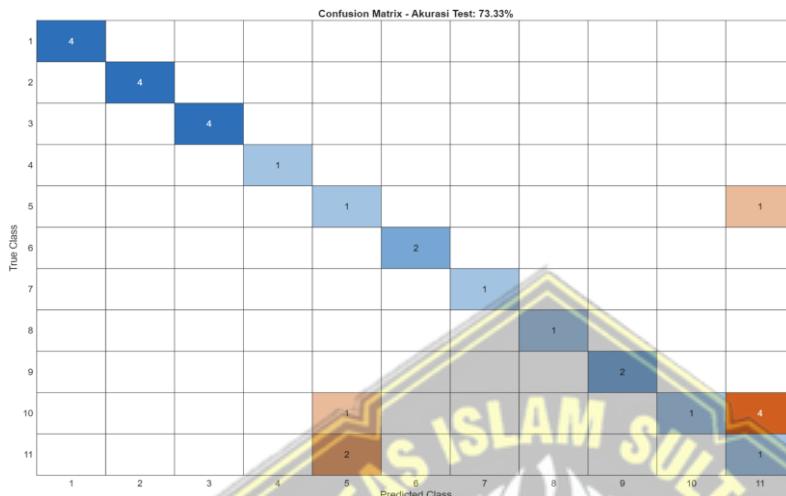
Untuk mengatasi permasalahan ini, khususnya *overfitting*, pendekatan standar yang digunakan adalah dengan memvalidasi model secara berkala selama proses pelatihan. Hal ini dilakukan dengan mempartisi dataset yang tersedia menjadi beberapa bagian, salah satunya adalah set data validasi (*validation subset*). Data validasi digunakan secara khusus untuk menguji dan memvalidasi kemampuan generalisasi model pada data yang tidak ikut serta dalam proses pembaruan bobot [15].

4. *Confusion Matrix* dan Evaluasi Model

Salah satu metode visualisasi dan analisis kinerja model kecerdasan buatan yang paling fundamental dalam *machine learning* adalah *Confusion Matrix*. Sebagaimana dipaparkan oleh Fawcett (2006), *confusion matrix* adalah sebuah tabel yang dirancang untuk menyajikan visualisasi performa dari sebuah model klasifikasi. Struktur dasarnya membandingkan antara Kelas Aktual (*True Class*) atau label yang sebenarnya dari data, dengan Kelas Prediksi (*Predicted Class*) atau label yang diprediksi oleh model [16].

Cara kerja *confusion matrix* adalah dengan menghitung dan mengkategorikan setiap prediksi yang dibuat oleh model pada data uji. Angka-angka di sepanjang diagonal utama dari matriks ini merepresentasikan jumlah total prediksi yang benar, di mana Kelas Prediksi (*Predicted Class*) sama dengan Kelas Aktual (*True Class*). Sebaliknya, semua angka yang berada di luar diagonal utama merepresentasikan total kesalahan atau "kebingungan" (*confusion*) yang dibuat oleh model, di mana Kelas Prediksi tidak sesuai dengan Kelas Aktual [16]. Meskipun seringkali diilustrasikan dalam bentuk matriks 2×2 untuk masalah klasifikasi biner, konsep ini dapat diperluas secara langsung untuk klasifikasi dengan jumlah kelas yang lebih besar. Pada penelitian ini, menghasilkan matriks berukuran $N \times N$ (misalnya, 11×11).





Gambar 2.7 Contoh Confusion Matrix

Gambar 2.8 menunjukkan contoh dari suatu *confusion matrix* di mana blok diagonal berwarna biru adalah kelas yang berhasil diprediksi dengan benar

Akurasi model secara keseluruhan juga dapat dihitung dari confusion matrix.

Akurasi ini merupakan rasio antara jumlah total prediksi yang benar terhadap jumlah total seluruh sampel data yang diuji. Akurasi dihitung menggunakan rumus berikut:

$$\text{Akurasi} = \frac{\text{Jumlah Prediksi Benar}}{\text{Total Jumlah Data Uji}} \times 100\% \quad (2.9)$$

Evaluasi model sering kali dilakukan secara berulang untuk mendapatkan gambaran kinerja yang lebih stabil dan tidak bias oleh fluktuasi karena pembagian data yang acak. Oleh karena itu beberapa fungsi statistik dasar juga disertakan. Rata-rata digunakan untuk menentukan nilai tengah dari serangkaian percobaan, dihitung dengan rumus:

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (2.10)$$

[17]

Sementara itu, standar deviasi digunakan untuk mengukur tingkat sebaran atau fluktuasi dari serangkaian hasil uji. Karena pengujian berulang ini dianggap

sebagai sampel dari semua kemungkinan hasil, maka digunakan rumus Standar Deviasi Sampel. Nilai yang rendah mengindikasikan kinerja model yang konsisten. Standar deviasi dihitung sebagai berikut:

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}} \quad (2.11)$$

[17]

Di mana,

\bar{x} : nilai rata-rata

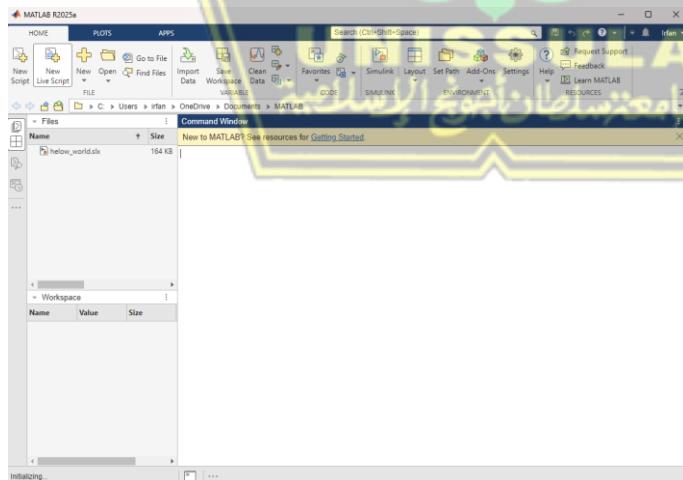
s : standar deviasi sampel

x_i : akurasi pada pengujian ke- i

n : jumlah total pengujian.

2.2.7 MATLAB

MATLAB (*Matrix Laboratory*) adalah sebuah *environment* komputasi numerik dan bahasa pemrograman tingkat tinggi yang dikembangkan oleh *MathWorks*. Perangkat lunak ini banyak digunakan dalam bidang rekayasa dan sains untuk analisis data, pengembangan algoritma, dan pembuatan model.



Gambar 2.8 Tampilan awal *MATLAB R2025a*

Gambar 2.9 merupakan tangkapan layar dari tampilan awal *MATLAB R2025a*

1. Simulink

Simulink adalah sebuah environment berbasis grafis yang terintegrasi di dalam *MATLAB* untuk pemodelan, simulasi, dan analisis sistem dinamis. Dalam tugas akhir ini, *Simulink* digunakan untuk membangun model sistem tenaga listrik, termasuk saluran transmisi, dan untuk mensimulasikan berbagai skenario gangguan.

2. Deep Learning Toolbox

Deep Learning Toolbox adalah salah satu fitur di dalam *MATLAB* untuk merancang, melatih, dan menganalisis model jaringan saraf tiruan, termasuk *Multi-Layer Perceptron (MLP)*. Toolbox ini memfasilitasi implementasi arsitektur jaringan, pengaturan parameter pelatihan seperti algoritma *backpropagation*, dan evaluasi performa model *MLP* yang sedang dirancang.

3. Signal Processing Toolbox

Signal Processing Toolbox menyediakan fungsi-fungsi untuk melakukan analisis sinyal digital. Dalam penelitian ini, *toolbox* tersebut dimanfaatkan untuk memproses data sinyal tegangan dan arus hasil simulasi *Simulink*, serta untuk mengekstraksi fitur-fitur relevan yang akan digunakan sebagai masukan untuk model *MLP*.

4. Pemrograman *MATLAB*

MATLAB sendiri sebenarnya adalah suatu bahasa pemrograman tingkat tinggi yang memungkinkan *engineer* dan peneliti melakukan pengembangan berbagai aplikasi berskala besar. Salah satu penggunaan pemrograman di *MATLAB* yang umum digunakan adalah penulisan skrip yang merupakan rangkaian perintah (*sequences of commands*). Skrip disimpan dan dijalankan dalam bentuk *file* berestensi .m yang memungkinkan pengguna untuk menjalankan fungsi custom, pemakaian ulang suatu *code blocks*, atau melakukan berbagai otomatisasi lainnya.

Berikut adalah daftar fungsi dan konsep sintaks fundamental yang digunakan:

a. Manajemen *Workspace* dan Variabel

- `clear` : Menghapus semua variabel dari *workspace* saat ini untuk memastikan tidak ada data sisa dari eksekusi sebelumnya.
- `clc` : Membersihkan teks pada Command Window untuk memudahkan pembacaan output.
- `[...]` : Sintaks untuk membuat matriks atau vektor numerik, yang merupakan tipe data dasar di *MATLAB*.
- `{...}` : Sintaks untuk membuat Cell Array, sebuah tipe data fleksibel yang dapat menyimpan elemen dengan tipe yang berbeda (misalnya, teks dan angka) dalam satu variabel.

Commented [ik3]: Memperbaiki bullet

b. Struktur Kontrol dan Iterasi

- `for ... end` : Struktur looping fundamental yang digunakan untuk mengeksekusi blok kode secara berulang.
- `length(A)` : Mengembalikan jumlah elemen dalam dimensi terpanjang dari sebuah array A.
- `size(A, dim)` : Mengembalikan jumlah elemen pada dimensi (dim) tertentu dari matriks A.

c. Interaksi dengan Model Simulink

- `set_param(...)` : Mengatur atau mengubah nilai parameter dari sebuah blok di dalam model Simulink. Fungsi ini memerlukan tiga argumen utama: path ke blok, nama parameter yang akan diubah, dan nilai baru yang akan diberikan.
- `sim(...)` : Memulai dan menjalankan simulasi pada model Simulink yang namanya dijadikan sebagai argumen.

d. Analisis dan Manipulasi Data

- `find(...)` : Mencari indeks dari elemen-elemen dalam sebuah array yang memenuhi kondisi tertentu. Sangat berguna untuk proses windowing atau memotong sinyal berdasarkan kriteria waktu.

- dwt(...) : Melakukan dekomposisi Discrete Wavelet Transform pada sebuah sinyal, yang digunakan untuk ekstraksi fitur di domain waktu-frekuensi.
 - sum(A) : Menjumlahkan semua elemen dalam sebuah vektor A.
- e. Fungsi Utilitas dan I/O (Input/Output)
- num2str(...) : Mengkonversi sebuah nilai numerik menjadi format string (teks). Seringkali diperlukan saat menggunakan set_param karena beberapa parameter di blok Simulink memerlukan input dalam bentuk string.
 - fprintf(...) : Menampilkan teks atau data yang terformat pada Command Window
 - save(...) : Menyimpan variabel dari *workspace MATLAB* ke dalam sebuah file biner (.mat) untuk penggunaan di masa mendatang.

f. Contoh Implementasi dalam Skrip

Sebagai contoh, potongan kode berikut akan melakukan iterasi untuk setiap variasi sudut awal gangguan, mengatur parameter di Simulink, dan menjalankan simulasi.

```
for sudut_idx = 1:length(sudut_list_deg)
    sudut = sudut_list_deg(sudut_idx);
    set_param('namafайл_simulink/Three-Phase Source', 'PhaseAngle',
        num2str(sudut));
    simout = sim('model_name');
```

Dalam contoh ini fungsi *loop for* memastikan bahwa blok kode di dalamnya akan dieksekusi sebanyak jumlah elemen dalam *sudut_list_deg*. Pada setiap iterasi, fungsi *set_param* secara dinamis memperbarui nilai *PhaseAngle* pada blok sesuai *path namafайл_simulink/Three-Phase Source* menjadi nilai yang tersimpan di dalam variabel *sudut*. Kemudian fungsi *sim* digunakan untuk menjalankan simulasi. Alur kerja seperti ini memungkinkan pembangkitan dataset yang komprehensif secara efisien.

BAB III

METODOLOGI PENELITIAN

3.1 Model Penelitian

Penelitian ini merupakan studi berbasis simulasi komputasi yang sepenuhnya dilakukan menggunakan perangkat lunak pada komputer. Oleh karena itu, penelitian ini tidak terikat pada lokasi fisik tertentu seperti laboratorium atau gardu induk. Seluruh proses, mulai dari perancangan sistem, simulasi gangguan, ekstraksi data, hingga perancangan model kecerdasan buatan, akan dilaksanakan secara virtual.

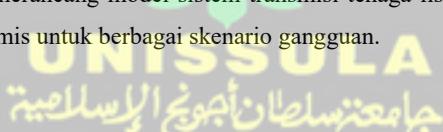
3.2 Alat dan Bahan

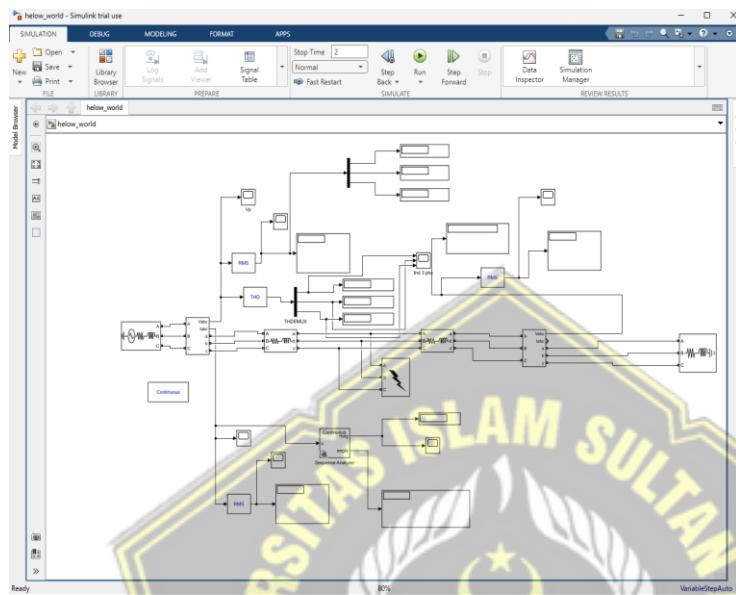
Alat dan bahan yang digunakan dalam penelitian ini terbagi menjadi perangkat keras (hardware), perangkat lunak (software), dan bahan penelitian (data).

1. Software MATLAB R2025a

MATLAB R2025a digunakan sebagai *software* komputasi teknis utama dalam penyusunan tugas akhir ini. *MATLAB* mencakup beberapa *toolbox* esensial yang akan digunakan secara terintegrasi:

- a. *Simulink*: Digunakan untuk merancang model sistem transmisi tenaga listrik dan melakukan simulasi dinamis untuk berbagai skenario gangguan.





Gambar 3.1 Simulasi Saluran Transmisi Menggunakan Simulink

Gambar 3.1 menunjukkan contoh rangkaian simulasi yang bisa dibuat menggunakan *MATLAB Simulink*.

- b. *Deep Learning Toolbox*: Digunakan untuk keseluruhan siklus kerja kecerdasan buatan. *Toolbox* ini akan dimanfaatkan untuk merancang arsitektur, melatih, melakukan validasi, dan mengevaluasi kinerja model *Multi-Layer Perceptron (MLP)*.
- c. *Signal Processing Toolbox*: Digunakan untuk melakukan analisis sinyal digital dan ekstraksi fitur dari data mentah hasil simulasi. Fungsi-fungsi dalam *toolbox* ini akan diaplikasikan untuk menghitung parameter penting seperti *Total Harmonic Distortion (THD)* dan komponen sinyal lainnya yang akan menjadi masukan bagi model.
- d. *Wavelet Toolbox*: Digunakan untuk melakukan transformasi wavelet pada fitur sinyal digital yang akan digunakan sebagai input pelatihan model *MLP*
- e. *Statistics and Machine Learning Toolbox*: Digunakan untuk melakukan perhitungan statistik dan membuat plot yang membantu analisis model yang sedang dirancang.

2. Komputer

Perangkat keras utama yang digunakan adalah satu unit *windows* laptop dan monitor dengan spesifikasi sebagai berikut:

- Prosesor: *Ryzen 7 6800U*
- RAM: 16 GB
- Penyimpanan: 1 TB SSD
- Monitor: 34" *ultrawide*

3.3 Prosedur Penelitian

Penelitian ini merupakan studi berbasis simulasi komputasi yang sepenuhnya dilakukan menggunakan perangkat lunak pada komputer. Seluruh proses, mulai dari perancangan sistem, simulasi gangguan, ekstraksi data, hingga perancangan model kecerdasan buatan, akan memanfaatkan berbagai fitur dan toolbox dalam aplikasi *MATLAB*.

Proses perancangan simulasi, pengumpulan data, hingga pembuatan model kecerdasan buatan dilaksanakan dengan beberapa prosedur yang sistematis sebagai berikut:

3.3.1 Perancangan dan Simulasi Sistem di Simulink

Perancangan model simulasi saluran transmisi pendek dalam penelitian ini menggunakan acuan penghantar SUTT 150kV Batang - Pekalongan 1. Saluran transmisi tersebut beroperasi pada tegangan nominal 150 kV dengan menggunakan konduktor jenis ACCC/TW LISBON 1x310 mm, yang memiliki Kapasitas Hantar Arus (KHA) sebesar 1285 A dan membentang sepanjang 13,16 km.

Komponen simetris yang dimiliki penghantar yaitu sebagai berikut:

1. Impedansi Urutan Positif
 - X_1 (ohm/km): 0,4000
 - R_1 (ohm/km): 0,0930
 - Z_1 (ohm): 5,4056
 - Sudut Line (deg): 76,91

2. Impedansi Urutan Nol
 - X_0 (ohm/km): 1,2000
 - R_0 (ohm/km): 0,2430
 - Z_0 (ohm): 16,1162
 - Sudut Line (deg): 78,55

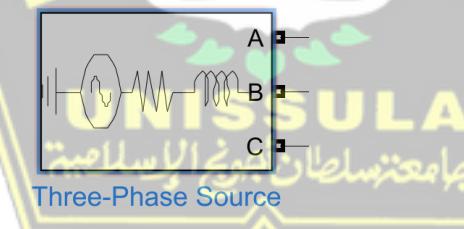
Karena saluran transmisi merupakan komponen statis, impedansi urutan negatif (Z_2) memiliki nilai yang sama dengan urutan positif ($Z_2 = Z_1$), dengan perbedaan hanya pada urutan fasa arusnya.

Untuk parameter lain yang diperlukan dalam model simulasi di *MATLAB Simulink* dan tidak secara spesifik tersedia akan diestimasi menggunakan nilai tipikal dan pendekatan teoritis. Pendekatan ini mengacu pada rentang nilai yang umum digunakan dalam literatur akademis dan penelitian relevan.

1. Blok *Simulink* yang digunakan

Saluran transmisi tegangan tinggi ini kemudian dimodelkan menggunakan *MATLAB Simulink* menggunakan beberapa blok sebagai berikut:

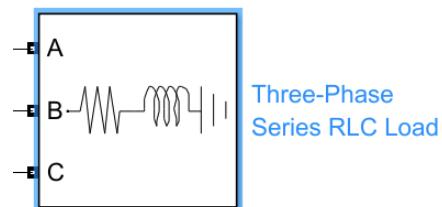
a. *Three-Phase Source*



Gambar 3.2 Blok *Three-Phase Source*

Blok pada Gambar 3.2 mewakili sumber daya yang akan menyuplai saluran transmisi. Untuk keperluan simulasi digunakan frekuensi 50 Hz sesuai standar PLN dan nilai tegangan tipikal 152 kV untuk mengkompensasi drop tegangan di sepanjang saluran.

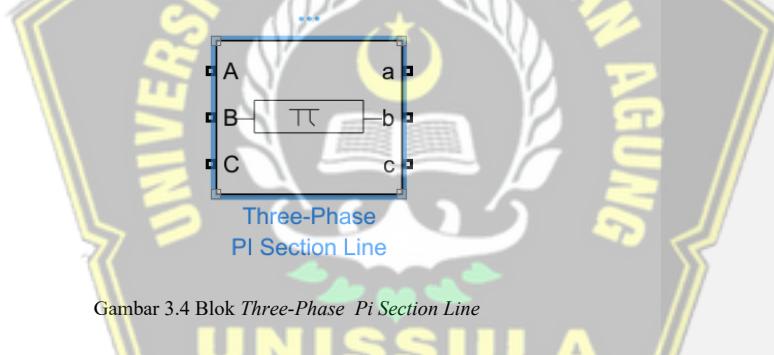
b. *Three-Phase Series RLC Load*



Gambar 3.3 Blok *Three-Phase Series RLC Load*

Blok pada Gambar 3.3 mewakili beban yang mengalir sepanjang penghantar. Pada model ini digunakan acuan nilai rata-rata beban puncak harian pada penghantar SUTT Batang - Pekalongan 1.

c. *Three-Phase Pi Section Line*



Gambar 3.4 Blok *Three-Phase Pi Section Line*

Blok pada Gambar 3.4 merepresentasikan saluran transmisi sepanjang 13,5 km dengan nilai kapasitansi yang diabaikan. Untuk parameter resistansi urutan positif dan urutan nol diambil dari data teknik penghantar PLN yaitu $R_1 = 0,0930$ dan $R_0 = 0,2430$.

Untuk parameter induktansi dapat dikalkulasikan sebagai berikut:

Diketahui $X_1 = 0,4$; $X_0 = 1,2$ (data teknik penghantar PLN)

maka menggunakan persamaan (2.4) L_1 dan L_0 dapat dihitung sebagai ,

$$X_1 = 2\pi f L_1$$

$$0,4 = 2\pi \cdot 50 \cdot L_1$$

$$0,4 = 314,159 \cdot L_1$$

$$L_1 = \frac{0,4}{314,159}$$

$$L_1 = 0,001273 \Omega/km \tan$$

dan

$$X_0 = 2\pi f L_0$$

$$1,2 = 2\pi \cdot 50 \cdot L_0$$

$$1,2 = 314,159 \cdot L_0$$

$$L_1 = \frac{1,2}{314,159}$$

$$L_0 = 0,003820 \Omega/km$$

dengan

X_1 = Reaktansi urutan positif

X_0 = Reaktansi urutan nol

L_1 = Induktansi urutan positif

L_0 = Induktansi urutan nol

f = Frekuensi sistem (50 Hz)

Pada model ini digunakan dua Blok *Three-Phase Pi Section Line* untuk mensimulasikan variasi gangguan yang berada di satu rentang saluran transmisi Batang - Pekalongan 1. Detail akan dijelaskan pada bagian rangkaian dan arsitektur simulasi.

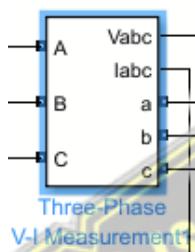
d. Three-Phase Fault



Gambar 3.5 Blok *Three-Phase Fault*

Blok pada Gambar 3.5 yaitu *Three-Phase Fault* akan mensimulasikan gangguan yang terjadi di fasa A, B, C, dan Ground. Waktu *switching* dan resistansi gangguan dapat diatur untuk menghasilkan model yang dikehendaki seperti 3,5

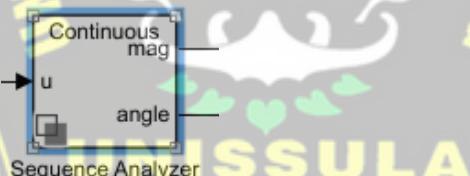
e. *Three-Phase V-I Measurement*



Gambar 3.6 Blok *Three-Phase V-I Measurement*

Blok pada Gambar 3.6 digunakan untuk melakukan pengukuran arus dan tegangan tiga fasa di titik yang dikehendaki. Arus dan tegangan dari sistem akan diproses untuk menjadi input bagi model *Multi-Layer Perceptron*.

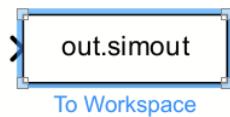
f. *Sequence Analyzer*



Gambar 3.7 Blok *Sequence Analyzer*

Blok pada Gambar 3.7 yaitu *Sequence Analyzer* digunakan untuk mengambil komponen positif, negatif, dan nol dari suatu sinyal tiga fasa. Pada model ini yang dipakai adalah komponen arus urutan nol (I0).

g. *To Workspace*



Gambar 3.8 Blok *To Workspace*

Blok pada Gambar 3.8 digunakan untuk menyimpan hasil pengukuran arus dan tegangan ke dalam *Workspace MATLAB*

h. Powergui



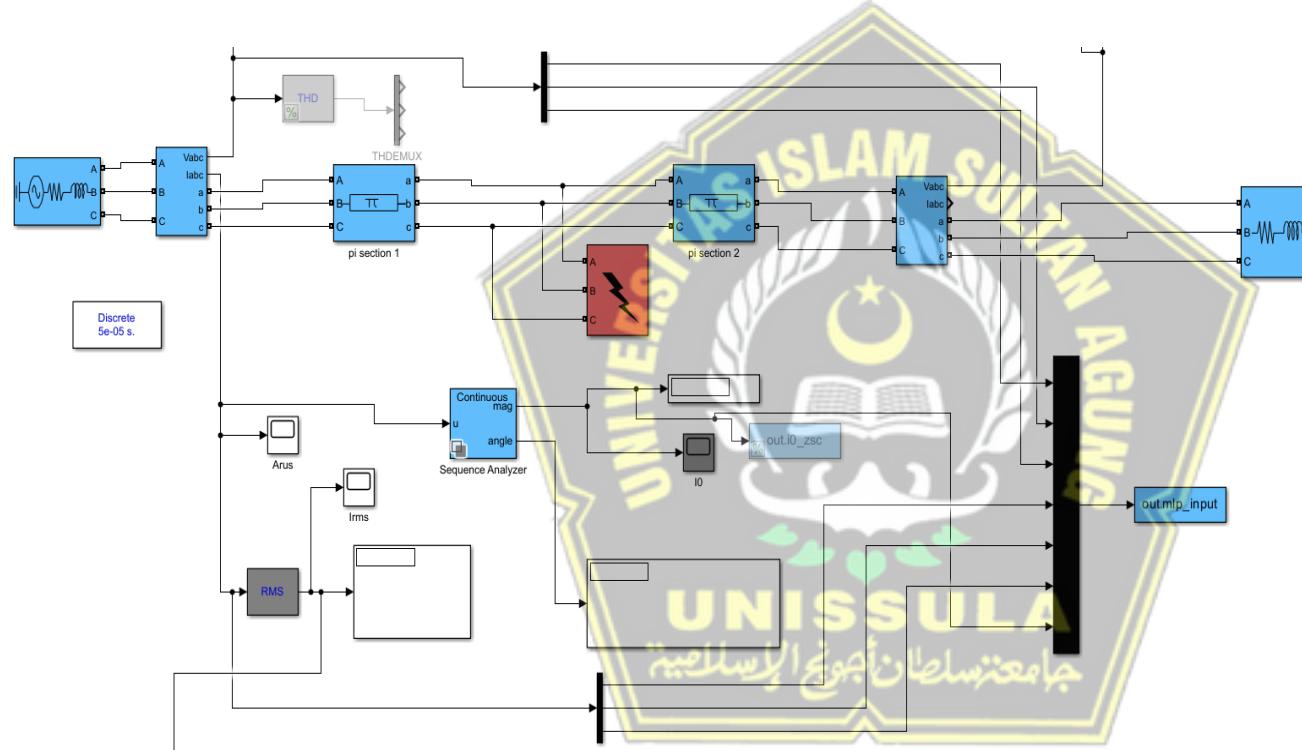
Gambar 3.9 Blok *Powergui*

Blok pada Gambar 3.9 berfungsi mengatur metode numerik yang digunakan untuk menjalankan simulasi di Simulink. Pada model ini digunakan pengaturan *Discrete Simulation Type* dengan *sample time* $5\text{e-}5$. Artinya ketika simulasi dijalankan *solver* akan melakukan *sampling* data tiap $5\text{e-}5$ s atau 50 mikrosekon.



2. Rangkaian dan Arsitektur Simulasi

Commented [ik4]: Dibuat landscape



Gambar 3.10 Rangkaian Simulasi Transmisi di Simulink

Rangkaian yang digunakan dalam simulasi ini ditunjukkan pada Gambar 3.10. Model ini merepresentasikan sebuah sistem transmisi tenaga listrik sederhana yang terdiri dari sebuah sumber tegangan tiga fasa menggunakan blok *Three-Phase Source* pada sisi pengirim dan sebuah beban seri menggunakan blok *Three-Phase Series RLC Load* pada sisi penerima. Sistem ini secara umum merepresentasikan karakteristik sistem transmisi tegangan tinggi.

Saluran transmisi di sini dimodelkan menggunakan dua buah blok *Pi Section* yang dihubungkan secara seri. Di antara kedua blok tersebut, ditempatkan sebuah blok gangguan tiga fasa (*Three-Phase Fault*). Arsitektur ini dirancang secara spesifik untuk memungkinkan simulasi gangguan pada berbagai titik di sepanjang saluran transmisi. Proses variasi lokasi ini diotomatisasi sepenuhnya melalui sebuah skrip *MATLAB* eksternal yang mengendalikan parameter di dalam blok *Pi Section* secara dinamis. Skrip tersebut melakukan iterasi pada sebuah *list* yang berisi daftar lokasi gangguan dalam satuan persen dari total panjang saluran.

Untuk setiap nilai persentase lokasi, skrip akan menghitung panjang masing-masing blok *Pi Section* secara proporsional. Berdasarkan Hukum Tegangan Kirchhoff (KVL) pada rangkaian AC, impedansi ekuivalen (Z_{total}) dari komponen yang terhubung seri adalah hasil penjumlahan fasor dari masing-masing impedansinya ($Z_{total} = Z_1 + Z_2$). Oleh karena itu, pada rangkaian ini panjang seksi pertama (dideklarasikan sebagai variabel '*L1_length*') dihitung sebagai persentase lokasi dikalikan dengan total panjang saluran (13.5 km), sementara panjang seksi kedua (dideklarasikan sebagai variabel '*L2_length*') dihitung sebagai sisanya dari total panjang saluran tersebut.

Sebagai contoh, untuk mensimulasikan gangguan yang terjadi tepat di tengah saluran (50%), skrip akan mengatur *L1_length* menjadi:

$$L1_length = 0.5 * 13.5 \text{ km} = 6.75 \text{ km}.$$

Maka *L2_length* dihitung sebagai:

$$L2_length = 13.5 \text{ km} - L1_length$$

$$L2_length = 13.5 \text{ km} - 6.75 \text{ km} = 6.75 \text{ km}.$$

Dengan cara ini, blok gangguan yang terletak di antara kedua seksi tersebut akan terletak pada jarak 6.75 km dari sisi pengirim. Metode ini memungkinkan untuk pembuatan dataset dalam skala besar secara efisien tanpa perlu mengubah parameter lokasi secara manual untuk setiap skenario lokasi gangguan.

Seluruh pengukuran sinyal tegangan (V_{abc}) dan arus (I_{abc}) dilakukan pada sisi pengirim (sending end). Pemilihan titik pengukuran ini didasarkan pada praktik umum sistem proteksi, di mana relai proteksi dan perangkat pemantauan umumnya ditempatkan di gardu induk pada pangkal saluran transmisi untuk mendeteksi gangguan secepat mungkin. Untuk mengekstrak fitur arus urutan nol (I_0), yang merupakan indikator penting untuk gangguan yang melibatkan tanah, sinyal arus tiga fasa dihubungkan ke sebuah blok Sequence Analyzer yang dikonfigurasi untuk mengeluarkan nilai magnitudo dari komponen urutan nol.

Untuk proses agregasi data, total tujuh sinyal yang terdiri dari tiga sinyal tegangan (V_{abc}), tiga sinyal arus (I_{abc}), dan satu sinyal magnitudo arus urutan nol (I_0) diumpulkan ke dalam sebuah blok Mux . Blok ini menggabungkan ketujuh sinyal tersebut kemudian disimpan ke *workspace MATLAB* menggunakan blok *To Workspace* dengan nama *out.mlp_input*.

3.3.1 Menjalankan Simulasi dan Generasi Dataset

Setelah setiap blok dihubungkan dan diberikan parameter yang sesuai, langkah selanjutnya adalah menjalankan simulasi secara berulang untuk 11 jenis gangguan utama yaitu gangguan satu fasa ke tanah (AG, BG, CG), antar fasa (AB, BC, CA), dua fasa ke tanah (ABG, BCG, CAG), dan tiga fasa (ABC, ABCG).

1. Parameter untuk Generasi Dataset

Untuk memastikan model *Multi-Layer Perceptron* yang akan dilatih bersifat andal dan mampu melakukan klasifikasi dengan baik, diperlukan dataset yang beragam dan informatif. Oleh karena itu, setiap jenis gangguan akan dijalankan berulang kali dengan melakukan perubahan terhadap beberapa parameter operasional inti setiap kali simulasi dilakukan. Variasi ini bertujuan agar model

‘belajar’ mengenali pola dari spektrum kondisi yang lebih luas. Parameter yang divariasikan adalah sebagai berikut:

a. **Tahanan Gangguan (*Fault Resistance*):**

Untuk mencakup spektrum kondisi gangguan yang lebih luas digunakan dua nilai resistansi yaitu 0.001Ω dan 15Ω . Nilai 0.001Ω merepresentasikan kondisi gangguan solid (*bolted fault*) dengan impedansi mendekati nol. Sementara itu, pemilihan nilai 15Ω sebagai representasi gangguan berimpedansi tinggi didasarkan pada sebuah studi oleh Virgilio dan Elmer pada tahun 2015 mengenai rentang tipikal resistansi busur api (*arc resistance*) pada sistem transmisi. Penelitian menunjukkan bahwa untuk sistem dengan level tegangan 115kV-230kV, nilai resistansi gangguan 15Ω merupakan nilai yang representatif dan umum dijumpai terutama untuk skema proteksi dengan waktu tunda [18]. Dengan mencakup kedua skenario ini, model dilatih untuk mengenali gangguan baik yang bersifat solid (resistansi rendah) maupun yang berimpedansi tinggi.

b. **Lokasi Gangguan (*Fault Location*):**

Pada satu seksi saluran transmisi tegangan tinggi yang bisa membentang hingga puluhan kilometer, lokasi kemungkinan terjadinya gangguan bisa sangat acak dan tidak bisa diprediksi.

Oleh karena hal tersebut, sembilan titik lokasi diskrit digunakan, yaitu pada titik 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, dan 90% dari total panjang saluran. Pemilihan rentang ini memberikan cakupan yang lebih luas resolusi yang memadai di sepanjang saluran transmisi dan memberikan kesempatan model untuk mempelajari bagaimana karakteristik sinyal (seperti magnitudo dan fasa) berubah seiring dengan perubahan jarak gangguan dari titik pengukuran.

c. **Sudut Awal Gangguan (*Fault Inception Angle*):**

Sebagaimana titik lokasi gangguan, waktu terjadinya gangguan juga bersifat acak dan dapat terjadi pada titik mana pun dalam satu siklus gelombang AC. Untuk merepresentasikan hal ini, tiga nilai sudut fasa tegangan yang representatif digunakan dalam simulasi yaitu: 0° , 45° , dan 90° . Pemilihan parameter ini sangat krusial karena sudut awal gangguan memiliki pengaruh fundamental terhadap karakteristik sinyal transien sebagaimana dijelaskan pada bagian 2.2.1.3 tentang *DC Offset Transient*.

Besarnya komponen DC offset ini dipengaruhi secara langsung oleh nilai sesaat dari arus pada momen terjadinya gangguan.

Pada sudut 0° gangguan terjadi tepat saat tegangan melintasi titik nol, artinya arus gangguan berada pada nilai puncaknya. Pada kondisi ini sistem akan membangkitkan komponen DC offset dengan magnitudo maksimum. Hasilnya adalah sebuah gelombang arus yang sangat asimetris, di mana puncak positif dan negatifnya tidak seimbang.

Pada sudut 90° gangguan terjadi pada puncak gelombang tegangan ketika arus tunak melintasi titik nol, akibatnya transisi terjadi dengan mulus. Komponen DC offset yang dihasilkan akan bernilai minimal atau bahkan nol. Kondisi ini menghasilkan gelombang arus gangguan yang simetris [8].

Sudut 45° juga digunakan untuk merepresentasikan kondisi di antara kedua kondisi tersebut, menghasilkan komponen DC offset dan tingkat asimetri gelombang yang moderat.

Dengan memberikan variasi terhadap sudut awal gangguan, dataset yang dihasilkan akan mencakup sinyal-sinyal dengan berbagai tingkat distorsi transien. Hal ini memaksa model kecerdasan buatan untuk tidak hanya menghafal satu bentuk gelombang ideal, tetapi untuk mempelajari berbagai pola-pola fundamental yang berubah tergantung momen saat terjadinya gangguan. Kemampuan ini sangat penting untuk memastikan model yang dibangun bersifat andal (*robust*) dan mampu melakukan klasifikasi dengan baik pada kondisi dunia nyata yang tidak dapat diprediksi.

d. Tingkat Pembebanan (Load Level)

Dua kondisi beban digunakan, yaitu beban tinggi (146 MW, 41 MVAR) dan beban rendah (46.7 MW, 18.6 MVAR). Kedua nilai ini diambil dari tipikal beban puncak harian di Gardu Induk 150 kV Batang pada bulan Desember 2024 yang ditunjukkan pada Lampiran 7. Hal ini ditujukan agar performa klasifikasi model tidak sensitif terhadap kondisi operasional sistem sebelum terjadinya gangguan.

Selanjutnya seluruh proses pembangkitan data termasuk iterasi untuk setiap jenis gangguan, tahanan, lokasi, sudut, dan beban diotomatisasi sepenuhnya menggunakan sebuah skrip *MATLAB*. Skrip ini akan secara dinamis mengatur parameter-parameter pada blok-blok yang diperlukan di dalam model *Simulink*. Skrip otomatisasi ini memungkinkan untuk menjalankan simulasi secara berulang sesuai kombinasi parameter yang unik. Kemudian sebagaimana dijelaskan pada 3.3.1.2 tentang arsitektur simulasi, skrip akan menyimpan pembacaan tiga sinyal tegangan (*Vabc*), tiga sinyal arus (*Iabc*), dan satu sinyal magnitudo arus urutan nol (*I0*) ke dalam bentuk matriks dataset akhir di dalam *workspace MATLAB*.

2. Penjelasan Skrip Generasi Dataset

Untuk proses generasi dataset ini, dibuat sebuah skrip yang bertanggung jawab untuk mengatur parameter, menjalankan model *Simulink* secara berulang, mengekstraksi fitur, dan menyimpan data. Cara kerja skrip ini dapat dibagi menjadi beberapa tahapan utama sebagai berikut:

a. Inisialisasi dan Definisi Path

Pada awal eksekusi, skrip melakukan pembersihan *workspace*. Selanjutnya, didefinisikan variabel-variabel *string* yang berisi nama model *Simulink* dan *path* spesifik menuju blok-blok utama yang parameternya akan diubah secara dinamis selama simulasi. Pendekatan ini membuat skrip lebih modular dan mudah dibaca.

Cuplikan skrip *MATLAB*:

```
% Membersihkan workspace dan command window
clear; clc;

% Mendefinisikan nama model dan path ke blok-blok utama
model_name = 'pake_pi_section_sudutv3';
fault_block_path = [model_name, '/Three-Phase Fault'];
source_block_path = [model_name, '/Three-Phase Source'];
load_block_path = [model_name, '/Three-Phase Series RLC Load'];
```

b. Deklarasi Variabel untuk Parameter yang akan divariasikan

Seluruh parameter yang akan divariasikan dideklarasikan dalam bentuk vektor atau matriks. Hal ini mencakup *list* untuk lokasi gangguan (*lokasi_persen*), tahanan gangguan (*ron_list*), sudut awal gangguan (*sudut_list_deg*), dan tingkat pembebangan (*beban_list*). Adapun untuk *switching* berbagai kondisi gangguan disimpan dalam sebuah *cell array* bernama *jenis_gangguan_db*. *Cell* ini memetakan nama setiap jenis gangguan (misalnya, 'AG') ke konfigurasi parameter 'on'/'off' yang sesuai pada blok *Three-Phase Fault* di *Simulink*.

Cuplikan skrip MATLAB:

```
% Daftar parameter yang akan diiterasi
beban_list = [146e6, 41e6; 46.7e6, 18.6e6];
sudut_list_deg = [0, 45, 90];
lokasi_persen = 0.1:0.1:0.9;
ron_list = [0.001, 15];

% Basis data untuk konfigurasi jenis gangguan
jenis_gangguan_db = { ...
    'AG', {'on', 'off', 'off', 'on'}; ...
    'BG', {'off', 'on', 'off', 'on'}; ...
    'ABC', {'on', 'on', 'on', 'off'}; ...}
```

```
% ... dan seterusnya untuk 11 jenis gangguan
};
```

c. Proses Iterasi Simulasi Menggunakan *Nested Loops*

Inti dari skrip ini adalah serangkaian lima *loop for* yang tersarang (*nested*). Struktur ini dirancang untuk menjalankan simulasi pada setiap kemungkinan kombinasi dari parameter yang telah dideklarasikan. Alur kerjanya adalah sebagai berikut:

- *Loop* terluar akan mengiterasi setiap lokasi gangguan..
- Di dalamnya, *loop* kedua mengiterasi setiap nilai tahanan gangguan.
- *Loop* ketiga mengiterasi setiap jenis gangguan.
- *Loop* keempat mengiterasi setiap tingkat pembebangan.
- *Loop* kelima mengiterasi setiap sudut gangguan.

Di dalam loop kelima ini skrip menggunakan fungsi *set_param* untuk mengatur semua parameter di model *Simulink* yang disebutkan pada bagian 3.3.2.1. Selanjutnya simulasi akan berjalan dengan perintah *sim(model_name)*, dan terakhir mengekstraksi fitur dari data mentah yang dihasilkan.

cuplikan skrip *MATLAB*:

```
% Struktur nested loop untuk iterasi semua kombinasi parameter
for loc_idx = 1:length(lokasi_persen)
    % ... (perhitungan L1_length dan L2_length)
    for res_idx = 1:length(ron_list)
        for ggn_idx = 1:length(ggn_jenis)
            for beban_idx = 1:size(beban_list, 1)
                for sudut_idx = 1:length(sudut_list_deg)

                    % 1. Menginject semua parameter yang ingin diubah di Simulink
                    set_param(...)
                    set_param(...)
```

...

```
% 2. Menjalankan simulasi via sim(...)

% 3. Melakukan ekstraksi fitur dari hasil simulasi
```

```
end
end
end
end
```

d. Pemantauan Progres Simulasi

Untuk memudahkan pengamatan selama proses simulasi berjalan, skrip dilengkapi dengan *feedback* visual yang dapat terlihat di bagian *console MATLAB*. Di bagian *loop* paling dalam ditambahkan fungsi *disp* untuk menampilkan status progres pada *Command Window*. Status ini mencakup informasi mengenai nomor simulasi saat ini, total simulasi yang akan dijalankan, serta detail kombinasi parameter yang sedang diuji.

Cuplikan skrip *MATLAB*:

```
disp(['[' num2str(sim_idx) '/' num2str(total_simulasi) ']' ...
'Loc ' num2str(current_lokasi*100) '%,' ...
'Ron ' num2str(current_ron) ' Ohm,' ...
% ... dan seterusnya untuk parameter lain
]);
```

e. Akumulasi dan Penyimpanan Data

Sebelum *loop* dimulai, dua variabel kosong (*dataset_fitur* dan *dataset_label*) diinisialisasi untuk digunakan menampung pembacaan hasil simulasi. Setelah setiap simulasi dan proses ekstraksi fitur selesai, baris fitur yang baru dihasilkan beserta labelnya ditambahkan ke dalam kedua variabel tersebut. Untuk mencegah kehilangan data jika terjadi interupsi, skrip

melakukan penyimpanan sementara (*checkpoint*) menggunakan *save* di setiap akhir iterasi. Setelah semua *loop* selesai, data yang telah terkumpul secara lengkap disimpan ke dalam sebuah *file .mat* final.

cuplikan skrip *MATLAB*:

```
dataset_fitur = [dataset_fitur; baris_fitur_sekarang];
dataset_label = [dataset_label; label_gangguan];

% Menyimpan checkpoint sementara untuk keamanan data
save('checkpoint_data.mat', 'dataset_fitur', 'dataset_label');
...

% Menyimpan data akhir setelah semua simulasi selesai
save('Dataset_Final_hasil.mat', 'dataset_fitur', 'dataset_label',
'data_mentah');
```

3. Menjalankan Skrip

Setelah Skrip selesai dibuat, buka model Simulink yang akan dijalankan kemudian klik tombol *Run* di *MATLAB*. Skrip akan mulai berjalan dan tunggu hingga proses selesai.

```
[979/1188] Loc 80%, Ron 0.001 Ohm, Beban 146 MW / 41 Mvar, Sudut 0 deg, fault ABC ...
selesai.
[980/1188] Loc 80%, Ron 0.001 Ohm, Beban 146 MW / 41 Mvar, Sudut 45 deg, fault ABC ...
selesai.
[981/1188] Loc 80%, Ron 0.001 Ohm, Beban 146 MW / 41 Mvar, Sudut 90 deg, fault ABC ...
selesai.
[982/1188] Loc 80%, Ron 0.001 Ohm, Beban 46.7 MW / 18.6 Mvar, Sudut 0 deg, fault ABC ...
```

Gambar 3.11 Skrip Generating Data sedang berjalan

```
[1188//1188] Loc 90%, Ron 15 Ohm, Beban 46.7 MW / 18.6 Mvar, Sudut 45 deg, TAUUT ABCG ...
selesai.
[1188/1188] Loc 90%, Ron 15 Ohm, Beban 46.7 MW / 18.6 Mvar, Sudut 90 deg, fault ABCG ...
selesai.

=====
Alhamdulillah selesai! 1188 set data generated.
Total waktu: 8162.00 s.
```

Gambar 3.12 Skrip selesai dijalankan

Gambar 3.11 dan 2.12 menunjukkan tampilan ketika skrip sedang berjalan hingga selesai. Setelah proses selesai maka akan muncul beberapa *item* di *workspace*.

Workspace	
Name	Value
dataset_fitur	1188×8 double
dataset_label	1188×1 cell

Gambar 3.133 Item yang muncul di *Workspace*

Di antaranya yaitu dua matriks utama dengan nama dataset_fitur dan dataset_label yang ditunjukkan pada gambar 3.13.

3.3.3 Ekstraksi Fitur dan Pembentukan Dataset

Sebelum diproses lebih lanjut, data mentah hasil simulasi perlu diubah menjadi sebuah set fitur yang ringkas sekaligus informatif. Proses ini sangat krusial karena meskipun lengkap, data mentah tidak efisien dan tidak optimal untuk digunakan secara langsung dalam pelatihan model Jaringan Saraf Tiruan (JST).

1. Analisis Data Mentah Hasil Simulasi

Setiap eksekusi simulasi menghasilkan sebuah objek *simout* yang di dalamnya terdapat variabel *mlp_input*. Data sinyal mentah tersimpan dalam *simout.mlp_input.signals.values* berupa sebuah matriks numerik berjenis *double* dengan dimensi **10041 baris × 7 kolom**. Setiap kolom pada matriks ini merepresentasikan satu sinyal pengukuran (*Vabc*, *Iabc*, dan *I0*), dan setiap barisnya merupakan seluruh pembacaan nilai sinyal tersebut pada satu titik waktu diskrit.

	simout	simout.mlp_input	dataset_label	simout.mlp_input.signals.values			
	10041x7 double						
	1	2	3	4	5	6	7
5922	-3.9563e+04	-6.4428e+04	1.0399e+05	-4.0747e+03	-2.2380e+03	6.3127e+03	4.1678e-11
5923	-3.8031e+04	-6.5721e+04	1.0375e+05	-3.9967e+03	-2.3319e+03	6.3286e+03	4.1762e-11
5924	-3.6490e+04	-6.6999e+04	1.0349e+05	-3.9177e+03	-2.4252e+03	6.3429e+03	4.2533e-11
5925	-3.4939e+04	-6.8260e+04	1.0320e+05	-3.8377e+03	-2.5180e+03	6.3557e+03	4.2125e-11
5926	-3.3380e+04	-6.9505e+04	1.0288e+05	-3.7567e+03	-2.6101e+03	6.3668e+03	4.1820e-11
5927	-3.1812e+04	-7.0732e+04	1.0254e+05	-3.6748e+03	-2.7016e+03	6.3764e+03	4.0831e-11
5928	-3.0237e+04	-7.1942e+04	1.0218e+05	-3.5921e+03	-2.7924e+03	6.3845e+03	4.1131e-11
5929	-2.8654e+04	-7.3134e+04	1.0179e+05	-3.5084e+03	-2.8826e+03	6.3910e+03	4.4365e-11
5930	-2.7065e+04	-7.4307e+04	1.0137e+05	-3.4239e+03	-2.9720e+03	6.3958e+03	4.3093e-11
5931	-2.5468e+04	-7.5463e+04	1.0093e+05	-3.3385e+03	-3.0607e+03	6.3991e+03	4.3553e-11
5932	-2.3865e+04	-7.6600e+04	1.0047e+05	-3.2523e+03	-3.1486e+03	6.4009e+03	4.4163e-11
5933	-2.2256e+04	-7.7718e+04	9.9974e+04	-3.1653e+03	-3.2357e+03	6.4010e+03	4.4575e-11
5934	-2.0642e+04	-7.8817e+04	9.9459e+04	-3.0775e+03	-3.3221e+03	6.3996e+03	4.4569e-11
5935	-1.9023e+04	-7.9896e+04	9.8919e+04	-2.9890e+03	-3.4076e+03	6.3966e+03	4.5475e-11
5936	-1.7399e+04	-8.0956e+04	9.8355e+04	-2.8997e+03	-3.4923e+03	6.3920e+03	4.4596e-11
5937	-1.5771e+04	-8.1996e+04	9.7767e+04	-2.8097e+03	-3.5762e+03	6.3858e+03	4.4160e-11
5938	-1.4139e+04	-8.3015e+04	9.7154e+04	-2.7190e+03	-3.6591e+03	6.3781e+03	4.4119e-11
5939	-1.2503e+04	-8.4014e+04	9.6517e+04	-2.6276e+03	-3.7411e+03	6.3688e+03	4.3558e-11
5940	-1.0864e+04	-8.4993e+04	9.5857e+04	-2.5356e+03	-3.8223e+03	6.3579e+03	4.6341e-11
5941	-9.2230e+03	-8.5950e+04	9.5173e+04	-2.4430e+03	-3.9025e+03	6.3455e+03	4.4237e-11
5942	-7.5793e+03	-8.6886e+04	9.4465e+04	-2.3498e+03	-3.9817e+03	6.3314e+03	4.4861e-11
5943	-5.9336e+03	-8.7801e+04	9.3734e+04	-2.2559e+03	-4.0599e+03	6.3159e+03	4.4609e-11
5944	-4.2866e+03	-8.8694e+04	9.2980e+04	-2.1616e+03	-4.1371e+03	6.2987e+03	4.4431e-11
5945	-2.6387e+03	-8.9565e+04	9.2204e+04	-2.0667e+03	-4.2134e+03	6.2800e+03	4.3977e-11
5946	-989.9395	-9.0414e+04	9.1404e+04	-1.9713e+03	-4.2885e+03	6.2598e+03	4.2909e-11
5947	659.1531	-9.1241e+04	9.0582e+04	-1.8754e+03	-4.3626e+03	6.2380e+03	4.1810e-11
5948	2.3078e+03	-9.2045e+04	8.9737e+04	-1.7790e+03	-4.4357e+03	6.2147e+03	4.0806e-11
5949	3.9560e+03	-9.2826e+04	8.8870e+04	-1.6822e+03	-4.5076e+03	6.1893e+03	4.0577e-11
5950	5.6034e+03	-9.3585e+04	8.7982e+04	-1.5850e+03	-4.5785e+03	6.1634e+03	4.1414e-11
5951	7.2493e+03	-9.4321e+04	8.7071e+04	-1.4874e+03	-4.6482e+03	6.1355e+03	4.2767e-11
5952	8.8932e+03	-9.5033e+04	8.6140e+04	-1.3894e+03	-4.7167e+03	6.1061e+03	4.2646e-11
5953	1.0535e+04	-9.5721e+04	8.5186e+04	-1.2911e+03	-4.7841e+03	6.0752e+03	4.1944e-11
5954	1.2175e+04	-9.6387e+04	8.4212e+04	-1.1924e+03	-4.8503e+03	6.0427e+03	4.1751e-11
5955	1.3811e+04	-9.7028e+04	8.3217e+04	-1.0935e+03	-4.9153e+03	6.0088e+03	4.2426e-11
5956	1.5443e+04	-9.7646e+04	8.2202e+04	-994.2905	-4.9791e+03	5.9734e+03	4.2580e-11
5957	1.7073e+04	-9.8239e+04	8.1166e+04	-894.8430	-5.0417e+03	5.9366e+03	4.0814e-11

Gambar 3.14 Contoh cuplikan data mentah hasil simulasi

Data seperti pada Gambar 3.14 tidak dapat digunakan secara langsung untuk pelatihan model JST karena beberapa alasan fundamental:

- Dimensionalitas yang Sangat Tinggi:** 10041 adalah banyaknya seluruh titik pengukuran selama simulasi berjalan, baik sebelum, saat, maupun setelah terjadi gangguan. Menggunakan 10041 titik data untuk setiap sinyal sebagai

input akan menciptakan model dengan jumlah parameter yang terlalu besar dan memerlukan data dalam jumlah jutaan untuk dilatih dan sangat rentan terhadap *overfitting*.

- b. **Informasi yang Berulang (*Redundant*):** Sebagian besar titik data dalam sinyal (terutama sebelum dan jauh setelah gangguan) tidak mengandung informasi yang relevan untuk klasifikasi.
- c. **Banyaknya *noise* dan kurang representatif:** Bentuk gelombang mentah sangat sensitif terhadap *noise*. Oleh karena itu diperlukan data yang lebih stabil untuk merepresentasikan masing-masing bentuk gangguan secara jelas.

2. Ekstraksi Fitur Energi Menggunakan *Discrete Wavelet Transform (DWT)*

Untuk mengatasi masalah di atas, dilakukan proses ekstraksi fitur untuk mereduksi dimensi dan menangkap esensi dari sinyal gangguan dengan tahapan berikut:

- a. **Mengambil Jendela Transien:** Langkah pertama adalah mengisolasi bagian sinyal tertentu yang paling kaya akan informasi, untuk kemudian diproses menjadi bentuk informasi yang lebih representatif. Dalam hal ini digunakan sinyal saat transien gangguan terjadi. Digunakan jendela waktu (*time window*) sepanjang 0.02 detik (setara dengan satu siklus penuh pada frekuensi 50 Hz) yang dimulai tepat pada saat gangguan terjadi (*fault_inception_time* = 0.2 detik) hingga 0.22 detik. Pemilihan jendela satu siklus ini bertujuan untuk menangkap seluruh fenomena transien awal sebagaimana telah dibahas pada bab 2 tentang *dc offset* dan analisis transien.
- b. **Melakukan Transformasi *Wavelet*:** Pada setiap sinyal di dalam jendela transien tersebut dilakukan transformasi *wavelet* menggunakan *mother wavelet Daubechies 4 (db4)*. Sebagaimana telah dibahas pada Bab 2, *Discrete Wavelet Transform (DWT)* sangat efektif dalam menganalisis sinyal non-stasioner seperti sinyal gangguan karena kemampuannya memisahkan komponen frekuensi tinggi (yang merepresentasikan transien) dari frekuensi rendah. Proses ini diimplementasikan dalam skrip sebagai berikut:

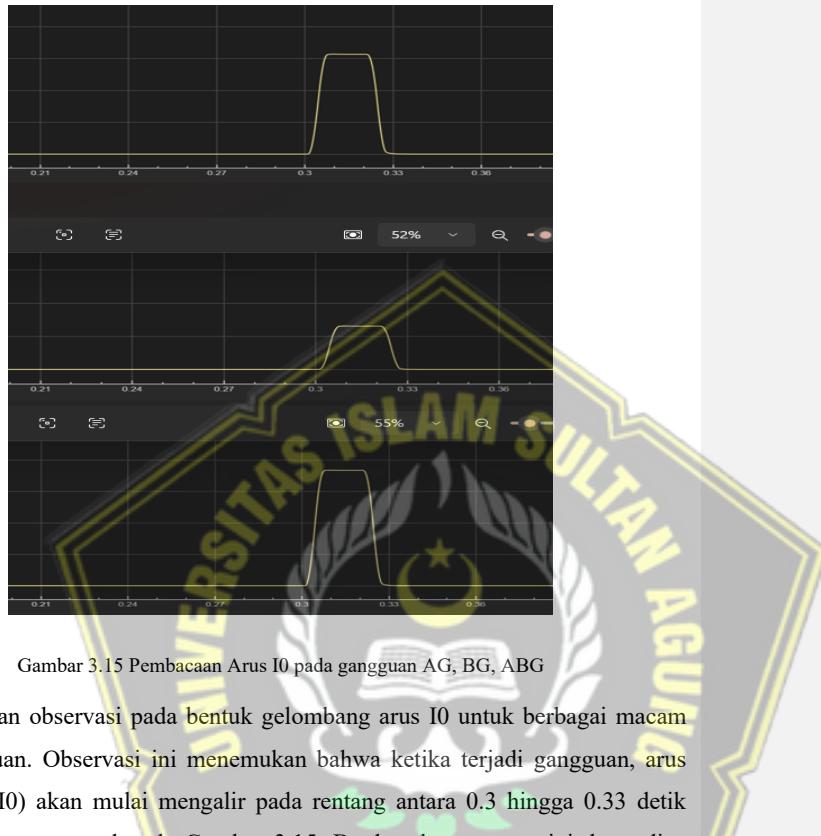
```
% Mengambil jendela transien dari data mentah
start_index = find(simout.tout >= fault_inception_time, 1, 'first');
end_index = find(simout.tout >= (fault_inception_time + window_duration), 1, 'first');
jendela_transien = data_mentah(start_index:end_index, :);

% Loop untuk setiap sinyal (Vabc, Iabc, I0)
for sinyal_idx = 1:size(jendela_transien, 2)
    satu_sinyal = jendela_transien(:, sinyal_idx);
    % Menerapkan DWT dan mengambil koefisien detail (cD)
    [~, cD] = dwt(satu_sinyal, 'db4');
    % Menghitung energi dari koefisien detail
    energi = sum(cD.^2);
    baris_fitur_dwt(sinyal_idx) = energi;
end
```

Hasil dari *DWT* adalah koefisien detail (*cD*) yang merepresentasikan komponen frekuensi tinggi dari sinyal. Sebagaimana telah dipaparkan pada bagian 2.2.3.3 tentang Aplikasi *DWT*, nilai ini akan diubah menjadi satu nilai **energi** dari koefisien tersebut. Energi didapatkan dengan menjumlahkan kuadrat dari koefisien detail ($\text{sum}(cD.^2)$) menggunakan persamaan (2.7). Proses ini menghasilkan **7 fitur energi** untuk setiap sampel data.

3. Ekstraksi Fitur Tambahan Arus Urutan Nol (I0)

Untuk memperkuat kemampuan klasifikasi model pada gangguan yang melibatkan tanah, ditambahkan satu fitur yang mengambil data dari pembacaan arus I0.



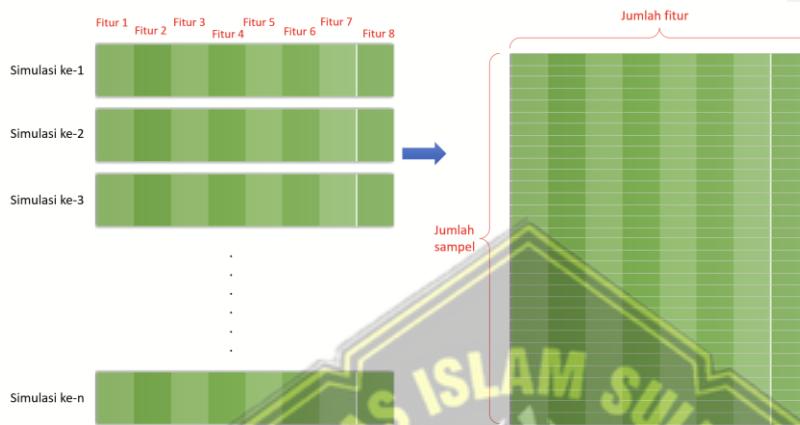
Gambar 3.15 Pembacaan Arus I0 pada gangguan AG, BG, ABG

Dilakukan observasi pada bentuk gelombang arus I0 untuk berbagai macam jenis gangguan. Observasi ini menemukan bahwa ketika terjadi gangguan, arus urutan nol (I0) akan mulai mengalir pada rentang antara 0.3 hingga 0.33 detik seperti beberapa sampel pada Gambar 3.15. Berdasarkan temuan ini, kemudian diambil jendela waktu dalam rentang waktu tersebut.

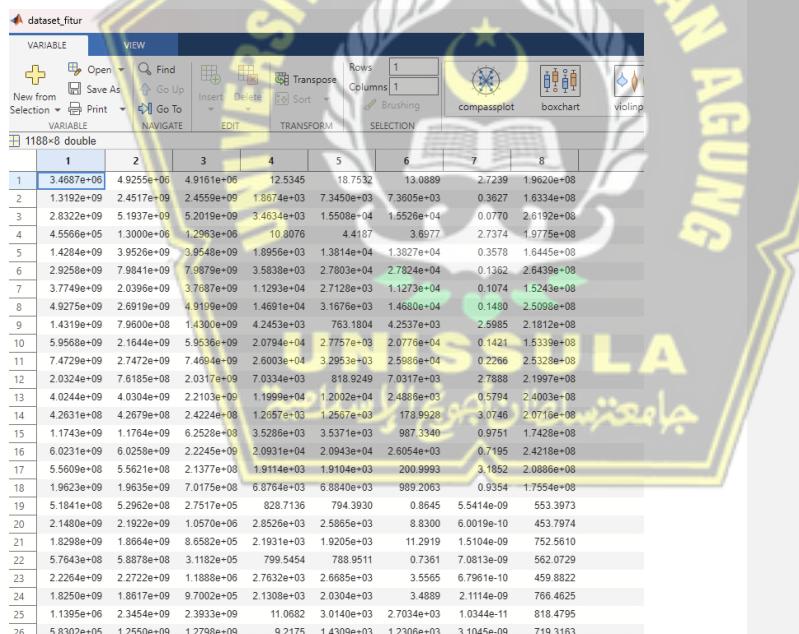
Fitur ini dihitung dengan menjumlahkan total amplitudo dari sinyal I0 dalam jendela waktu yang telah ditentukan. Selanjutnya, nilai total ini dikalikan dengan faktor 100. Penskalaan ini bertujuan untuk mengamplifikasi nilai fitur terutama pada kasus ketika arus I0 sangat kecil (mendekati nol). Dengan demikian, perbedaan nilai antara kondisi "ada arus I0" dan "tidak ada arus I0" menjadi lebih signifikan. Perhitungan ini diimplementasikan dalam cuplikan program berikut:

$$\text{sumI0} = \text{sum}(\text{abs}(\text{sinyali0})) * 100;$$

4. Pembentukan Dataset Final



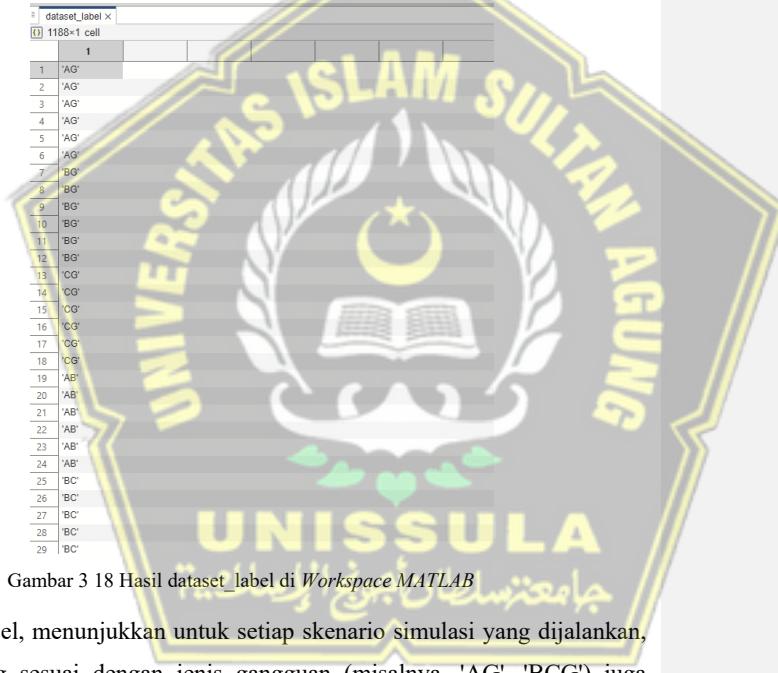
Gambar 3.16 Ilustrasi pembentukan dataset fitur



Gambar 3.177 Hasil dataset_fitur di Workspace MATLAB

Setiap kali simulasi selesai dijalankan, delapan fitur yang telah diekstraksi (tujuh fitur energi + satu fitur jumlah I0) disusun menjadi satu baris data baru. Ilustrasi pada Gambar 3.16 menunjukkan bahwa proses ini diulangi untuk seluruh 1188 skenario simulasi gangguan atau hingga simulasi ke-n. Kemudian hasilnya diakumulasi menjadi sebuah matriks *dataset_fitur* dengan dimensi 1188×8 . Setiap baris dalam matriks ini merepresentasikan satu sampel data dari setiap simulasi, dan setiap kolom merepresentasikan satu fitur digunakan model dalam proses pelatihan.

Hasil akhir proses ini ditunjukkan pada Gambar 3.17.



Secara paralel, menunjukkan untuk setiap skenario simulasi yang dijalankan, label kelas yang sesuai dengan jenis gangguan (misalnya, 'AG', 'BCG') juga ditambahkan. Akumulasi dari seluruh 1188 simulasi ini kemudian membentuk sebuah matriks *dataset_label* di *workspace MATLAB* dengan dimensi 1188×1 seperti pada cuplikan pada Gambar 3.18. Data ini berfungsi sebagai *target* atau "kunci jawaban" yang bersesuaian langsung dengan setiap baris pada *dataset_fitur* dan akan digunakan oleh model selama proses pelatihan.

3.3.4 Persiapan Data untuk Pelatihan Model

Sebelum dapat digunakan untuk melatih model, data harus disesuaikan dengan *environment* yang dibutuhkan di *MATLAB*. Tahap ini mencakup dua proses utama: encoding label target dan transposisi matriks data.

Model *Multi-Layer Perceptron* tidak dapat memproses data target yang masih dalam format teks (misalnya, 'AG', 'BCG'). Oleh karena itu, format ini diubah menjadi numerik menggunakan teknik *One Hot Encoding*. Proses ini mengubah setiap label kelas menjadi sebuah vektor biner yang panjangnya sama dengan jumlah total kelas (11). Vektor ini akan berisi nilai 0 di semua posisi kecuali pada satu indeks yang merepresentasikan kelas tersebut akan bernilai 1.

Sebagai contoh, label 'AG' akan dipetakan ke indeks 1, kemudian menghasilkan bentuk One Hot [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]. Proses ini dieksekusi dengan cuplikan program berikut:

```
for i = 1:jumlah_sampel
    label = dataset_label{i};
    label_nomor = 0;
    % indexing dari 1 - 11
    switch label
        case 'AG'
            label_nomor = 1;
        case 'BG'
            label_nomor = 2;
    end
    % (dan seterusnya hingga indeks 11)
    if label_nomor > 0
        one_hot = zeros(1, jumlah_kelas);
        one_hot(label_nomor) = 1;
        dataset_label_encoded(i, :) = one_hot;
    end
end
```

Fungsi pelatihan *MATLAB* mensyaratkan data masukan harus dalam format [*jumlah fitur* × *jumlah sampel*] dan data target [*jumlah kelas* × *jumlah sampel*].

Oleh karena itu dilakukan transposisi matrik sesuai dengan cuplikan program berikut:

```
Y = dataset_label_encoded';
```

```
X = dataset_fitur';
```

Hasil dari proses ini adalah dua matriks final yang sudah sesuai dengan dimensi yang diperlukan untuk pelatihan model:

- X: Matriks input dengan dimensi 8×1188 .
- Y: Matriks target dengan dimensi 11×1188 .

3.3.5 Pelatihan Model *Multi-Layer Perceptron*

Setelah data yang diperlukan sudah siap dalam format yang sesuai, tahap selanjutnya adalah proses pelatihan dan validasi model *Multi-Layer Perceptron* (*MLP*). Tahapan ini mencakup pembagian dataset, konfigurasi arsitektur dan parameter pelatihan, serta evaluasi kinerja model yang sudah terlatih.

1. Pembagian Dataset (Latih, Validasi, dan Uji)

Sesuai dengan praktik standar pelatihan model kecerdasan buatan yang telah diuraikan pada Bab 2, dataset dibagi menjadi tiga bagian yaitu: 70% untuk data latih (*Training Set*), 15% untuk data validasi (*Validation Set*), dan 15% untuk data uji (*Test Set*).

Proses pembagian acak ini diimplementasikan menggunakan fungsi *divideParam* sesuai dengan cuplikan program berikut:

```
% split data
net.divideParam.trainRatio = 0.70;
net.divideParam.valRatio = 0.15;
net.divideParam.testRatio = 0.15;
```

2. Konfigurasi Arsitektur Jaringan dan Parameter Pelatihan

Jaringan *Multi-Layer Perceptron* dalam penelitian ini dibangun menggunakan perintah `net = feedforwardnet()`. Arsitektur yang digunakan adalah sebagai berikut:

- a. *Input Layer*: Jumlah neuron pada lapisan ini ditentukan berdasarkan dimensi matriks input X. Dalam penelitian ini, *input layer* akan memiliki 8 neuron.
- b. *Hidden Layers*: Digunakan dua lapisan tersembunyi (*hidden layers*) pada jaringan ini. Lapisan pertama memiliki 22 neuron, dan lapisan tersembunyi kedua memiliki 10 neuron.
- c. *Output Layer*: Sama seperti *input layer*, jumlah neuron pada lapisan ini juga ditentukan berdasarkan dimensi matriks target Y. Dalam penelitian ini, digunakan 11 neuron.

Adapun beberapa parameter dan algoritma untuk pelatihan model menggunakan pengaturan *default* dari *Deep Learning Toolbox*, sebagai berikut:

- a. Fungsi Aktivasi: *Hyperbolic Tangent Sigmoid* (*tansig*) untuk *hidden layer*, dan *Linear* (*purelin*) untuk *output layer*.
- b. Algoritma Pelatihan: *Levenberg-Marquardt* (*trainlm*).
- c. Fungsi Kinerja (*Loss Function*): *Mean Squared Error* (*MSE*)
- d. Jumlah Epoch Maksimal: 1000 epoch.

```

subobjects:
    input: Equivalent to inputs{1}
    output: Equivalent to outputs{3}

    inputs: {1x1 cell array of 1 input}
    layers: {3x1 cell array of 3 layers}
    outputs: {1x3 cell array of 1 output}
    biases: {3x1 cell array of 3 biases}
    inputWeights: {3x1 cell array of 1 weight}
    layerWeights: {3x3 cell array of 2 weights}

functions:
    adaptFcn: 'adaptwb'
    adaptParam: {none}
    derivFcn: 'defaultderiv'
    divideFcn: 'dividerand'
    divideParam: '.trainRatio', '.valRatio', '.testRatio'
    divideMode: 'sample'
    initFcn: 'initlay'
    performFcn: 'mse'
    performParam: 'regularization', 'normalization'
    plotFcns: {'plotperform', 'plottrainstate', 'ploterrhist',
        'plotregression'}
    plotParams: {1x4 cell array of 4 params}
    trainFcn: 'trainlm'
    trainParam: '.showWindow', '.showCommandLine', '.show',
        '.epochs',
        '.time', '.goal', '.min_grad', '.max_fail', '.mu', '.mu_dec,
        '.mu_inc', '.mu_max'

weight and bias values:
    lW: {3x1 cell} containing 1 input weight matrix
    LW: {3x3 cell} containing 2 layer weight matrices
    b: {3x1 cell} containing 3 bias vectors

```

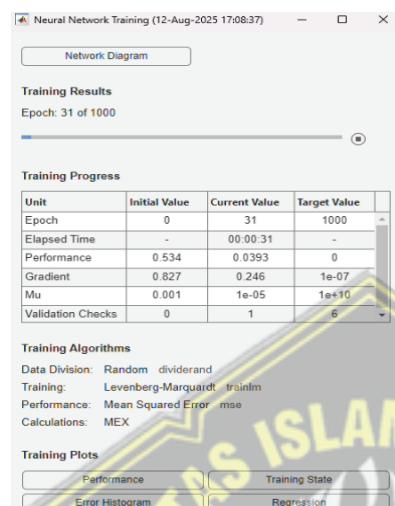
Gambar 3.19 Beberapa parameter, fungsi dan algoritma yang digunakan jaringan

Beberapa parameter, fungsi dan algoritma yang digunakan jaringan ditunjukkan pada Gambar 3.19.

3. Proses Pelatihan dan Validasi Model

Selanjutnya, proses pelatihan model diinisiasi menggunakan fungsi *train* dari MATLAB.

```
% Inisiasi proses pelatihan
[net, tr] = train(net, X, Y);
```



Gambar 3.20 Proses pelatihan model sedang berjalan

Fungsi ini menjalankan proses pelatihan secara iteratif yang berjalan *epoch-by-epoch*. Dalam setiap epoch, model akan secara aktif belajar dengan menggunakan data latih. *Error* yang dihitung dari data ini digunakan oleh algoritma *backpropagation* untuk memperbarui dan mengoptimalkan seluruh bobot sinaptik (*synaptic weight*) di dalam jaringan. Proses ini bertujuan meminimalkan kesalahan prediksi pada data yang sedang dipelajari. Pelatihan yang sedang berjalan ditunjukkan pada Gambar 3.20.

Setelah bobot diperbarui, performa model dievaluasi menggunakan data validasi. *Error* dari data ini tidak digunakan untuk mengubah bobot, melainkan hanya untuk memonitor kemampuan generalisasi. Jika performa pada data validasi tidak mengalami peningkatan atau mulai menurun, mekanisme *early stopping* akan berjalan. Hal ini bertujuan untuk mencegah overfitting dan menyimpan versi model dengan performa validasi terbaik. Fungsi *train* menghasilkan dua output penting yang tersimpan di *workspace MATLAB*:

- a. net: Objek jaringan *Multi-Layer Perceptron* yang sudah terlatih dengan semua bobot sinaptik final yang telah dioptimalkan.

- b. tr: Sebuah training record yang berisi semua informasi historis dari proses pelatihan

net.x			
1x1 network			
Property	Value	Size	Class
version	'9'	1x1	char
name	'Feed-Forward Neur...'	1x27	char
efficiency	1x1 struct	1x1	struct
userdata	1x1 struct	1x1	struct
numInputs	1	1x1	double
numLayers	3	1x1	double
numOutputs	1	1x1	double
numInputDelays	0	1x1	double
numLayerDelays	0	1x1	double
numFeedbackDelays	0	1x1	double
numWeightElements	511	1x1	double
sampleTime	1	1x1	double
biasConnect	3x1 logical	3x1	logical
inputConnect	3x1 logical	3x1	logical
layerConnect	3x3 logical	3x3	logical
outputConnect	1x3 logical	1x3	logical
inputs	1x1 cell	1x1	cell
layers	3x1 cell	3x1	cell
biases	3x1 cell	3x1	cell
outputs	1x3 cell	1x3	cell
inputWeights	3x1 cell	3x1	cell
layerWeights	3x3 cell	3x3	cell
adaptFcn	'adaptwb'	1x7	char
adaptParam	1x1 nnetParam	1x1	nnetParam
divideFcn	'dividerand'	1x10	char
divideParam	1x1 nnetParam	1x1	nnetParam
divideMode	'sample'	1x6	char
initFcn	'initlay'	1x7	char
performFcn	'mse'	1x3	char
performParam	1x1 nnetParam	1x1	nnetParam
plotFcn	1x4 cell	1x4	cell
plotParams	1x4 cell	1x4	cell
derivFcn	'defaultderiv'	1x12	char
trainFcn	'trainlm'	1x7	char
trainParam	1x1 nnetParam	1x1	nnetParam
LMW	2x4 cell	2x4	cell

Gambar 3.21 Hasil dari Jaringan yang telah dilatih dengan bias dan weight yang optimal

Gambar 3.22 hasil dari Jaringan yang telah dilatih dengan bias dan weight yang telah disesuaikan dengan target prediksi dari model.

Selanjutnya, dilakukan tahap evaluasi menggunakan data uji yang belum pernah dilihat oleh model. Pertama, data uji diekstraksi dari matriks X dan Y menggunakan indeks yang tersimpan di *tr.testInd*. Kemudian, model *net* yang telah terlatih digunakan untuk membuat prediksi pada data uji tersebut. Proses ini diimplementasikan sesuai cuplikan program berikut:

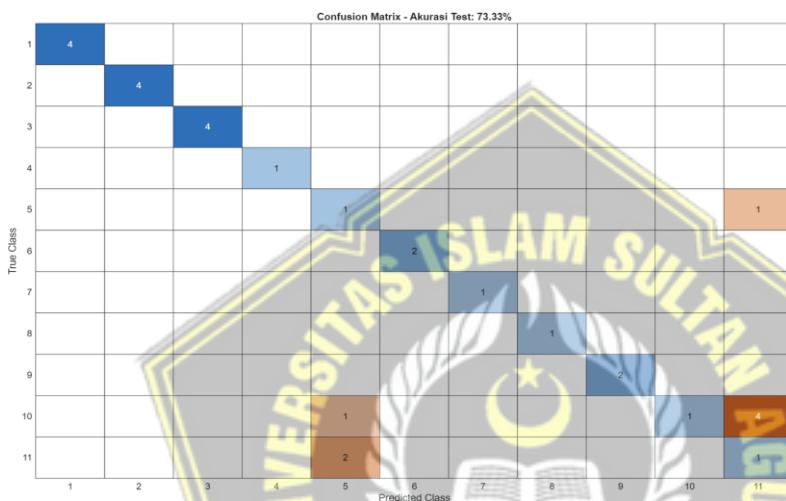
```
% Ekstraksi data uji dan pembuatan prediksi
```

```
idxTest = tr.testInd;
XTest = X(:, idxTest);
YTest_onehot = Y(:, idxTest);
```

% Melatih jaringan yang sudah terlatih dengan data uji

```
Y_pred_onehot = net(XTest);
```

4. Evaluasi Kinerja Model



Gambar 3.23 Contoh Confusion Matrix

Untuk mengukur performa model secara kuantitatif digunakan akurasi, yang dihitung dari *Confusion Matrix* seperti pada Gambar 3.22. *Confusion Matrix* adalah sebuah tabel yang memvisualisasikan kinerja model dengan membandingkan antara kelas aktual dari data uji (*True Class*) dan kelas yang diprediksi oleh model (*Predicted Class*). Diagonal utama dari matriks ini menunjukkan jumlah prediksi yang benar, sementara elemen di luar diagonal menunjukkan kesalahan klasifikasi yang dilakukan oleh model.

Adapun akurasi mengukur persentase total prediksi yang benar dari keseluruhan data uji. Akurasi dihitung dengan rumus pada persamaan (2.3) .

$$\text{Akurasi} = \frac{\text{Jumlah Prediksi Benar}}{\text{Total Jumlah Data Uji}} \times 100\%$$

Akurasi ini akan menjadi tolok ukur utama untuk menentukan keberhasilan model dalam mengklasifikasikan jenis gangguan secara tepat. Proses evaluasi kinerja model ini diimplementasikan menggunakan cuplikan program berikut:

```
% Hitung akurasi  
correct_predictions = sum(Y_pred == YTest_labels);  
total_samples = numel(YTest_labels);  
accuracy = (correct_predictions / total_samples) * 100;  
% Plot confusion matrix  
confusionchart(YTest_cat_from_labels, Y_pred_cat);
```

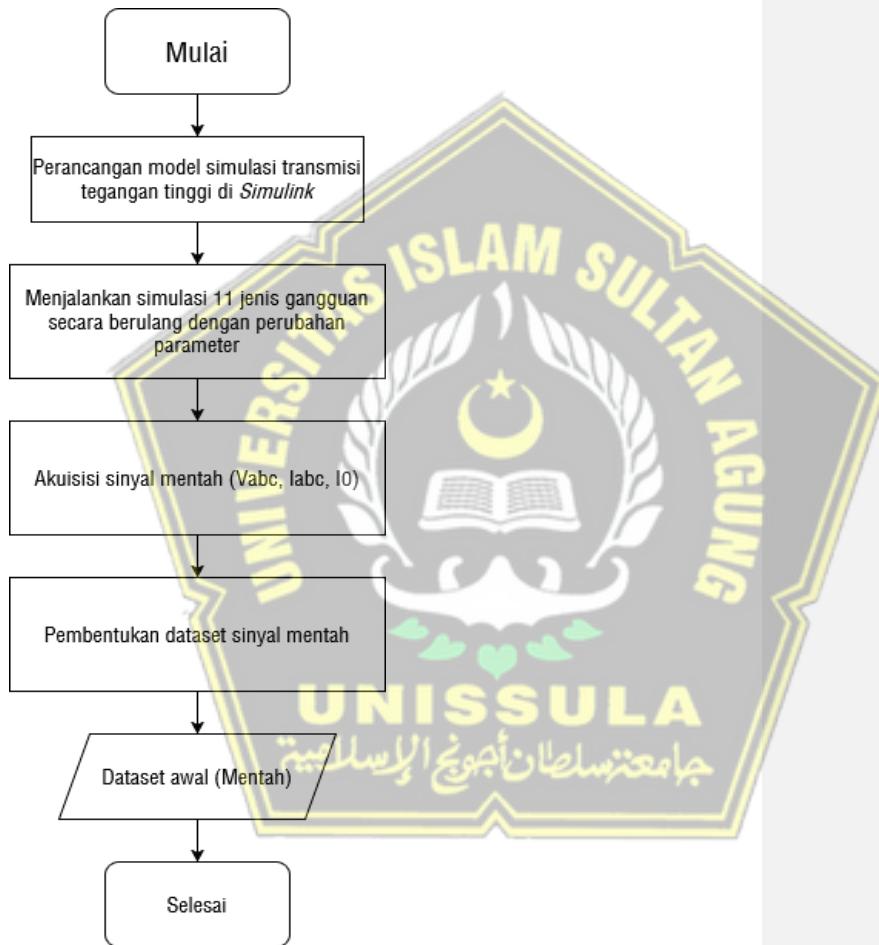


Commented [ik5]: Menambahkan narasi untuk flow chart

3.4 Diagram Alir Penelitian

Penelitian dilaksanakan dalam beberapa tahapan yang ditunjukkan pada tiga diagram alir berikut:

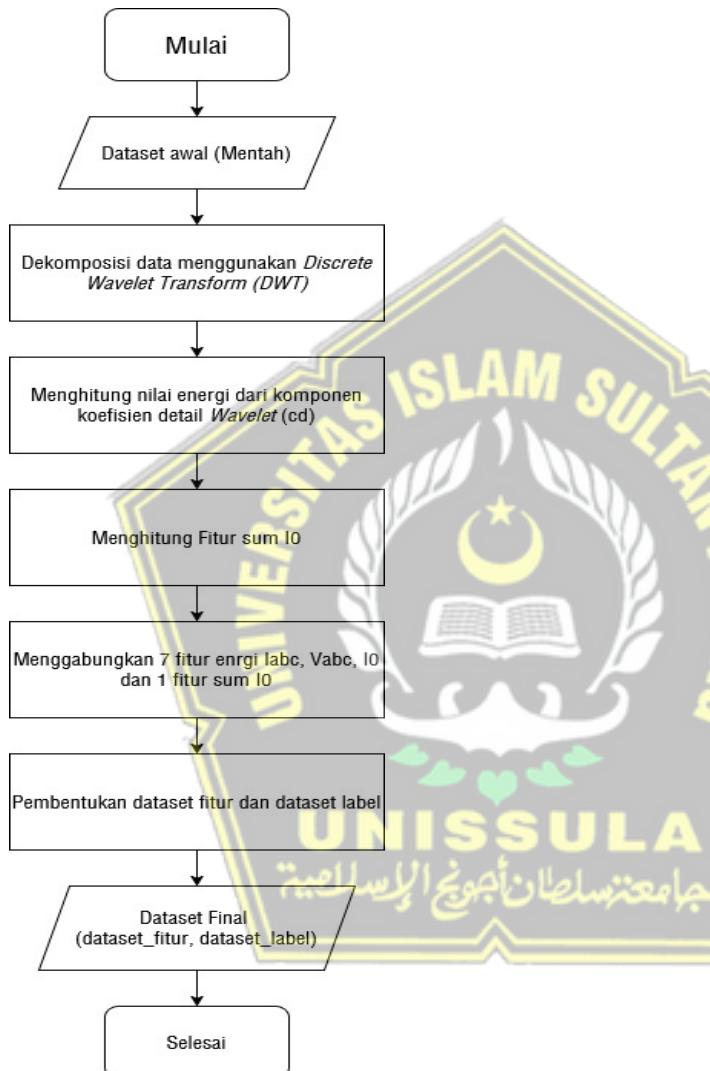
- Pembuatan Simulasi dan Pembentukan Dataset Awal



Gambar 3.24 Diagram alir pembentukan dataset mentah

Tahap ini menghasilkan output berupa dataset awal yang akan diproses lebih lanjut.

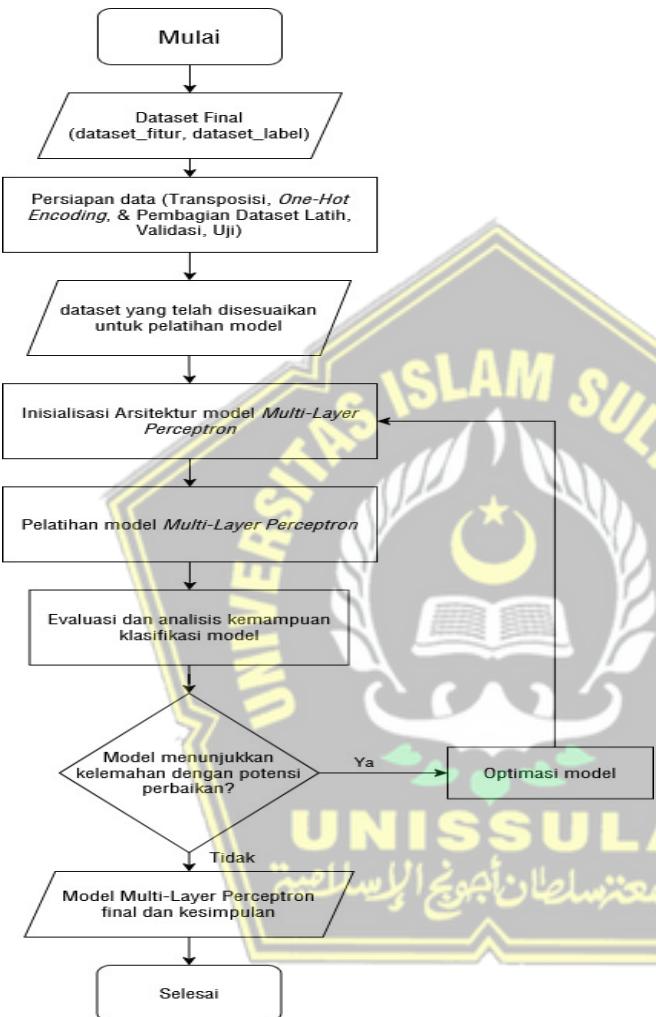
b. Ekstraksi Fitur dan Pemrosesan Data



Gambar 3.24 Diagram alir ekstraksi fitur

Data mentah dari tahap sebelumnya diproses menggunakan energi wavelet untuk menghasilkan output akhir berupa dua buah dataset final

c. Pelatihan, Evaluasi, dan Optimasi Model



Gambar 3.25 Diagram alir pelatihan model *MLP*

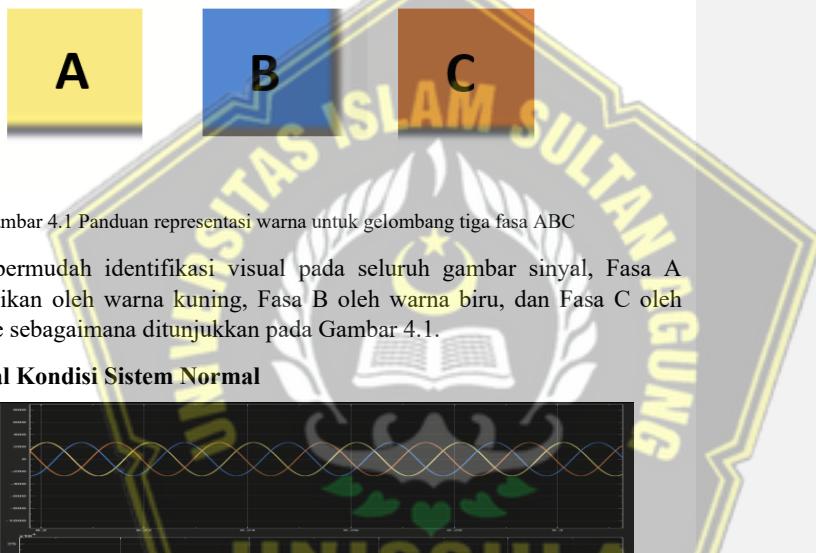
Pada tahap ini, dataset final dari proses sebelumnya digunakan lebih lanjut untuk melatih model Multi-Layer Perceptron hingga menghasilkan model yang optimal.

BAB IV

HASIL DAN ANALISIS

4.1 Hasil Simulasi

Pada sub-bab ini, disajikan hasil dari beberapa sampel simulasi sistem tenaga yang telah dirancang sesuai dengan metodologi pada Bab III. Hasil yang ditampilkan adalah bentuk gelombang sinyal mentah dari tegangan (V_{abc}), arus (I_{abc}), dan arus residual (I_0) untuk berbagai skenario.



Gambar 4.1 Panduan representasi warna untuk gelombang tiga fasa ABC

Untuk mempermudah identifikasi visual pada seluruh gambar sinyal, Fasa A direpresentasikan oleh warna kuning, Fasa B oleh warna biru, dan Fasa C oleh warna oranye sebagaimana ditunjukkan pada Gambar 4.1.

4.1.1 Sinyal Kondisi Sistem Normal

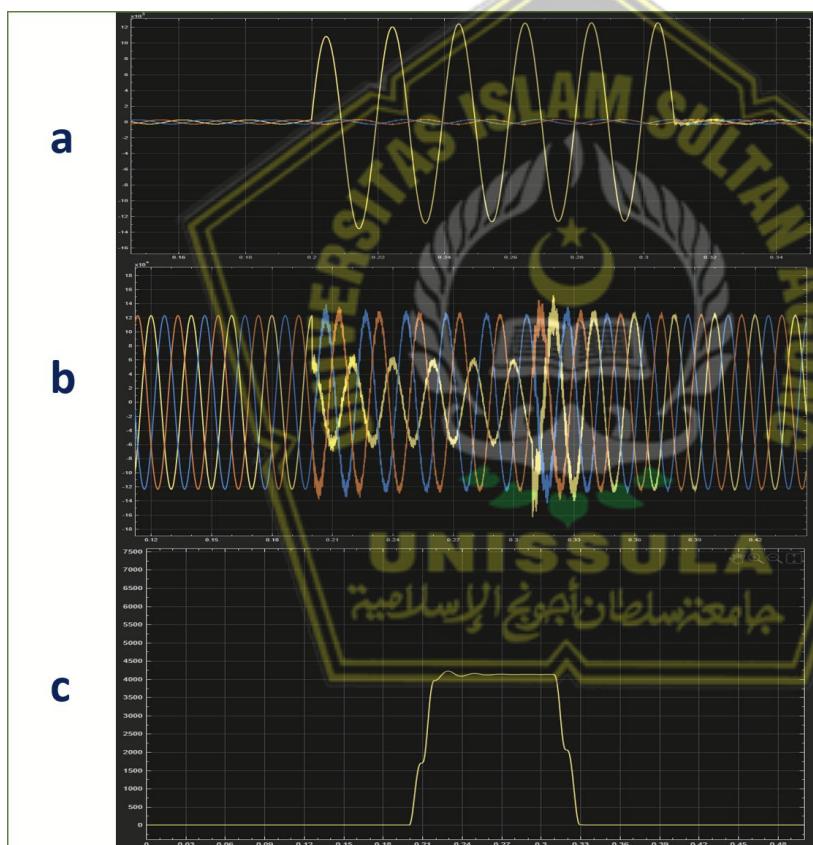


Gambar 4.2 Sinyal Kondisi Sistem Normal (a) Sinyal Arus; (b) Sinyal Tegangan; (c) Sinyal Arus I_0

Analisis sinyal pada Gambar 4.2 menunjukkan sistem yang beroperasi dalam kondisi normal dan seimbang. Sinyal tegangan dan arus tiga fasa menunjukkan bentuk gelombang sinusoidal yang sempurna, dengan magnitudo yang identik dan pergeseran fasa sebesar 120 derajat antara satu sama lain. Karakteristik dari kondisi sistem normal ini divalidasi oleh sinyal arus I_0 yang bernilai nol. Hal ini mengkonfirmasi bahwa tidak ada ketidakseimbangan atau jalur arus bocor ke tanah pada sistem.

4.1.2 Sinyal Gangguan Satu Fasa ke Tanah

- Gangguan Fasa A ke Tanah (A-G)



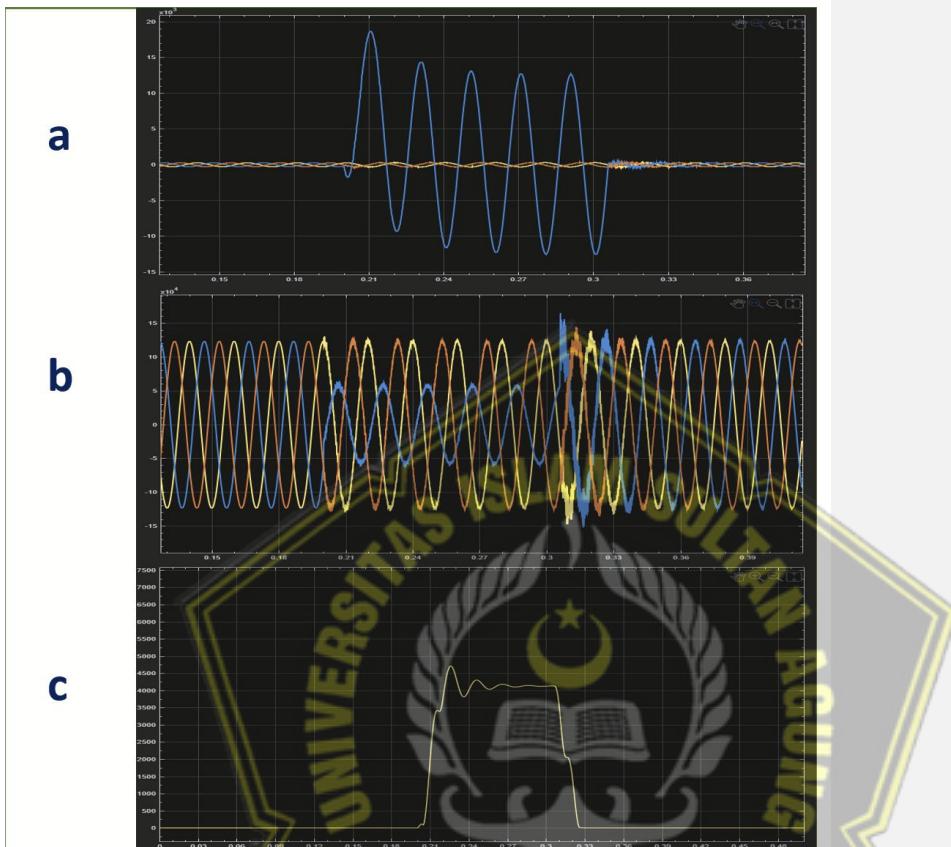
Gambar 4.3 Sinyal Gangguan Fasa A-G (a) Sinyal Arus; (b) Sinyal Tegangan; (c) Sinyal Arus I_0

Analisis sinyal tegangan pada Gambar 4.3 menunjukkan sistem yang awalnya beroperasi dalam kondisi seimbang dengan ketiga fasa memiliki magnitudo yang identik. Saat gangguan terjadi, terjadi penurunan tegangan yang sangat drastis pada Fasa A. Fenomena ini merupakan karakteristik dari hubung singkat yaitu impedansi yang sangat rendah pada titik gangguan menyebabkan tegangan pada fasa tersebut jatuh. Sementara itu, fasa B dan C tetap mengalami distorsi sebagai akibat dari ketidakseimbangan yang terjadi pada sistem.

Sejalan dengan jatuhnya tegangan, sinyal arus pada fasa A menunjukkan lonjakan magnitudo yang sangat besar dan disertai dengan distorsi transien. Fenomena ketidakseimbangan ini juga divalidasi oleh adanya aliran arus I_0 . Hal ini mengkonfirmasi bahwa gangguan yang terjadi bersifat asimetris dan memiliki jalur hubung singkat ke tanah.

- b. Gangguan Fasa B ke Tanah (B-G)





Gambar 4.4 Sinyal Gangguan Fasa B-G (a) Sinyal Arus; (b) Sinyal Tegangan; (c) Sinyal Arus I0

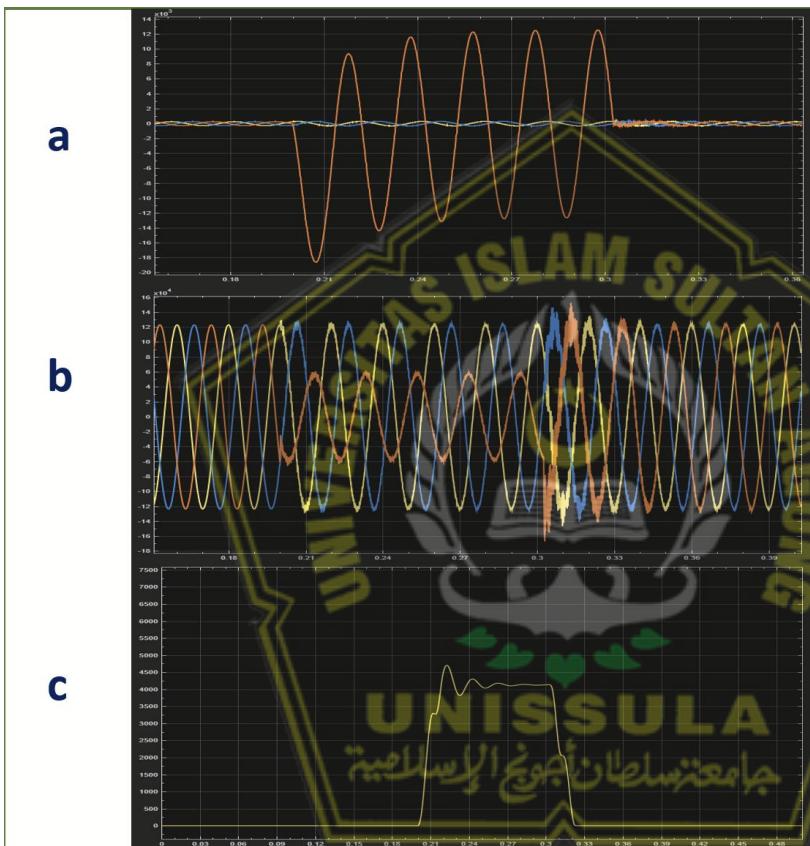
UNISSULA

Analisis sinyal tegangan pada Gambar 4.4 menunjukkan sistem yang awalnya beroperasi dalam kondisi normal dan seimbang. Ketika gangguan terjadi, sistem mengalami jatuh tegangan (*voltage sag*) yang signifikan pada Fasa B. Fenomena ini merupakan karakteristik dari hubung singkat yaitu impedansi yang rendah pada titik gangguan menyebabkan tegangan pada fasa tersebut jatuh. Sebagai dampaknya, fasa A dan C juga ikut mengalami distorsi akibat ketidakseimbangan sistem.

Bersamaan dengan penurunan tegangan tersebut, sinyal arus pada Fasa B mengalami lonjakan magnitudo yang masif dan disertai distorsi transien. Validasi bahwa gangguan ini melibatkan tanah diperkuat oleh munculnya aliran arus I0 yang

sebelumnya bernilai nol. Kombinasi fenomena ini mengkonfirmasi bahwa gangguan yang terjadi bersifat asimetris dan memiliki jalur hubung singkat ke tanah.

c. Gangguan Fasa C ke Tanah (C-G)



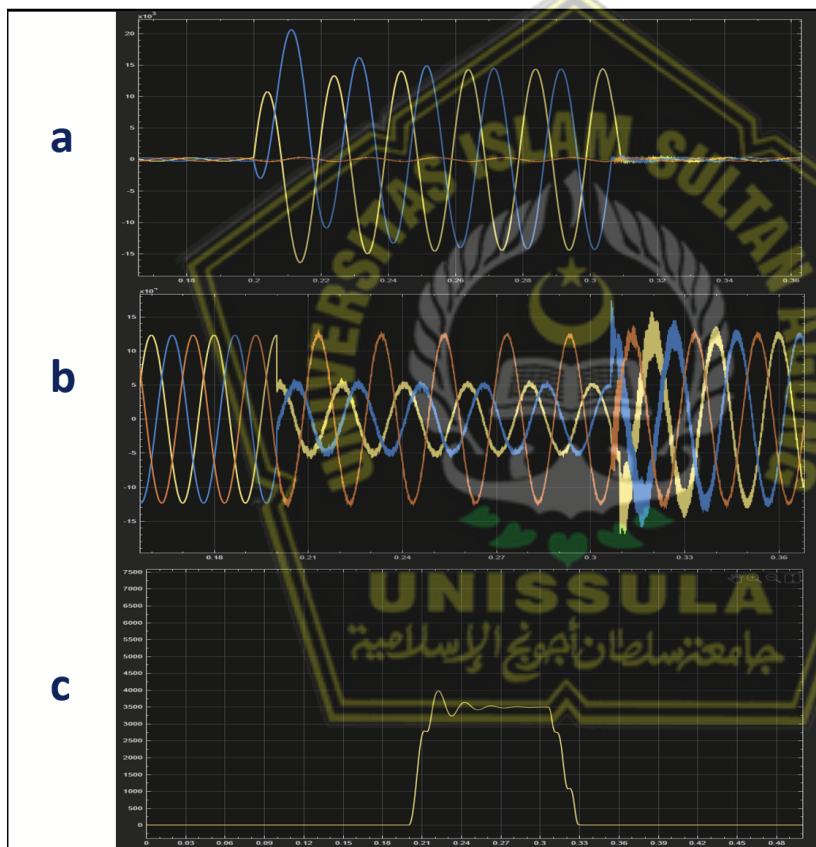
Gambar 4.5 Sinyal Gangguan Fasa B-C (a) Sinyal Arus; (b) Sinyal Tegangan; (c) Sinyal Arus I0

Analisis sinyal tegangan pada Gambar 4.5 menunjukkan sistem yang pada awalnya berjalan seimbang. Saat gangguan terjadi, Fasa C secara spesifik mengalami jatuh tegangan yang drastis. Penurunan tegangan ini adalah ciri khas dari gangguan hubung singkat, di mana impedansi rendah pada titik gangguan menyebabkan tegangan pada fasa tersebut turun secara signifikan. Sementara itu, ketidakseimbangan ini juga menyebabkan distorsi pada fasa A dan B.

Sebagai konsekuensinya, sinyal arus pada Fasa C menunjukkan lonjakan magnitudo yang sangat besar disertai distorsi transien. Fenomena ketidakseimbangan ini divalidasi oleh kemunculan arus I_0 , yang secara definitif mengkonfirmasi bahwa gangguan ini bersifat asimetris dan memiliki jalur hubung singkat ke tanah.

4.1.3 Sinyal Gangguan Dua Fasa ke Tanah

a. Gangguan Fasa AB ke Tanah (A-B-G)



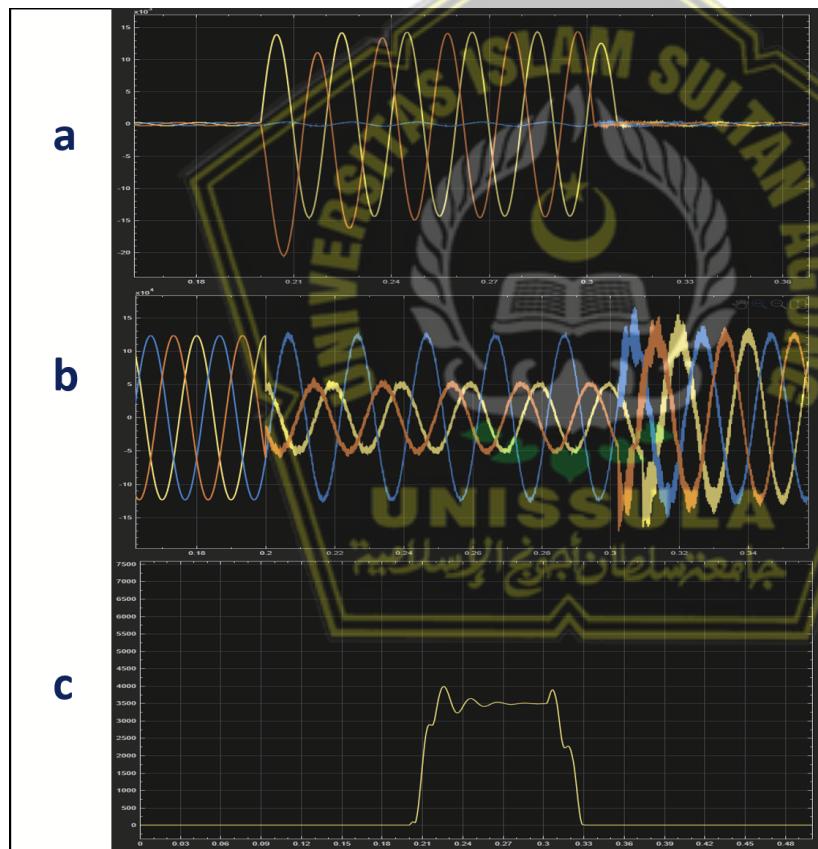
Gambar 4.6 Sinyal Gangguan Fasa A-B-G (a) Sinyal Arus; (b) Sinyal Tegangan; (c) Sinyal Arus I_0

Analisis sinyal tegangan pada Gambar 4.6 menunjukkan sistem yang pada awalnya beroperasi dalam kondisi seimbang. Ketika gangguan terjadi, Fasa A dan

Fasa B secara bersamaan mengalami penurunan tegangan yang sangat drastis. Fenomena ini merupakan karakteristik hubung singkat, di mana impedansi yang sangat rendah pada titik gangguan menyebabkan tegangan pada kedua fasa tersebut jatuh. Sementara itu, Fasa C mengalami distorsi akibat ketidakseimbangan yang terjadi pada sistem.

Sejalan dengan jatuhnya tegangan, sinyal arus pada Fasa A dan Fasa B menunjukkan lonjakan magnitudo yang sangat besar, disertai dengan distorsi transien. Adanya aliran arus I_0 yang signifikan juga memvalidasi bahwa gangguan yang terjadi melibatkan hubung singkat ke tanah.

b. Gangguan Fasa AC ke Tanah (A-C-G)



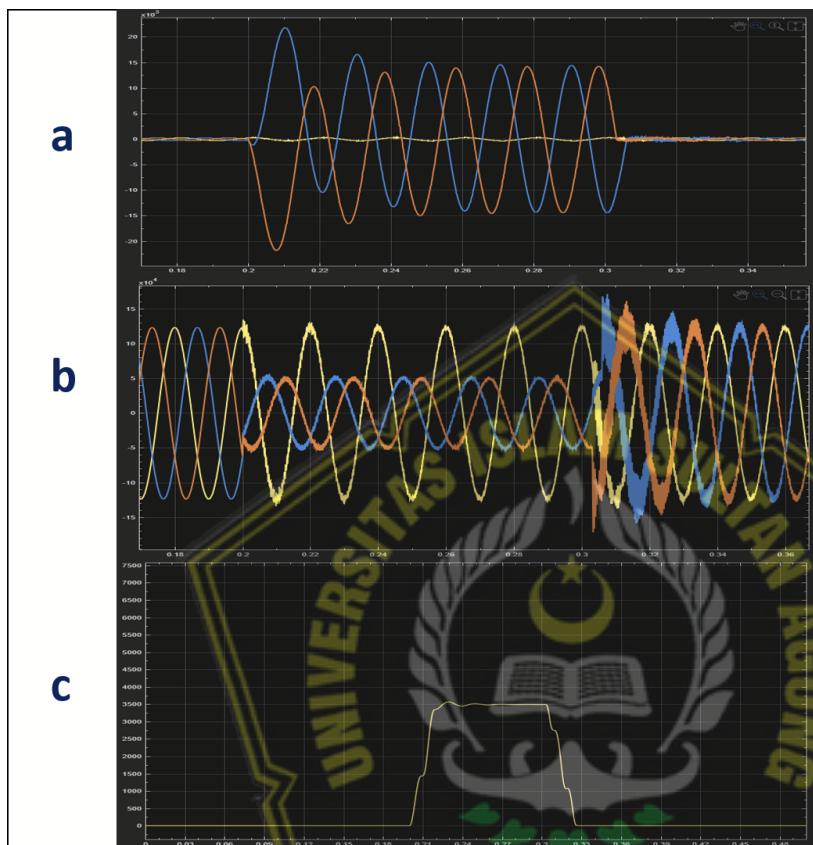
Gambar 4.7 Sinyal Gangguan Fasa A-C-G (a) Sinyal Arus; (b) Sinyal Tegangan; (c) Sinyal Arus I_0

Analisis sinyal tegangan pada Gambar 4.7 menunjukkan sistem yang pada awalnya beroperasi dalam kondisi seimbang. Ketika gangguan terjadi, Fasa A dan Fasa C secara bersamaan mengalami penurunan tegangan yang sangat drastis. Fenomena ini merupakan karakteristik hubung singkat, di mana impedansi yang sangat rendah pada titik gangguan menyebabkan tegangan pada kedua fasa tersebut jatuh. Sementara itu, Fasa B mengalami distorsi akibat ketidakseimbangan yang terjadi pada sistem.

Sejalan dengan jatuhnya tegangan, sinyal arus pada Fasa A dan Fasa C menunjukkan lonjakan magnitudo yang sangat besar, disertai dengan distorsi transien. Adanya aliran arus I_0 yang signifikan juga memvalidasi bahwa gangguan yang terjadi melibatkan hubung singkat ke tanah.



c. Gangguan Fasa BC ke Tanah (B-C-G)



Gambar 4.8 Sinyal Gangguan Fasa B-C-G (a) Sinyal Arus; (b) Sinyal Tegangan; (c) Sinyal Arus

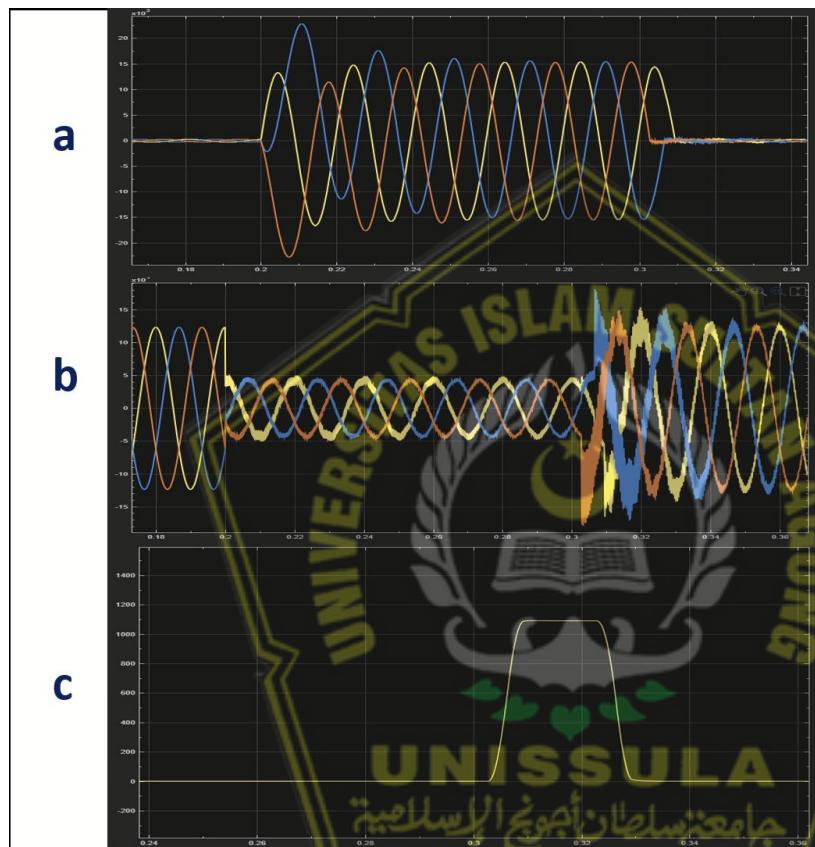
10

Analisis sinyal tegangan pada Gambar 4.8 menunjukkan sistem yang pada awalnya beroperasi dalam kondisi seimbang. Ketika gangguan terjadi, Fasa B dan Fasa C secara bersamaan mengalami penurunan tegangan yang sangat drastis. Fenomena ini merupakan karakteristik hubung singkat, di mana impedansi yang sangat rendah pada titik gangguan menyebabkan tegangan pada kedua fasa tersebut jatuh. Selain itu, Fasa A yang tidak langsung terhubung dengan gangguan juga mengalami distorsi akibat ketidakseimbangan yang terjadi pada sistem.

Sejalan dengan jatuhnya tegangan, sinyal arus pada Fasa A dan Fasa B menunjukkan lonjakan magnitudo yang sangat besar, disertai dengan distorsi

transien. Adanya aliran arus I_0 yang signifikan juga memvalidasi bahwa gangguan yang terjadi melibatkan hubung singkat ke tanah.

4.1.4 Sinyal Gangguan Tiga Fasa ke Tanah (A-B-C-G)



Gambar 4.9 Sinyal Gangguan Fasa A-B-C-G (a) Sinyal Arus; (b) Sinyal Tegangan; (c) Sinyal Arus I_0

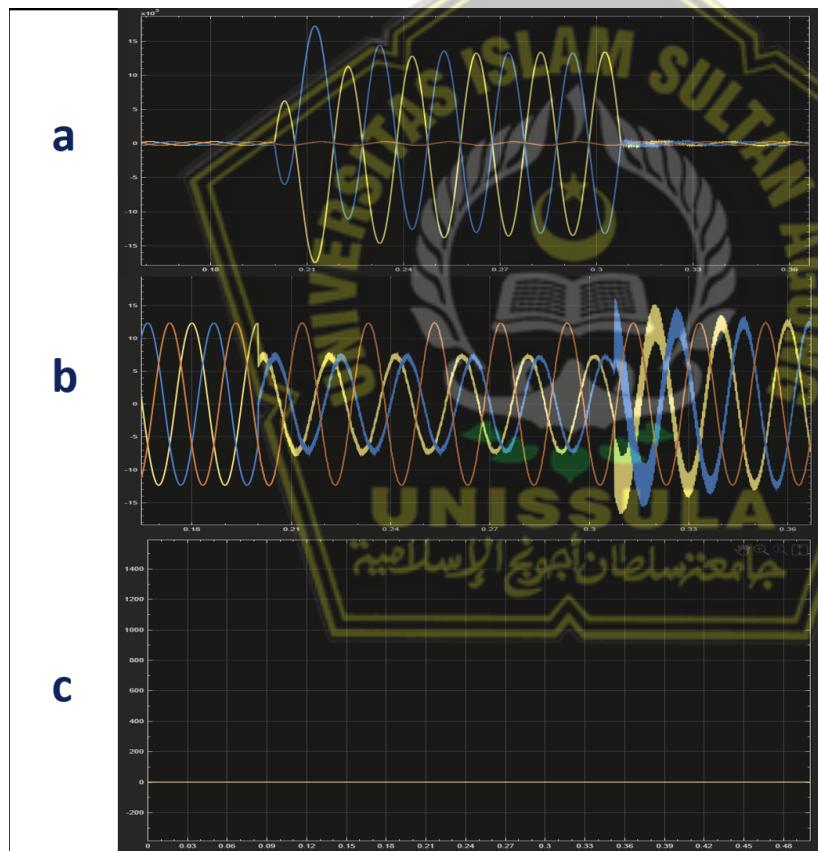
Analisis sinyal tegangan pada Gambar 4.9 menunjukkan sistem yang awalnya beroperasi dalam kondisi seimbang. Saat gangguan terjadi, ketiga fasa (A, B, dan C) secara serentak mengalami penurunan magnitudo tegangan yang signifikan. Fenomena ini merupakan ciri khas dari gangguan simetris, di mana hubung singkat

terjadi pada seluruh fasa secara bersamaan sehingga sistem yang terganggu masih berada pada kondisi yang relatif seimbang.

Sejalan dengan penurunan tegangan, sinyal arus pada ketiga fasa menunjukkan lonjakan magnitudo yang sangat besar. Pembeda utama dari gangguan ini adalah adanya aliran arus I_0 residual meski magnitudonya tidak sebesar gangguan asimetris lainnya. Hal ini menunjukkan bahwa fenomena yang terjadi adalah gangguan tiga fasa ke tanah.

4.1.5 Sinyal Gangguan Fasa ke Fasa

a. Gangguan Fasa A-B

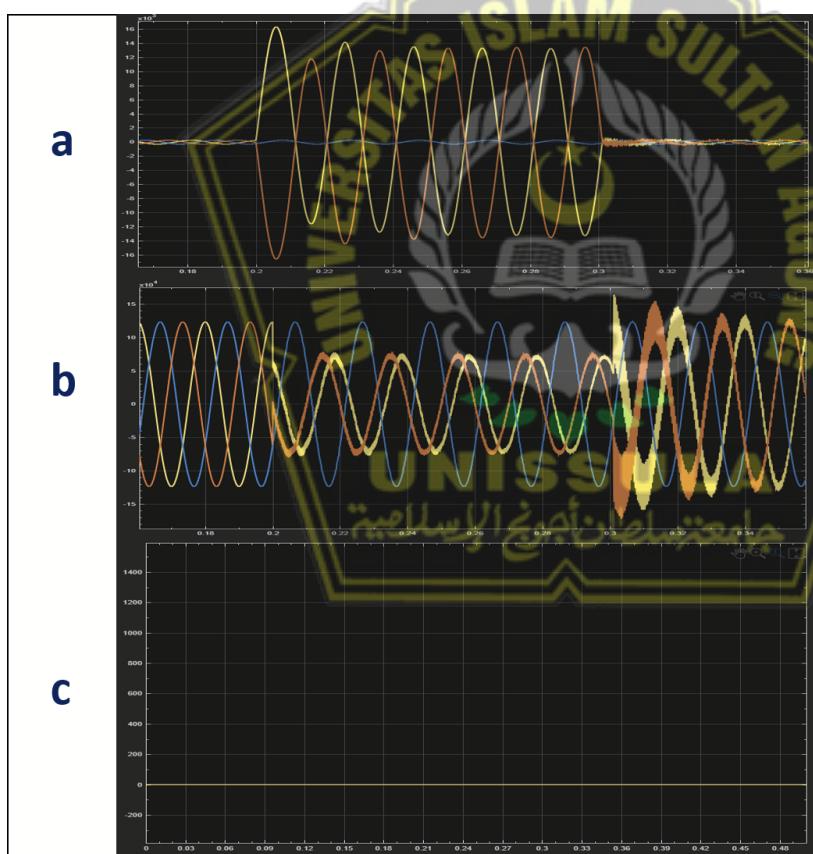


Gambar 4.10 Sinyal Gangguan Fasa A-B (a) Sinyal Arus; (b) Sinyal Tegangan; (c) Sinyal Arus I_0

Analisis sinyal tegangan pada Gambar 4.10 menunjukkan sistem yang pada awalnya beroperasi dalam kondisi seimbang. Saat gangguan terjadi, Fasa A dan Fasa B mengalami penurunan tegangan yang signifikan. Sementara itu, Fasa C yang tidak terlibat langsung dalam gangguan, cenderung mempertahankan magnitudo tegangannya meskipun turut mengalami distorsi akibat ketidakseimbangan sistem.

Sejalan dengan perubahan tegangan, sinyal arus pada Fasa A dan Fasa B menunjukkan lonjakan magnitudo yang sangat besar. Karakteristik paling penting dari gangguan ini adalah tidak adanya aliran arus I_0 yang mengkonfirmasi bahwa gangguan terjadi karena hubungan antar fasa dan tidak memiliki jalur langsung ke tanah.

b. Gangguan Fasa A-C



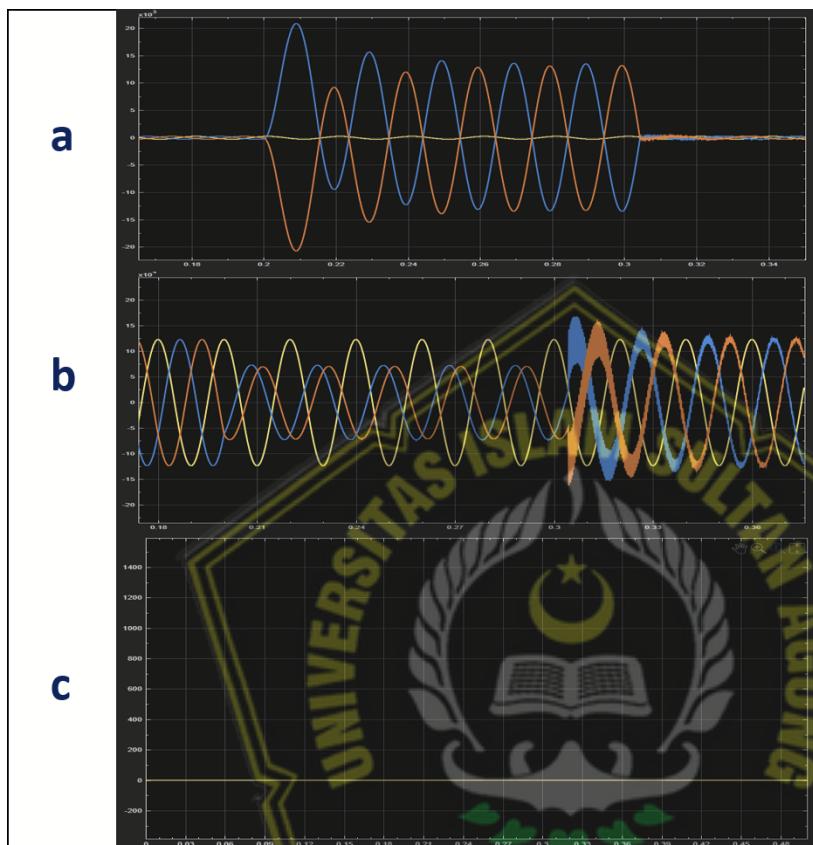
Gambar 4.11 Sinyal Gangguan Fasa A-C (a) Sinyal Arus; (b) Sinyal Tegangan; (c) Sinyal Arus I_0

Analisis sinyal tegangan pada Gambar 4.11 menunjukkan sistem yang pada awalnya beroperasi dalam kondisi seimbang. Saat gangguan terjadi, Fasa A dan Fasa C mengalami penurunan tegangan yang signifikan. Sementara itu, Fasa B yang tidak terlibat langsung dalam gangguan, cenderung mempertahankan magnitudo tegangannya meskipun turut mengalami distorsi akibat ketidakseimbangan sistem.

Sejalan dengan perubahan tegangan, sinyal arus pada Fasa A dan Fasa C menunjukkan lonjakan magnitudo yang sangat besar. Karakteristik paling penting dari gangguan ini adalah tidak adanya aliran arus I_0 yang mengkonfirmasi bahwa gangguan terjadi karena hubungan antar fasa dan tidak memiliki jalur langsung ke tanah.



c. Gangguan Fasa B-C



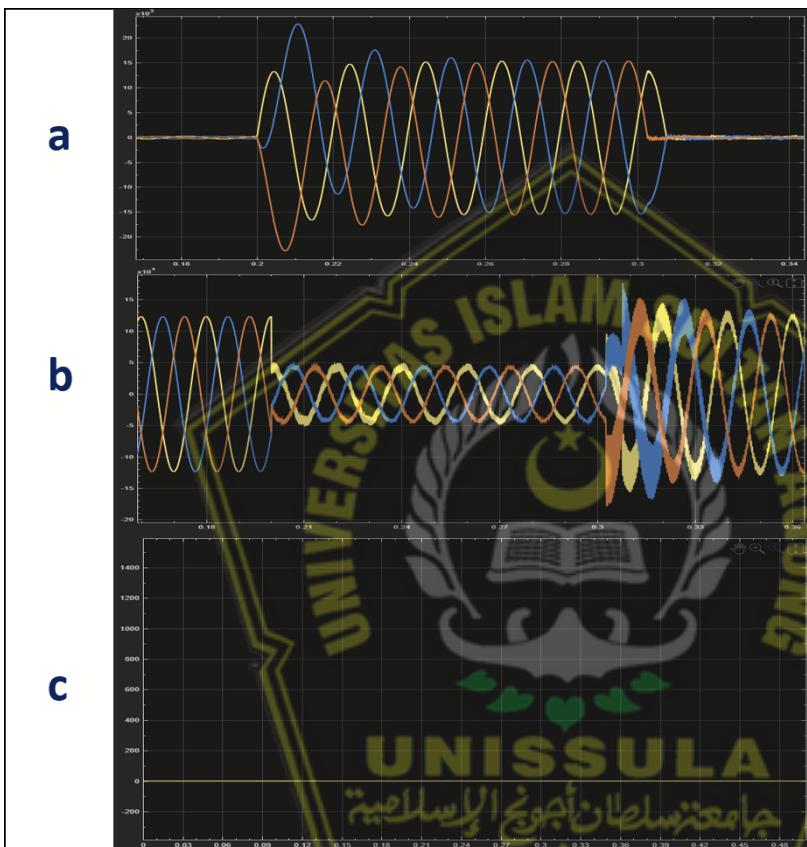
Gambar 4.12 Sinyal Gangguan Fasa B-C (a) Sinyal Arus; (b) Sinyal Tegangan; (c) Sinyal Arus I0

Analisis sinyal tegangan pada Gambar 4.12 menunjukkan sistem yang pada awalnya beroperasi dalam kondisi seimbang. Saat gangguan terjadi, Fasa B dan Fasa C mengalami penurunan tegangan yang signifikan. Sementara itu, Fasa A yang tidak terlibat langsung dalam gangguan, cenderung mempertahankan magnitudo tegangannya meskipun turut mengalami distorsi akibat ketidakseimbangan sistem.

Sejalan dengan perubahan tegangan, sinyal arus pada Fasa B dan Fasa C menunjukkan lonjakan magnitudo yang sangat besar. Karakteristik paling penting dari gangguan ini adalah tidak adanya aliran arus I0 yang mengkonfirmasi bahwa

gangguan terjadi karena hubungan antar fasa dan tidak memiliki jalur langsung ke tanah.

d. Gangguan Fasa A-B-C



Gambar 4.13 Sinyal Gangguan Fasa A-B-C (a) Sinyal Arus; (b) Sinyal Tegangan; (c) Sinyal Arus

10

Analisis sinyal tegangan pada Gambar menunjukkan sistem yang awalnya beroperasi dalam kondisi seimbang. Saat gangguan terjadi, ketiga fasa (A, B, dan C) secara serentak mengalami penurunan magnitudo tegangan yang signifikan. Meskipun terjadi penurunan, sistem tetap berada dalam kondisi simetris selama gangguan, yang merupakan ciri khas dari gangguan tiga fasa.

Sejalan dengan penurunan tegangan, sinyal arus pada ketiga fasa menunjukkan lonjakan magnitudo yang sangat besar. Karakteristik utama pada gangguan ini adalah tidak adanya aliran arus I_0 yang muncul. Hal ini secara definitif mengkonfirmasi bahwa gangguan yang terjadi bersifat simetris dan hanya melibatkan ketiga fasa tanpa hubungan langsung ke tanah.

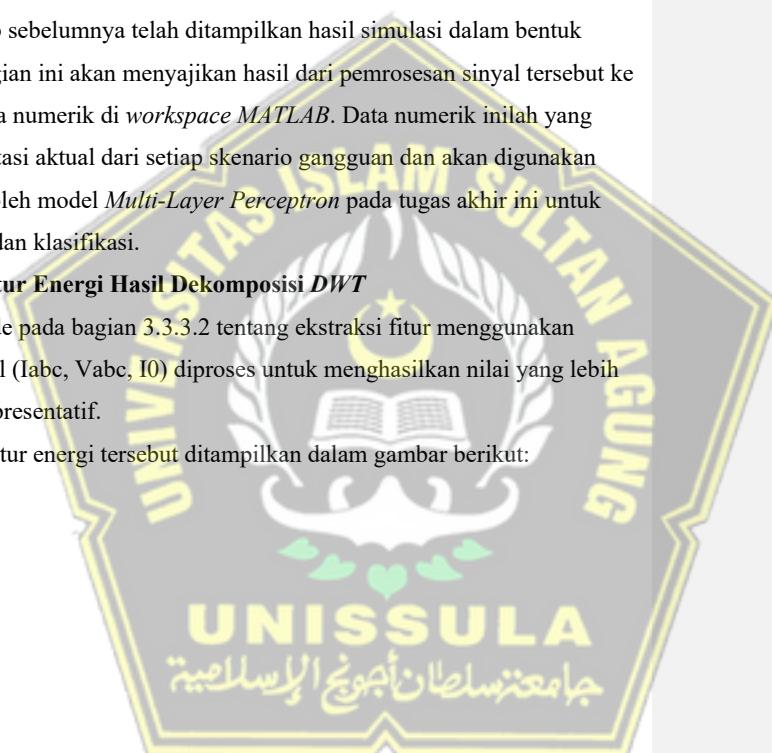
4.2 Hasil Dataset dan Pemrosesan Data

Pada sub-bab sebelumnya telah ditampilkan hasil simulasi dalam bentuk sinyal visual. Bagian ini akan menyajikan hasil dari pemrosesan sinyal tersebut ke dalam bentuk data numerik di *workspace MATLAB*. Data numerik inilah yang menjadi representasi aktual dari setiap skenario gangguan dan akan digunakan secara langsung oleh model *Multi-Layer Perceptron* pada tugas akhir ini untuk proses pelatihan dan klasifikasi.

4.2.1 Tujuh Fitur Energi Hasil Dekomposisi DWT

Sesuai metode pada bagian 3.3.3.2 tentang ekstraksi fitur menggunakan *DWT*, tujuh sinyal (I_{abc} , V_{abc} , I_0) diproses untuk menghasilkan nilai yang lebih sederhana dan representatif.

Hasil tujuh fitur energi tersebut ditampilkan dalam gambar berikut:



	1	2	3	4	5	6	7
1169	1.0684e+09	4.0468e+09	4.0389e+09	2.1100e+03	7.5573e+03	6.9475e+03	0.0121
1170	2.1066e+09	1.5221e+09	1.5113e+09	4.1968e+03	3.4928e+03	3.5998e+03	0.1434
1171	2.8397e+09	1.3706e+09	2.8514e+09	5.1594e+03	2.5710e+03	5.6426e+03	0.0168
1172	1.6817e+09	1.8021e+09	1.6844e+09	3.8884e+03	3.3839e+03	3.5731e+03	0.2266
1173	4.9035e+09	5.3055e+08	4.8726e+09	8.7817e+03	1.0059e+03	8.2729e+03	0.3031
1174	2.7777e+09	1.5927e+09	2.7937e+09	5.1496e+03	3.1568e+03	5.6370e+03	0.0201
1175	1.8453e+09	1.9584e+09	1.8444e+09	4.2631e+03	3.9113e+03	3.8434e+03	0.2963
1176	5.2856e+09	5.1520e+08	5.2496e+09	9.4231e+03	1.0444e+03	8.8656e+03	0.3163
1177	1.6169e+07	5.9571e+09	6.5822e+09	78.2647	1.0088e+04	1.1014e+04	1.3328e-23
1178	3.8242e+09	7.9761e+09	7.5517e+08	6.5251e+03	1.3374e+04	1.2811e+03	1.0832e-23
1179	8.3542e+09	2.4132e+09	1.7881e+09	1.4042e+04	4.0324e+03	3.1059e+03	1.5467e-23
1180	6.7684e+05	6.5117e+09	6.5613e+09	48.1386	1.1012e+04	1.0982e+04	3.3949e-24
1181	4.3292e+09	8.1457e+09	5.9876e+08	7.3646e+03	1.3660e+04	1.0175e+03	4.7948e-24
1182	8.7151e+09	2.2041e+09	2.1545e+09	1.4647e+04	3.6830e+03	3.7126e+03	7.5396e-24
1183	1.6169e+07	5.9571e+09	6.5822e+09	78.2647	1.0088e+04	1.1014e+04	1.1337e-23
1184	3.8242e+09	7.9761e+09	7.5517e+08	6.5251e+03	1.3374e+04	1.2811e+03	9.9970e-24
1185	8.3542e+09	2.4132e+09	1.7881e+09	1.4042e+04	4.0324e+03	3.1059e+03	1.2743e-23
1186	6.7684e+05	6.5117e+09	6.5613e+09	48.1386	1.1012e+04	1.0982e+04	2.8686e-24
1187	4.3292e+09	8.1457e+09	5.9876e+08	7.3646e+03	1.3660e+04	1.0175e+03	3.5877e-24
1188	8.7151e+09	2.2041e+09	2.1545e+09	1.4647e+04	3.6830e+03	3.7126e+03	4.4222e-24
1189							
1190							

Gambar 4.14 Hasil tujuh fitur energi

Kolom 1-7 pada Gambar 4.14 secara berurutan menunjukkan energi dari sinyal V_{abc} (kolom 1-3), I_{abc} (kolom 4-6), dan I_0 (kolom 7).

4.2.2 Satu Fitur Arus Tambahan Hasil dari Magnitudo Total I_0

	8
1168	5.6093e+07
1169	4.7936e+07
1170	3.9762e+07
1171	4.2979e+07
1172	6.5498e+07
1173	5.8199e+07
1174	4.5215e+07
1175	6.8986e+07
1176	6.1537e+07
1177	296.4135
1178	297.0643
1179	297.3182
1180	320.5893
1181	318.5594
1182	319.4468
1183	1.8871e+07
1184	1.8854e+07
1185	1.8885e+07
1186	1.9879e+07
1187	1.9874e+07
1188	1.9885e+07
1189	

Gambar 4.15 Hasil satu fitur tambahan total magnitudo 10

Sesuai metode pada bagian 3.3.3.3, selain diambil energi *DWT* nya, sinyal I0 juga dihitung nilai magnitudo totalnya agar menjadi fitur yang lebih diskriminatif. Hasil satu fitur tambahan tersebut ditampilkan pada gambar 4.15.

4.2.3 Dataset Fitur dan Dataset Label Final

Hasil dari tujuh fitur pada 4.2.1 dan satu fitur pada bagian 4.2.2 digabungkan menjadi satu Dataset Fitur Final yang terdiri dari total delapan fitur yang diwakili oleh masing-masing kolom seperti pada Gambar 4.16.

	1	2	3	4	5	6	7	8
1168	1.3611e+06	6.5212e+09	6.5521e+09	4.4196	1.1010e+04	1.0973e+04	0.0664	5.6093e+07
1169	1.0684e+09	4.0468e+09	4.0389e+09	2.1100e+03	7.5573e+03	6.9475e+03	0.0121	4.7936e+07
1170	2.1066e+09	1.5221e+09	1.5113e+09	4.1968e+03	3.4928e+03	3.5998e+03	0.1434	3.9762e+07
1171	2.8397e+09	1.3706e+09	2.8514e+09	5.1594e+03	2.5710e+03	5.6426e+03	0.0168	4.2979e+07
1172	1.6817e+09	1.8021e+09	1.6844e+09	3.8884e+03	3.3839e+03	3.5731e+03	0.2266	6.5498e+07
1173	4.9035e+09	5.3055e+08	4.8726e+09	8.7817e+03	1.0059e+03	8.2729e+03	0.3031	5.8199e+07
1174	2.7777e+09	1.5927e+09	2.7937e+09	5.1496e+03	3.1568e+03	5.6370e+03	0.0201	4.5215e+07
1175	1.8453e+09	1.9584e+09	1.8444e+09	4.2631e+03	3.9113e+03	3.8434e+03	0.2963	6.8986e+07
1176	5.2856e+09	5.1520e+08	5.2496e+09	9.4231e+03	1.0444e+03	8.8656e+03	0.3163	6.1537e+07
1177	1.6169e+07	5.9571e+09	6.5822e+09	78.2647	1.0088e+04	1.1014e+04	1.3328e-23	296.4135
1178	3.8242e+09	7.9761e+09	7.5517e+08	6.5251e+03	1.3374e+04	1.2811e+03	1.0832e-23	297.0643
1179	8.3542e+09	2.4132e+09	1.7881e+09	1.4042e+04	4.0324e+03	3.1059e+03	1.5467e-23	297.3182
1180	6.7684e+05	6.5117e+09	6.5613e+09	48.1386	1.1012e+04	1.0982e+04	3.3949e-24	320.5893
1181	4.3292e+09	8.1457e+09	5.9876e+08	7.3646e+03	1.3660e+04	1.0175e+03	4.7948e-24	318.5594
1182	8.7151e+09	2.2041e+09	2.1545e+09	1.4647e+04	3.6830e+03	3.7126e+03	7.5396e-24	319.4468
1183	1.6169e+07	5.9571e+09	6.5822e+09	78.2647	1.0088e+04	1.1014e+04	1.1337e-23	1.8871e+07
1184	3.8242e+09	7.9761e+09	7.5517e+08	6.5251e+03	1.3374e+04	1.2811e+03	9.9970e-24	1.8854e+07
1185	8.3542e+09	2.4132e+09	1.7881e+09	1.4042e+04	4.0324e+03	3.1059e+03	1.2743e-23	1.8885e+07
1186	6.7684e+05	6.5117e+09	6.5613e+09	48.1386	1.1012e+04	1.0982e+04	2.8686e-24	1.9879e+07
1187	4.3292e+09	8.1457e+09	5.9876e+08	7.3646e+03	1.3660e+04	1.0175e+03	3.5877e-24	1.9874e+07
1188	8.7151e+09	2.2041e+09	2.1545e+09	1.4647e+04	3.6830e+03	3.7126e+03	4.4222e-24	1.9885e+07
1189								

Gambar 4.16 Dataset Fitur Final

Di sisi lain, sebagaimana proses pada bagian 3.3.3.4 dan 3.3.4, label jenis gangguan yang sudah dikonversi kedalam format *one-hot encoding* ditampilkan pada Gambar 4.17.

UNISSULA
جامعة سلطان قابوسي الإسلامية



The screenshot shows the MATLAB interface with the title bar "dataset_label_encoded". The "VIEW" tab is selected. In the "PLOTS" section, there is a single plot labeled "dataset_label_encoded(1,1)". The main workspace displays a table titled "dataset_label_encoded" with dimensions "1188x11 double". The columns are numbered 1 through 11. The data consists of 1188 rows of binary values (0 or 1). The first few rows are:

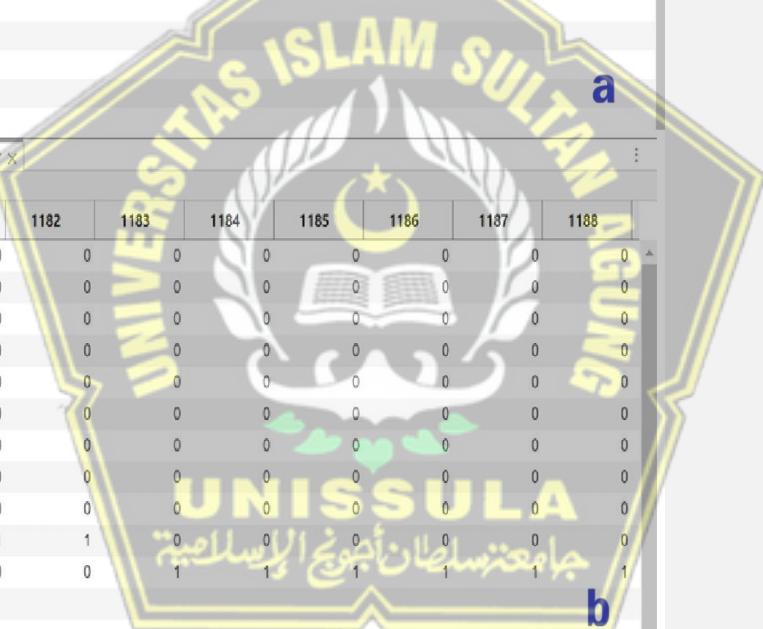
	1	2	3	4	5	6	7	8	9	10	11
1167	0	0	0	0	0	0	0	1	0	0	0
1168	0	0	0	0	0	0	0	1	0	0	0
1169	0	0	0	0	0	0	0	1	0	0	0
1170	0	0	0	0	0	0	0	1	0	0	0
1171	0	0	0	0	0	0	0	0	1	0	0
1172	0	0	0	0	0	0	0	0	1	0	0
1173	0	0	0	0	0	0	0	0	1	0	0
1174	0	0	0	0	0	0	0	0	1	0	0
1175	0	0	0	0	0	0	0	0	1	0	0
1176	0	0	0	0	0	0	0	0	1	0	0
1177	0	0	0	0	0	0	0	0	0	1	0
1178	0	0	0	0	0	0	0	0	0	1	0
1179	0	0	0	0	0	0	0	0	0	1	0
1180	0	0	0	0	0	0	0	0	0	1	0
1181	0	0	0	0	0	0	0	0	0	1	0
1182	0	0	0	0	0	0	0	0	0	1	0
1183	0	0	0	0	0	0	0	0	0	1	0
1184	0	0	0	0	0	0	0	0	0	1	0
1185	0	0	0	0	0	0	0	0	0	0	1
1186	0	0	0	0	0	0	0	0	0	0	1
1187	0	0	0	0	0	0	0	0	0	0	1
1188	0	0	0	0	0	0	0	0	0	0	1
1189											
1190											

Gambar 4.17 Dataset Label Final

Dari kedua gambar tersebut, diketahui bahwa *dataset_fitur* memiliki dimensi 1188x8, sementara *dataset_label_encoded* memiliki dimensi 1188x11. Jumlah baris menunjukkan banyaknya sampel sementara jumlah kolom merepresentasikan banyaknya fitur yang digunakan (8 fitur), dan jumlah kelas gangguan (11 gangguan).

4.2.4 Matriks Input (*X*) dan Target (*Y*) untuk Pelatihan

Sesuai metode pada bagian 3.3.4 , Matriks Input (X) dan Target/Output (Y) yang telah disesuaikan dengan kebutuhan *MATLAB* ditunjukkan oleh Gambar 4.18.



	X X	Y X	dataset_fitur X						
1	30	1181	1182	1183	1184	1185	1186	1187	1188
2	84e+05	4.3292e+09	8.7151e+09	1.6169e+07	3.8242e+09	8.3542e+09	6.7684e+05	4.3292e+09	8.7151e+09
3	17e+09	8.1457e+09	2.2041e+09	5.9571e+09	7.9761e+09	2.4132e+09	6.5117e+09	8.1457e+09	2.2041e+09
4	13e+09	5.9876e+08	2.1545e+09	6.5822e+09	7.5517e+08	1.7881e+09	6.5613e+09	5.9876e+08	2.1545e+09
5	8.1386	7.3646e+03	1.4647e+04	78.2647	6.5251e+03	1.4042e+04	48.1386	7.3646e+03	1.4647e+04
6	12e+04	1.3660e+04	3.6830e+03	1.0088e+04	1.3374e+04	4.0324e+03	1.1012e+04	1.3660e+04	3.6830e+03
7	82e+04	1.0175e+03	3.7126e+03	1.1014e+04	1.2811e+03	3.1059e+03	1.0982e+04	1.0175e+03	3.7126e+03
8	49e-24	4.7948e-24	7.5396e-24	1.1337e-23	9.9970e-24	1.2743e-23	2.8686e-24	3.5877e-24	4.4222e-24
9	0.5893	318.5594	319.4468	1.8871e+07	1.8854e+07	1.8885e+07	1.9879e+07	1.9874e+07	1.9885e+07
10									
11									
12									
13									

	X X	Y X	dataset_fitur X						
1	30	1181	1182	1183	1184	1185	1186	1187	1188
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0
10	1	1	1	0	0	0	0	0	0
11	0	0	0	1	1	1	1	1	1
12									
13									

Gambar 4.18 (a) Hasil Matriks Input (X); (b) Matriks Target (Y)

Hasil akhir data yang telah disesuaikan yaitu matriks input (X) memiliki dimensi 8x1188 dan matriks target (Y) memiliki dimensi 11x1188

4.3 Kinerja Model dan Optimasi

Pada sub-bab ini, akan disajikan secara bertahap hasil dari model dasar, analisis kelemahannya, hingga rangkaian proses optimasi yang dilakukan untuk meningkatkan akurasi dan ketepatan kinerja model dalam melakukan klasifikasi.

Sebagaimana langkah pada bagian 3.3.4 tentang penyesuaian data untuk pelatihan, setiap jenis gangguan direpresentasikan oleh label numerik. Konversi dari label jenis gangguan yang digunakan di seluruh sub-bab ini adalah sebagai berikut:

- Kelas 1: AG
- Kelas 2: BG
- Kelas 3: CG
- Kelas 4: AB
- Kelas 5: BC
- Kelas 6: CA
- Kelas 7: ABG
- Kelas 8: BCG
- Kelas 9: CAG
- Kelas 10: ABC
- Kelas 11: ABCG

4.3.1 Model Dasar

Model dasar yang dievaluasi pada percobaan awal menggunakan arsitektur *Multi-Layer Perceptron (MLP)* yang terdiri dari:

- Sebuah input layer dengan 7 neuron, sesuai dengan jumlah fitur energi.
- Hidden layer pertama dengan 20 neuron.
- Hidden layer kedua dengan 10 neuron.
- Sebuah output layer dengan 11 neuron, sesuai dengan jumlah kelas gangguan yang akan diklasifikasi.

Model ini dilatih dan diuji menggunakan dataset yang terdiri dari 198 sampel data. Dataset ini merupakan kombinasi dari 11 jenis gangguan yang disimulasikan pada 2 nilai resistansi, dan 9 lokasi berbeda.

Sesuai dengan metodologi yang dijelaskan pada Bab 3 bagian 3.3.5.3 tentang Proses Pelatihan Model, setiap kali sesi latihan dijalankan menggunakan perintah *train*, maka set data latih termasuk data uji akan diacak menggunakan fungsi *dividerand*. Untuk mengevaluasi stabilitas dan konsistensi kinerja model, dilakukan serangkaian pengujian berulang sebanyak dua puluh kali *Run* sesi latihan. Hasil akurasi dari setiap pengujian disajikan sebagai berikut:

Tabel 4.1 Hasil akurasi dua puluh kali run model dasar

Run ke-	Akurasi (%)
1	73.33
2	73.33
3	76.67
4	76.67
5	70
6	73.33
7	76.67
8	86.67
9	80
10	76.67
11	66.67
12	76.67
13	70
14	76.67
15	73.33
16	63.33
17	76.67
18	70
19	83.33
20	76 .67

Dengan persamaan (2.4) dan (2.5) didapatkan,

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{20}$$

$$\bar{x} = 74,83$$

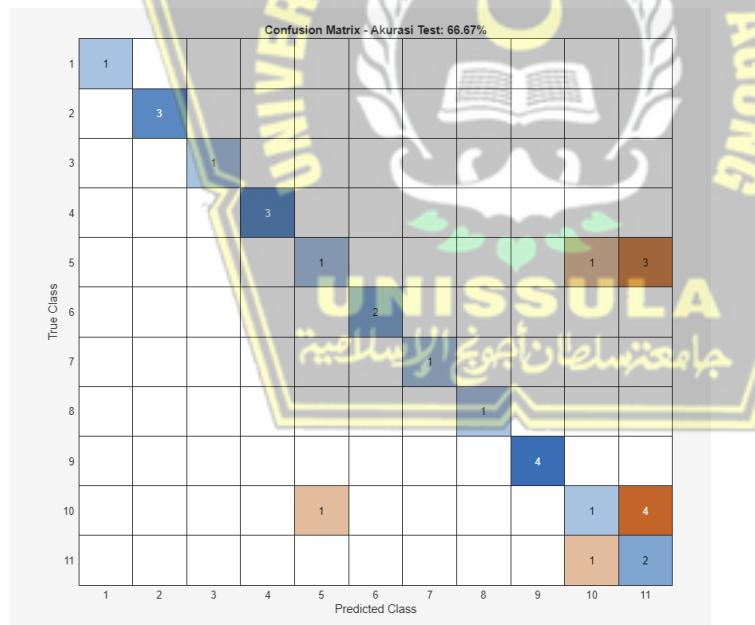
dan

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$$

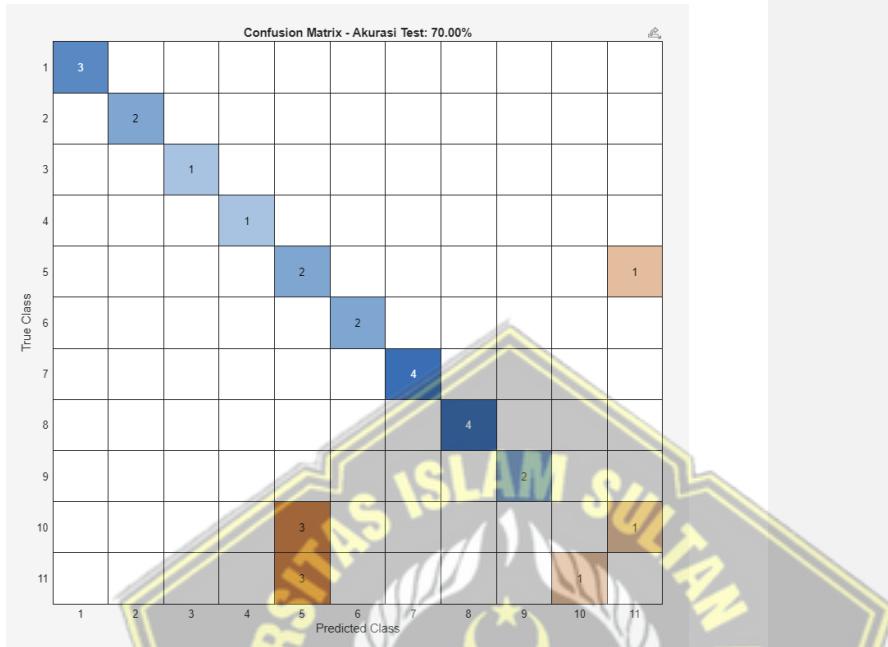
$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - 74,83)^2}{20 - 1}}$$

$$s = 5,35$$

Akurasi pada Tabel 4.1 menghasilkan nilai rata-rata sebesar 74,83 dengan standar deviasi $\pm 5,35\%$, menunjukkan bahwa sebaran data cukup stabil dan konsisten.



Gambar 4.19 Sampel Confusion Matrix untuk akurasi 66.67%



Gambar 4.20 Sampel Confusion Matrix untuk akurasi 70%

Serangkaian hasil pengujian yang dilakukan juga menunjukkan informasi secara umum mengenai kinerja model dalam melakukan klasifikasi. Gambar 4.19 dan Gambar 4.20 merupakan sampel *confusion matrix* ketika model mencapai akurasi 66.67% dan 70%. Analisis menunjukkan pola kesalahan yang terkonsentrasi pada tiga kelas. Kelemahan utama terletak pada Kelas 5, yang seringkali salah diklasifikasikan sebagai Kelas 10 atau 11. Kesalahan ini bersifat dua arah, di mana Kelas 10 dan 11 juga saling tertukar satu sama lain atau terkadang diprediksi sebagai Kelas 5. Kelemahan fundamental dalam membedakan beberapa kelas inilah yang menjadi target utama perbaikan pada tahap optimasi selanjutnya.

4.3.2 Optimasi Data Set

Berdasarkan analisis pada model dasar, hipotesis yang diajukan adalah bahwa ketidakstabilan dan kesalahan klasifikasi terjadi disebabkan oleh jumlah data latih yang tidak memadai. Untuk itu dilakukan penambahan dataset yang lebih besar

sementara arsitektur model 7-20-10-11 tetap dipertahankan konstan. Langkah ini bertujuan untuk mengisolasi dan mengukur secara langsung dampak dari peningkatan volume data terhadap kemampuan model dalam mempelajari pola jenis gangguan dan meningkatkan akurasi.

Dataset yang digunakan terdiri dari 1188 sampel data. Penambahan sampel ini merupakan kombinasi dari 11 jenis gangguan yang disimulasikan pada 2 nilai resistansi, 9 lokasi, 2 kondisi pembebanan dan 3 sudut fasa yang berbeda.

Seperti proses sebelumnya, dilakukan dua puluh kali *Run* sesi latihan untuk mendapatkan stabilitas dan konsistensi dari kinerja model. Hasil akurasi dari setiap pengujian disajikan sebagai berikut:

Tabel 4.2 Hasil akurasi dua puluh kali *run* model setelah optimasi data

Run ke-	Akurasi (%)
1	91.57
2	88.2
3	91.01
4	88.2
5	88.76
6	90.45
7	90.45
8	89.89
9	90.45
10	88.76
11	87.64
12	88.2
13	89.33
14	89.89
15	88.76
16	89.89
17	86.52
18	88.2

19	88.2
20	89.89

Dengan persamaan (2.4) dan (2.5) didapatkan,

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{20}$$

$$\bar{x} = 89,21$$

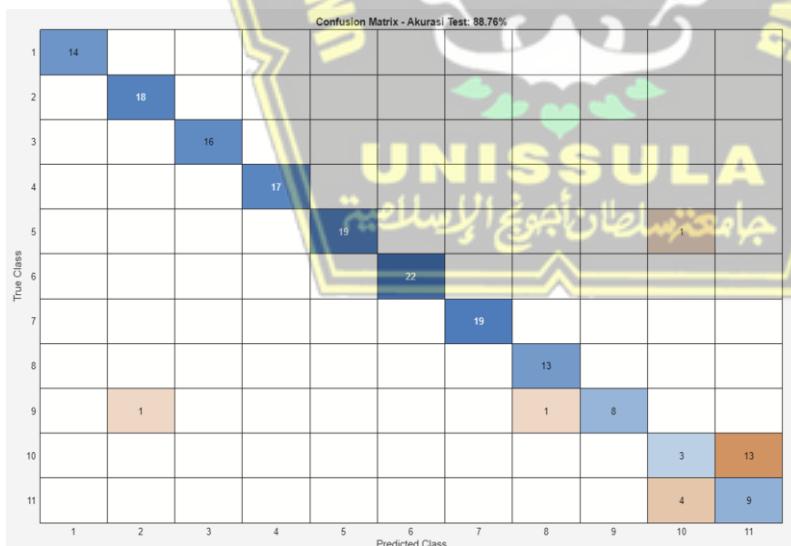
dan

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$$

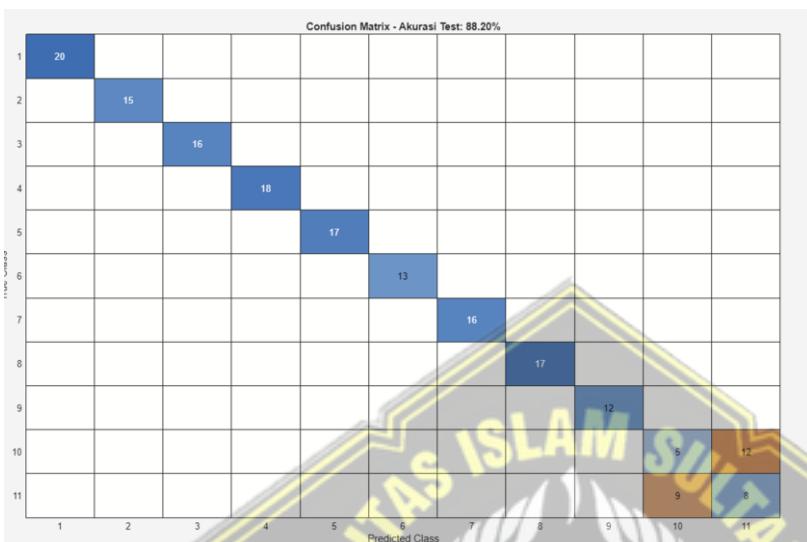
$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - 89,21)^2}{20 - 1}}$$

$$s = 1,27$$

Akurasi pada Tabel 4.2 menghasilkan nilai rata-rata sebesar 89,21 dengan standar deviasi $\pm 1,27\%$, menunjukkan bahwa sebaran data cukup stabil dan konsisten.



Gambar 4.21 Sampel Confusion Matrix untuk akurasi 88.76 %



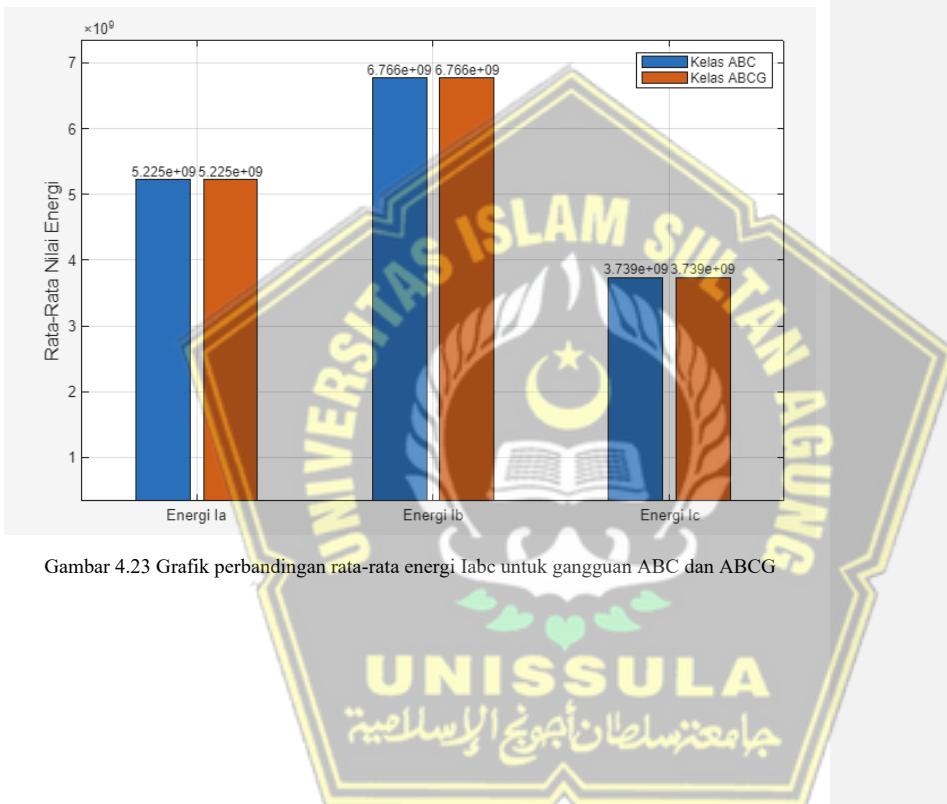
Gambar 4.22 Sampel Confusion Matrix untuk akurasi 88.2 %

Penambahan dataset menjadi 1188 sampel memberikan peningkatan kinerja yang signifikan, dengan akurasi model melonjak ke rentang 88-91%. Sampel *confusion matrix* representatif pada Gambar 4.21 dan Gambar 4.22 menunjukkan bahwa performa klasifikasi untuk Kelas 1 hingga 9 sudah sangat solid dengan tingkat kesalahan yang minimal. Kemajuan signifikan terlihat pada kemampuan klasifikasi untuk gangguan Kelas 5, di mana model tidak lagi menunjukkan kebingungan dengan Kelas 10 dan 11. Namun, sumber kesalahan utama kini terkonsentrasi dan terisolasi pada kebingungan dua arah antara Kelas 10 dan Kelas 11.

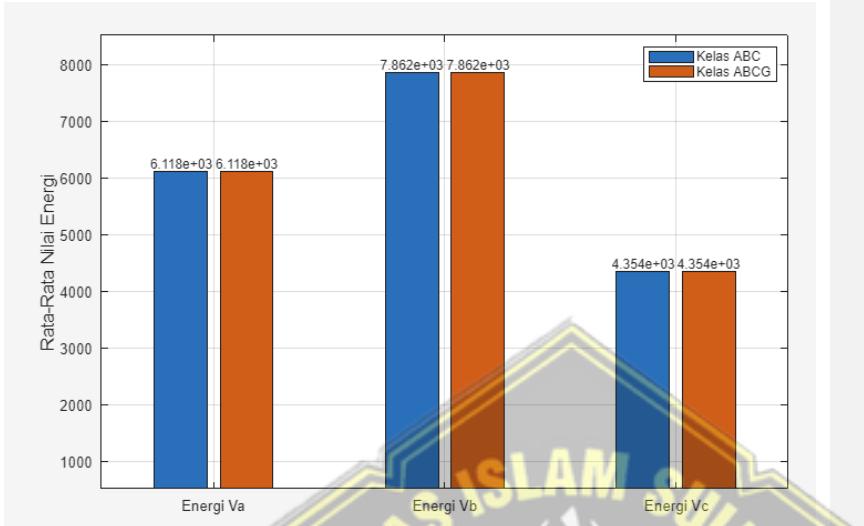
Peningkatan performa ini membuktikan bahwa penambahan data berhasil menyederhanakan masalah dari sebelumnya kebingungan antara tiga kelas (5, 10, 11) menjadi masalah yang lebih spesifik. Meskipun demikian, kelemahan model untuk membedakan antara Kelas 10 dan 11 belum terselesaikan dan akan menjadi target utama untuk optimasi selanjutnya.

4.3.3 Analisis Kebingungan pada Gangguan Simetris (Kelas 10 dan 11)

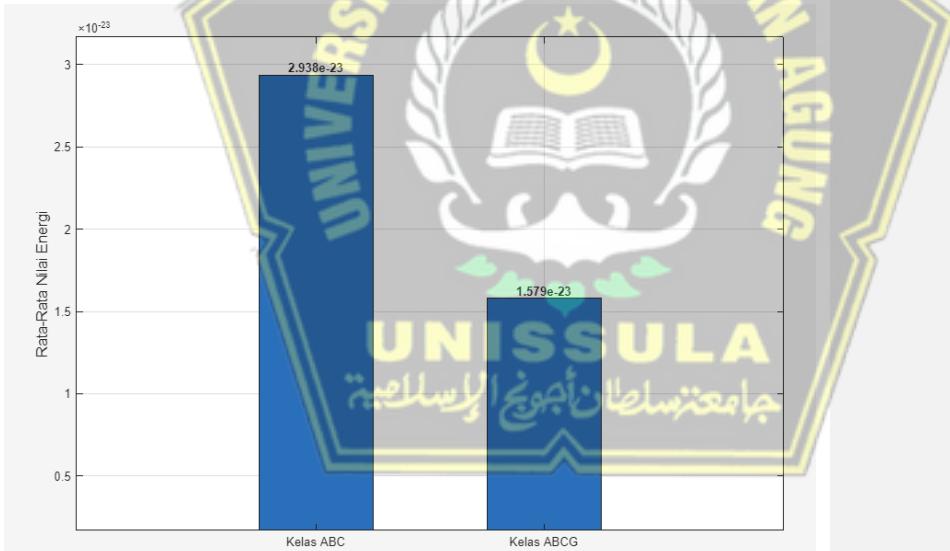
Untuk memahami akar penyebab kebingungan dua arah antara Kelas 10 (gangguan tiga fasa, ABC) dan Kelas 11 (gangguan tiga fasa ke tanah, ABCG), dilakukan analisis mendalam terhadap sinyal V_{abc} , I_{abc} , dan I_0 untuk kedua tipe gangguan tersebut. Dilakukan perhitungan nilai rata-rata fitur dari matriks *dataset_fitur* untuk seluruh sampel gangguan ABC dan ABCG.



Gambar 4.23 Grafik perbandingan rata-rata energi I_{abc} untuk gangguan ABC dan ABCG



Gambar 4.24 Grafik perbandingan rata-rata energi Vabc untuk gangguan ABC dan ABCG



Gambar 4.25 Grafik perbandingan rata-rata energi I0 untuk gangguan ABC dan ABCG

Gambar 4.24 dan 4.25 menunjukkan bahwa karakteristik sinyal dari gangguan ABC dan ABCG memiliki kemiripan yang sangat tinggi, baik dalam pembacaan arus maupun tegangan. Satu-satunya pembeda yang teramati pada Gambar 4.25

adalah munculnya arus residual I_0 pada gangguan ABCG meskipun besarnya tidak signifikan.

Fenomena ini sejalan dengan teori analisis gangguan sistem tenaga, di mana kedua jenis gangguan ini tergolong sebagai gangguan simetris. Kondisi simulasi yang ideal menyebabkan nilai fitur yang dihasilkan untuk kedua gangguan menjadi sangat mirip. Akibatnya, model mengalami kesulitan untuk menemukan pola pembeda yang jelas hanya dari fitur energi yang ada.

Berdasarkan temuan tersebut, diajukan sebuah hipotesis penambahan fitur baru untuk mengamplifikasi perbedaan minor pada sinyal I_0 . Fitur baru ini dihitung menggunakan formula $\text{sum}(\text{abs}(I_0)) * 100$. Perhitungan ini dilakukan pada jendela waktu spesifik di mana sinyal I_0 terdeteksi muncul berdasarkan observasi langsung ketika gangguan ABCG terjadi. Tujuan dari fitur ini adalah untuk mengubah perbedaan magnitudo energi I_0 antara Kelas 10 dan Kelas 11 yang sebelumnya tidak signifikan secara numerik menjadi sebuah fitur yang lebih diskriminatif bagi model.

Total 1188 set data yang digunakan tetap dipertahankan. Kemudian *hidden layer* pertama sedikit diperlebar menjadi 22 neuron untuk mengakomodir penambahan fitur di *layer input*. Pemilihan arsitektur tersebut didasarkan serangkaian proses trial and error berikut:

Tabel 4.3 Percobaan arsitektur hidden layer

No	Arsitektur Hidden Layer	Total Neuron	Akurasi Rata-Rata
1	[20 10]	30	94,8%
2	[22 10]	32	96,18%
3	[22 11]	33	91,99%
4	[30 15]	45	96,07%
5	[31 14]	45	95,63%
6	[40 20]	60	95,67%

Arsitektur lengkap model terbaru setelah penambahan fitur ke-delapan adalah sebagai berikut:

- Sebuah input layer dengan 8 neuron, sesuai dengan jumlah fitur energi + satu fitur sum I_0 .

- Hidden layer pertama dengan 22 neuron.
- Hidden layer kedua dengan 10 neuron.
- Sebuah output layer dengan 11 neuron, sesuai dengan jumlah kelas gangguan yang akan diklasifikasi.

Setelah dilakukan dua puluh kali *Run* perulangan sesi latihan, model menunjukkan akurasi seperti pada tabel berikut:

Tabel 4.4 Hasil akurasi dua puluh kali run model setelah optimasi data

Run ke-	Akurasi (%)
1	96,63
2	96,67
3	96,63
4	97,17
5	97,19
6	95,51
7	96,63
8	96,63
9	97,19
10	94,94
11	96,63
12	96,07
13	96,07
14	95,51
15	93,82
16	94,38
17	95,51
18	98,31
19	95,51
20	96,63

Dengan persamaan (2.4) dan (2.5) didapatkan,

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{20}$$

$$\bar{x} = 96,18$$

dan

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$$

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - 72,5)^2}{20 - 1}}$$

$$s = 1,05$$

Setelah dilakukan optimasi fitur, Tabel 4.4 menunjukkan sebaran data konsisten dengan akurasi rata-rata 96,18% dan standar deviasi 1,05%.

4.3.4 Rekapitulasi Optimasi Kinerja Model

Seluruh proses optimasi yang telah diuraikan pada sub-bab sebelumnya dirangkum dalam tabel berikut:

Tabel 4.5 Rekapitulasi Optimasi Kinerja Model

No	Analisis	Optimasi yang Dilakukan	Hasil Kinerja		Keterangan
			Akurasi (%)	Standar Deviasi (%)	
1	-	-	74,83	5,35	Model dasar
2	Indikasi <i>underfitting</i> karena keterbatasan data latih (kekurangan data)	Penambahan data dari 198 sampel menjadi 1188 sampel	89,21	1,27	Peningkatan akurasi ±14,38%
3	Model mengalami peningkatan performa namun lemah dalam membedakan kelas 10 dan kelas 11	Penambahan fitur arus I0 dan memperlebar hidden layer 1 menjadi 22 neuron	96,18	1,05	Peningkatan akurasi ±6,97% dan kebingungan antara kelas 10 dan 11 terselesaikan



BAB V

PENUTUP

5.1 Kesimpulan

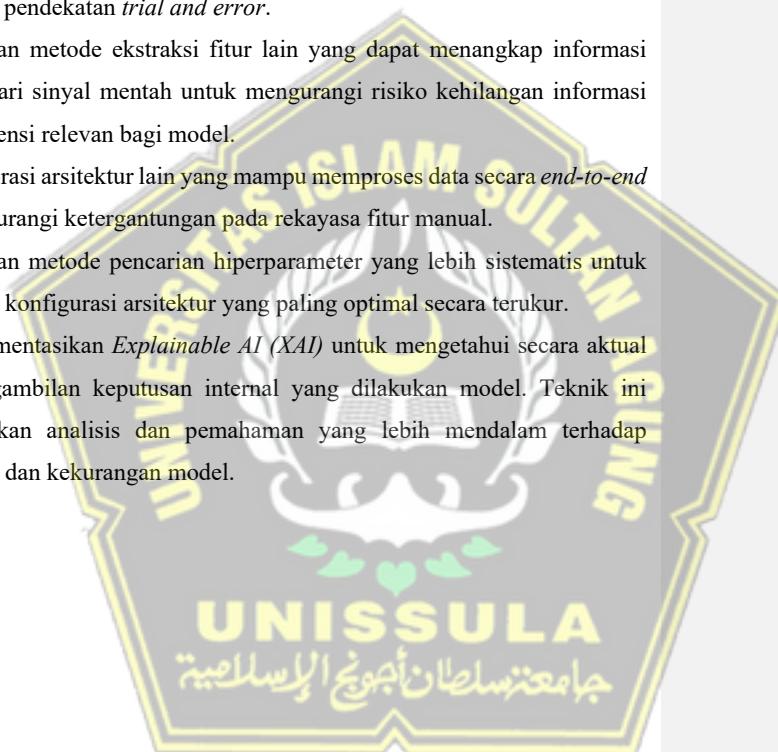
1. Model simulasi saluran transmisi tegangan tinggi 150 kV yang dibangun mampu membangkitkan 11 jenis gangguan dengan karakteristik sinyal arus dan tegangan yang konsisten dengan prinsip-prinsip dasar analisis gangguan sistem tenaga.
2. Delapan fitur esensial, yang terdiri dari tujuh fitur energi *DWT* dan satu fitur magnitudo *I0*, telah berhasil diekstraksi dari sinyal mentah untuk membentuk 1188 sampel data yang merepresentasikan karakteristik gangguan secara numerik.
3. Arsitektur *Multi-Layer Perceptron* final yang digunakan dalam penelitian ini memiliki konfigurasi: 8 neuron input, 22 neuron pada hidden layer pertama, 10 pada hidden layer kedua, dan 11 neuron output. Konfigurasi ini didapatkan dari hasil serangkaian proses optimasi dan perbaikan.
4. Model *Multi-Layer Perceptron* akhir menunjukkan kemampuan klasifikasi dengan akurasi rata-rata sebesar 96,18% dan simpangan baku yang rendah (1,05%). Model final ini mampu mengklasifikasikan seluruh 11 jenis gangguan tanpa menunjukkan adanya kelemahan sistematis yang berulang.

Commented [ik6]: Menyederhanakan kesimpulan

5.2 Saran

1. Penelitian selanjutnya dapat mengembangkan model simulasi untuk mencakup dinamika sistem yang lebih luas, seperti efek kapasitansi atau anomali jaringan lainnya sehingga tidak terbatas pada satu ruas transmisi spesifik,
2. Memperkenalkan model pada data yang mengandung sedikit noise untuk merepresentasikan anomali sistem. Tujuannya agar model yang dihasilkan tidak bergantung pada data simulasi yang terlalu ideal dan bersih.
3. Melakukan validasi model menggunakan data rekaman gangguan riil untuk menguji kinerjanya dalam kondisi operasional yang sebenarnya.

4. Melakukan pengujian dengan dataset dari sistem transmisi yang benar-benar independen dan terpisah untuk mengukur konsistensi dan generalisasi model secara lebih objektif.
5. Memperluas skenario simulasi untuk mencakup variasi gangguan yang lebih kompleks dan lebih menekan batas kemampuan model.
6. Penelitian lanjutan dapat melakukan studi dalam menentukan menentukan jumlah data minimal atau optimal secara lebih sistematis, tidak hanya berdasarkan pendekatan *trial and error*.
7. Menggunakan metode ekstraksi fitur lain yang dapat menangkap informasi lebih luas dari sinyal mentah untuk mengurangi risiko kehilangan informasi yang berpotensi relevan bagi model.
8. Mengeksplorasi arsitektur lain yang mampu memproses data secara *end-to-end* untuk mengurangi ketergantungan pada rekayasa fitur manual.
9. Menggunakan metode pencarian hiperparameter yang lebih sistematis untuk menemukan konfigurasi arsitektur yang paling optimal secara terukur.
10. Mengimplementasikan *Explainable AI (XAI)* untuk mengetahui secara aktual proses pengambilan keputusan internal yang dilakukan model. Teknik ini memungkinkan analisis dan pemahaman yang lebih mendalam terhadap kemampuan dan kekurangan model.



DAFTAR PUSTAKA

- [1] H. Saadat, *Power System Analysis*, 3rd ed. California: PSA Publishing, 2011.
- [2] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurements, Simulation, and Modeling*. New York: John Wiley & Sons, 1991.
- [3] M. C. Ali, "Faults Detection and Classification of Transmission Lines in Power System Using Wavelet Transform," Thesis, International University of Business Agriculture and Technology, 2023.
- [4] Y. S. Rao, G. R. Kumar, and G. K. Rao, "A New Approach for Classification of Fault in Transmission Line With Combination of Wavelet Multi Resolution Analysis and Neural Networks," *Int. J. Power Electron. Drive Syst.*, vol. 8, no. 1, pp. 505–512, 2017, doi: 10.11591/ijpeds.v8i1.pp505-512.
- [5] T. Guo, G. Tian, Z. Ao, X. Fang, L. Wei, and F. Li, "Classification of Transmission Line Ground Short Circuit Fault Based on Convolutional Neural Network," *Energy Eng.*, vol. 119, no. 4, pp. 1715–1728, 2022, doi: 10.32604/ee.2022.018185.
- [6] N. A. Tunio, A. A. Hashmani, S. Khokhar, M. A. Tunio, and M. Faheem, "Fault Detection and Classification in Overhead Transmission Lines Through Comprehensive Feature Extraction Using Temporal Convolution Neural Network," *Eng. Reports*, vol. 6, no. 12, p. e12950, 2024, doi: 10.1002/eng2.12950.
- [7] L. R. Almobasher and I. O. Habiballah, "Review of Power System Faults," *Int. J. Eng. Res. Technol.*, vol. 9, no. 11, pp. 586–590, 2020, doi: 10.17577/IJERTV9IS110036.
- [8] B. De Metz-Noblat, F. Dumas, and C. Poulain, "Calculation of Short-Circuit Currents (Cahier Technique No. 158)," Rueil-Malmaison, France: Schneider Electric, 2005.
- [9] J. D. Glover, M. S. Sarma, and T. J. Overbye, *Power System Analysis and Design: Chapter 7 "Symmetrical Faults,"* 5th ed. Stamford: Cengage Learning, 2012.
- [10] F. B. Costa and B. A. Souza, "Fault-Induced Transient Analysis for Real-Time Fault Detection and Location in Transmission Lines," in *Proceedings of the International Conference on Power Systems Transients*, Delft, Netherlands, 2011. [Online]. Available: https://www.ipstconf.org/papers/Proc_IPST2011/11IPST069.pdf

- [11] N. Bayati and M. Savaghebi, "A Transient Component-Based Technique for Fault Detection in Distributed Generation Systems," in *IEEE 13th International Symposium on Power Electronics for Distributed Generation Systems*, 2022, pp. 590–595. doi: 10.1109/PEDG54999.2022.9923155.
- [12] M. H. J. Bollen and I. Y.-H. Gu, *Signal Processing of Power Quality Disturbances*. Hoboken: John Wiley & Sons, 2006.
- [13] P. S. Addison, "Introduction to Redundancy Rules: The Continuous Wavelet Transform Comes of Age," *Philos. Trans. R. Soc. A*, vol. 376, no. 2116, p. 20170258, 2018, doi: 10.1098/rsta.2017.0258.
- [14] S. O. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Upper Saddle River: Pearson Education, 2009.
- [15] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge: MIT Press, 2017.
- [16] T. Fawcett, "An Introduction to ROC Analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006, doi: 10.1016/j.patrec.2005.10.010.
- [17] D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*, 6th ed. Hoboken: John Wiley & Sons, 2014.
- [18] V. De Andrade and E. Sorrentino, "Typical Expected Values of the Fault Resistance in Power Systems," in *Transmission and Distribution Conference and Exposition: Latin America*, 2010, pp. 1–6. doi: 10.1109/TDC-LA.2010.5762944.