



TUGAS AKHIR - TF 141581

**DIAGNOSIS KONDISI TRANSFORMATOR  
TENAGA MENGGUNAKAN PENDEKATAN  
JARINGAN SYARAF TIRUAN DI PERUSAHAAN  
LISTRIK NEGARA TRANSMISI JAWA BAGIAN  
TIMUR DAN BALI**

GUNTUR PUTRA PAMUNGKAS  
NRP. 02311645000010

Dosen Pembimbing  
Ir. Ya'umar, MT.

DEPARTEMEN TEKNIK FISIKA  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018





**FINAL PROJECT - TF 141581**

**DIAGNOSIS OF POWER TRANSFORMER  
CONDITIONS USING ARTIFICIAL NEURAL  
NETWORK APPROACH IN TRANSMISSION  
STATE ELECTRICAL COMPANY EAST PART  
JAVA AND BALI**

GUNTUR PUTRA PAMUNGKAS  
NRP. 02311645000010

*Supervisor*  
Ir. Ya'umar, MT.

**DEPARTMENT OF ENGINEERING PHYSICS**  
*Faculty of Industrial Technology*  
*Institut Teknologi Sepuluh Nopember*  
*Surabaya 2018*



## **PERNYATAAN BEBAS PLAGIASI**

Saya yang bertanda tangan di bawah ini:

Nama: Guntur Putra Pamungkas

NRP: 02311645000010

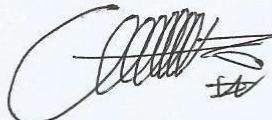
Departemen : Teknik Fisika FTI-ITS

Dengan ini menyatakan bahwa tugas akhir saya berjudul Diagnosis Kondisi Transformator Tenaga Menggunakan Pendekatan Jaringan Syaraf Tiruan di Perusahaan Listrik Negara Transmisi Jawa Bagian Timur dan Bali adalah bebas dari plagiasi. Apabila pernyataan ini terbukti tidak benar, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya.

Surabaya, Juli 2018

Yang membuat pernyataan,



Guntur Putra Pamungkas

*“Halaman ini sengaja dikosongkan”*

## LEMBAR PENGESAHAN

### DIAGNOSIS KONDISI TRANSFORMATOR TENAGA MENGGUNAKAN PENDEKATAN JARINGAN SYARAF TIRUAN DI PERUSAHAAN LISTRIK NEGARA TRANSMISI JAWA BAGIAN TIMUR DAN BALI

#### TUGAS AKHIR

Oleh :



Guntur Putra Pamungkas  
NRP : 02311645000010

Surabaya, 18 Juli 2018  
Mengetahui

Dosen Pembimbing



Ir. Yajumar, M.T.

NIPN. 19540406 198103 1 003



*“Halaman ini sengaja dikosongkan”*

## LEMBAR PENGESAHAN

### DIAGNOSIS KONDISI TRANSFORMATOR TENAGA MENGGUNAKAN PENDEKATAN JARINGAN SYARAF TIRUAN DI PERUSAHAAN LISTRIK NEGARA TRANSMISI JAWA BAGIAN TIMUR DAN BALI

#### TUGAS AKHIR

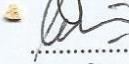
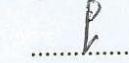
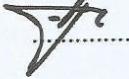
Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Teknik  
pada

Bidang Studi Rekayasa Instrumenasi dan Kontrol  
Program Studi S-1 Departemen Teknik Fisika  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember

Oleh :

Guntur Putra Pamungkas  
NRP. 02311645000010

Disetujui oleh Tim Penguji Tugas Akhir :

1. Ir. Ya'umar, M.T.  ..... (Pembimbing)
2. Dr. Katherin Indriawati, S.T., M.T.  ..... (Ketua Penguji)
3. Hendra Cordova, S.T., M.T.  ..... (Penguji)
4. Ir. Roekmono, M.T.  ..... (Penguji)

SURABAYA, Juli 2018

*“Halaman ini sengaja dikosongkan”*

**DIAGNOSIS KONDISI TRANSFORMATOR TENAGA  
MENGGUNAKAN PENDEKATAN JARINGAN SYARAF  
TIRUAN DI PERUSAHAAN LISTRIK NEGARA  
TRANSMISI JAWA BAGIAN TIMUR DAN BALI**

**Nama : Guntur Putra Pamungkas**  
**NRP : 02311645000010**  
**Departemen : Teknik Fisika FTI ITS**  
**Dosen Pembimbing : Ir. Ya'umar, MT**

**Abstrak**

Terdapat beberapa pengujian untuk menentukan kondisi transformator tenaga dengan memberikan klasifikasi kondisi transformator tenaga yang berbeda-beda dan membutuhkan proses perhitungan matematika. Penggabungan seluruh parameter dan klasifikasi dari berbagai metode tersebut diperlukan agar memudahkan dalam melakukan analisis kondisi transformator tenaga dan membuat keputusan tindak lanjut yang menyeluruh terhadap kondisi transformator tenaga. Maka dari digunakanlah Jaringan Syaraf Tiruan (JST) untuk membuat klasifikasi keputusan yang melibatkan seluruh parameter kondisi transformator tenaga. Oleh karena itu dilakukan penelitian mengenai jaringan syaraf tiruan yang menjadi sebuah perangkat analisis dengan memanfaatkan data analisis yang sudah ada untuk membuat keputusan dalam menentukan kondisi transformator tenaga. Telah diperoleh rancangan arsitektur jaringan syaraf tiruan yang menghasilkan kesalahan minimum (RMSE terkecil dari variasi yang dilakukan) dalam mendiagnosis kondisi transformator tenaga adalah rancangan arsitektur 1 *hidden layer* dengan fungsi aktivasi *logsig (binary sigmoid)* di *hidden layer*, fungsi aktivasi *purelin* (identitas) di *outputlayer*, *input* berjumlah 5 buah. *Hidden node* di *hidden layer* berjumlah 8 buah dan 1 buah *hidden node* target (5-8-1) serta menggunakan 640 data *input-target* menghasilkan 1.5063 untuk nilai RMSE *training*, 1.425 untuk nilai RMSE *testing* dengan secara otomatis di dalam *software*, 1.501 untuk nilai RMSE

*testing* dengan secara manual menggunakan GUI dan 1.4635 untuk nilai RMSE *testing* dengan 25% data.

Kata kunci: Jaringan Syaraf Tiruan, RMSE, Diagnosis, Kondisi Transformator tenaga.

***DIAGNOSIS OF POWER TRANSFORMER  
CONDITIONS USING ARTIFICIAL NEURAL  
NETWORK APPROACH IN TRANSMISSION STATE  
ELECTRICAL COMPANY EAST PART JAVA AND  
BALI***

<b>Name</b>	<b>: Guntur Putra Pamungkas</b>
<b>NRP</b>	<b>: 02311645000010</b>
<b>Department</b>	<b>: Engineering Physics FTI ITS</b>
<b>Supervisor</b>	<b>: Ir. Ya'umar, MT</b>

***Abstract***

*There are several tests to determine the condition of the power transformer by providing a classification of vary power transformer conditions and requires the process of mathematical calculations. The incorporation of all parameters and classifications of the various methods is necessary to facilitate the analysis of power transformer conditions and make a thorough follow-up decision on power transformer conditions. So from the use of Artificial Neural Network (ANN) to make a decision classification involving all parameters of power transformer conditions. Therefore, research on artificial neural networks that become an analytical tool by utilizing existing data analysis to make decisions in determining the condition of power transformers. Artificial architectural design of artificial neural network that has produced minimum error (smallest RMSE of variation done) in diagnosing power transformer condition is architectural design 1 hidden layer with logsig (binary sigmoid) activation function in hidden layer, purelin activation function in outputlayer, input amounted to 5 pieces. Hidden node in the hidden layer of 8 pieces and 1 target hidden node (5-8-1) and using 640 input-target data resulted in 1.5063 for RMSE training, 1,425 for RMSE testing automatically in software, 1.501 for RMSE testing with manually using GUI and 1.4635 for RMSE testing with 25% data.*

*Key words: Artificial Neural Network, RMSE, Diagnosis, Power Transformator Condition.*

## **KATA PENGANTAR**

Syukur Alhamdulillah kami panjatkan kehadirat Allah SWT, karena rahmat dan hidayah-Nya penulis diberikan kesehatan, kemudahan, dan kelancaran dalam menyusun laporan tugas akhir ini. Tidak lupa juga penulis menyampaikan ucapan terimakasih kepada keluarga dan para sahabat yang telah memberi dukungan sehingga penulis mampu menyusun laporan tugas akhir yang berjudul:

### **“DIAGNOSIS KONDISI TRANSFORMATOR TENAGA MENGGUNAKAN PENDEKATAN JARINGAN SYARAF TIRUAN DI PERUSAHAAN LISTRIK NEGARA (PLN) TRANSMISI JAWA BAGIAN TIMUR DAN BALI (JBTB)**

Tugas akhir ini merupakan salah satu persyaratan akademik yang harus dipenuhi dalam Program Studi S-1 Teknik Fisika FTI-ITS. Penulis menyampaikan terimakasih kepada:

1. Agus Muhamad Hatta, ST, MSi, Ph.D. selaku ketua departemen Teknik Fisika ITS.
2. Ir. Ya'umar, MT selaku dosen pembimbing tugas akhir ini, yang selalu memberikan semangat dan ide-ide baru.
3. Hendra Cordova, S.T, M.T selaku dosen wali penulis yang telah memberikan bantuan, arahan, dan nasehat selama penulis menjadi mahasiswa di Departemen Teknik Fisika FTI-ITS.
4. Segenap Bapak/Ibu dosen pengajar dan karyawan di departemen Teknik Fisika ITS.
5. Rekan-rekan warga Teknik Fisika ITS yang telah memotivasi dan memberikan bantuan dalam penyelesaian laporan tugas akhir ini.

Penulis menyadari bahwa masih ada kekurangan dalam laporan ini, untuk itu kritik dan saran penulis terima. Semoga laporan ini dapat berguna dan bermanfaat bagi penulis dan pembaca.

Surabaya, Juli 2018

Penulis

*“Halaman ini sengaja dikosongkan”*

## DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN JUDUL.....	iii
PERNYATAAN BEBAS PLAGIASI .....	v
LEMBAR PENGESAHAN.....	vii
ABSTRAK.....	xi
ABSTRACT.....	xiii
KATA PENGANTAR.....	xv
DAFTAR ISI.....	xvii
DAFTAR GAMBAR .....	xix
DAFTAR TABEL.....	xxi
BAB I PENDAHULUAN .....	1
1.1    Latar Belakang .....	1
1.2    Rumusan Masalah .....	2
1.3    Tujuan .....	2
1.4    Batasan Masalah.....	2
1.5    Sistematika Laporan .....	3
BAB II TEORI PENUNJANG.....	5
2.1    Transformator Tenaga .....	5
2.2    Diagnosis Transformator .....	7
2.2.1    Pengujian <i>Capacitance</i> .....	9
2.2.2    Pengujian Furan.....	10
2.2.3    Pengujian <i>Insulation Resistance</i> .....	11
2.2.4    Pengujian RDC (Tahanan DC) .....	12
2.2.5    Pengujian Ratio .....	13
2.2.6    Pengujian SFRA .....	13
2.2.7    Pengujian Tan Delta Belitan .....	16
2.2.8    Pengujian <i>Dynamic Resistance</i> .....	17

2.2.9	Pengujian Trafo Life Assessment .....	17
2.3	Korelasi <i>Product Moment Pearson</i> .....	18
2.4	Jaringan Syaraf Tiruan.....	18
2.5	Metode Pelatihan <i>Levenberg-Marquardt</i> .....	21
2.6	<i>Root Mean Square Error</i> (RMSE).....	24
2.7	Aplikasi Jaringan Syaraf Tiruan pada Matlab.....	24
BAB III METODE PENELITIAN .....		27
3.1	Studi Literatur.....	28
3.2	Pengumpulan Data <i>Input</i> dan Data Target Di PLN Transmisi JBTB .....	28
3.3	Pengujian Data .....	28
3.4	Perancangan Arsitektur Jaringan Syaraf Tiruan.....	30
3.5	<i>Training</i> Jaringan Syaraf Tiruan .....	32
3.6	<i>Testing</i> Jaringan Syaraf Tiruan .....	32
3.7	Analisa Hasil dan Pelaporan .....	33
BAB IV HASIL DAN PEMBAHASAN .....		35
4.1	Analisa Hasil Pengujian Korelasi Data .....	35
4.2	Analisa Hasil <i>Training</i> Arsitektur Jaringan Syaraf Tiruan .....	37
4.3	Analisa Hasil <i>Testing</i> Jaringan Syaraf Tiruan Dengan Secara Otomatis Di Dalam <i>Software</i> .....	39
4.4	Analisa Hasil <i>Testing</i> Jaringan Syaraf Tiruan Dengan Secara Manual Menggunakan GUI.....	40
BAB V PENUTUP.....		45
5.1	Kesimpulan.....	45
5.2	Saran.....	45
DAFTAR PUSTAKA.....		47
LAMPIRAN .....		49

## DAFTAR GAMBAR

<b>Gambar 2.1.</b> Transformator Siemens 40 MVA [20].....	5
<b>Gambar 2.2.</b> Prinsip Kerja Transformator [11]. .....	6
<b>Gambar 2.3.</b> <i>Bushing</i> [6].....	10
<b>Gambar 2.4.</b> <i>Wiring pengujian SFRA</i> [6]. .....	14
<b>Gambar 2.5.</b> OLTC pada Transformator [6]. .....	17
<b>Gambar 2.6.</b> Neuron Y.....	19
<b>Gambar 2.7.</b> Fungsi aktivasi <i>bipolar sigmoid</i> . .....	20
<b>Gambar 2.8.</b> Fungsi aktivasi <i>binary sigmoid</i> .....	20
<b>Gambar 2.9.</b> Fungsi aktivasi <i>Purelin</i> .....	21
<b>Gambar 2.10.</b> <i>Flowchart</i> Algoritma <i>Levenberg-Marquardt</i> [7].	23
<b>Gambar 2.11.</b> GUI <i>Neural Network Start</i> (nnstart) [9].....	25
<b>Gambar 2.12.</b> Contoh pada Matlab baris perintah program Jaringan Syaraf Tiruan [9]. .....	26
<b>Gambar 2.13.</b> Arsitektur JST [9].....	26
<b>Gambar 3.1.</b> Diagram alir penelitian.....	27
<b>Gambar 3.2.</b> Arsitektur Jaringan Saraf Tiruan .....	31
<b>Gambar 4.1.</b> Hasil <i>training</i> arsitektur JST (5-8-1) .....	38
<b>Gambar 4.2.</b> Hasil <i>testing</i> arsitektur JST (5-8-1) dengan secara otomatis di dalam <i>software</i> .....	40
<b>Gambar 4.3.</b> Tampilan <i>user interface</i> . .....	41
<b>Gambar 4.4.</b> Hasil <i>testing</i> arsitektur JST (5-8-1) dengan secara manual menggunakan GUI .....	42
<b>Gambar 4.5.</b> Hasil <i>testing</i> arsitektur JST (5-8-1) menggunakan 25% dari data total dan dengan 5 <i>Input</i> . .....	43

**(Halaman ini sengaja dikosongkan)**

## **DAFTAR TABEL**

<b>Tabel 3.1.</b> Interpretasi Koefisien Korelasi [13].....	29
<b>Tabel 3.2.</b> Perancangan Arsitektur JST.....	30
<b>Tabel 4.1.</b> Koefisien korelasi (R) antara Xn (n= 1 s/d 9) dengan T .....	36

**(Halaman ini sengaja dikosongkan)**

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Jaringan listrik memiliki salah satu komponen yang mahal yaitu transformator tenaga yang berfungsi mengubah tingkat tegangan dan tingkat arus dari sumber daya listrik. 60% dari total investasi di gardu tegangan tinggi adalah Transformator tenaga [1]. Supaya tidak terjadi kegagalan transformator tenaga, perlu dilakukan pemantauan dan pemeliharaan transformator tenaga. Kondisi transformator tenaga merupakan salah satu aspek penting dalam pemantauan dan pemeliharaan [2].

Dalam pengujian kondisi transformator tenaga, terdapat banyak metode pengujian yang digunakan beserta parameter dan klasifikasinya masing-masing. Macam-macam pengujiannya antaralain DGA, *Physical Inspection, Ultrasonic and Sonic Contact and Noncontact Fault Detection, IR Analysis, Vibration Analysis, Corona Measurement* [4] dan ditambah lagi dengan pengujian *Insulation Power Factor, O & M, Age, Turn Ratio Test, Short Circuit Impedance, DC Winding Resistance, SFRA, Degree of Polymerization, FDS/PDC, Partial Discharge, Bushing test* [2][5]. Setiap pengujian memberikan klasifikasi kondisi transformator tenaga yang berbeda-beda [2] dan membutuhkan proses perhitungan matematika. Penggabungan seluruh parameter dan klasifikasi dari berbagai metode tersebut diperlukan agar memudahkan dalam melakukan analisis kondisi transformator tenaga dan membuat keputusan tindak lanjut yang menyeluruh terhadap kondisi transformator tenaga.

Dikarenakan banyaknya parameter dengan klasifikasi keputusannya masing-masing dan untuk mempersingkat proses penentuan kondisi transformator tenaga maka digunakanlah Jaringan Syaraf Tiruan (JST) untuk membuat klasifikasi keputusan yang melibatkan seluruh parameter kondisi transformator tenaga. JST merupakan alat pemodelan data statistik non-linear yang dapat digunakan untuk memodelkan (memetakan) hubungan kompleks antara *input* dan *output* untuk menemukan pola-pola data yang

diikuti dengan proses *training* dan *testing* [10]. Oleh karena itu dalam tugas akhir ini dilakukan penelitian mengenai jaringan syaraf tiruan yang menjadi sebuah perangkat lunak (*software*) identifikasi kondisi transformator tenaga dengan memanfaatkan data analisis yang sudah ada untuk membuat keputusan dalam menentukan kondisi transformator tenaga.

## 1.2 Rumusan Masalah

Berdasarkan pada latar belakang tersebut, maka permasalahan yang diangkat dalam tugas akhir ini sebagai berikut:

1. Rancangan arsitektur jaringan syaraf tiruan seperti apakah yang menghasilkan kesalahan minimum (RMSE terkecil dari variasi yang dilakukan) dalam mendiagnosis kondisi transformator tenaga.

## 1.3 Tujuan

Berdasarkan permasalahan yang ada, tujuan dilakukan tugas akhir ini adalah sebagai berikut:

- a. Merancang arsitektur jaringan syaraf tiruan dengan kesalahan minimum (RMSE terkecil dari variasi yang dilakukan) dalam mendiagnosis kondisi transformator tenaga.

## 1.4 Batasan Masalah

Adapun lingkup kajian yang digunakan pada tugas akhir ini adalah:

1. Pengambilan data pengujian kondisi transformator tenaga 150/20 kV pada PLN Transmisi Jawa Timur dan Bali.
2. Hasil kondisi tiap pengujian transformator tenaga telah ditentukan oleh PLN Transmisi Jawa Timur dan Bali.
3. Pemrosesan data indikator kondisi transformator tenaga menggunakan Jaringan Syaraf Tiruan (JST).
4. Arsitektur Jaringan Syaraf Tiruan menggunakan *multilayer* dengan algoritma *training Levenberg-Marquadt*.
5. Analisa hasil menggunakan *software* dan Laporan.

## 1.5 Sistematika Laporan

Sistematika laporan dalam penyusunan laporan tugas akhir ini adalah sebagai berikut :

1. Bab I Pendahuluan

Berisi tentang latar belakang, permasalahan, tujuan, batasan masalah, dan sistematika laporan

2. Bab II Teori Penunjang

Berisi tentang teori yang berhubungan dengan transformator tenaga, diagnosis kondisi transformator tenaga, korelasi *product momentum*, sistem jaringan syaraf tiruan, arsitektur jaringan syaraf tiruan, serta metode *training* dari jaringan syaraf tiruan, *root mean squared error*, aplikasi jaringan syaraf tiruan.

3. Bab III Metodologi

Berisi tentang metode yang digunakan dalam penggerjaan tugas akhir dan pengolahan data.

4. Bab IV Analisa Data dan Pembahasan

Berisi tentang evaluasi pengolahan data transformator tenaga, hasil rancangan arsitektur sistem jaringan syaraf tiruan, proses *training* dan *testing* dari sistem jaringan syaraf tiruan yang telah dibuat.

5. Bab V Penutup

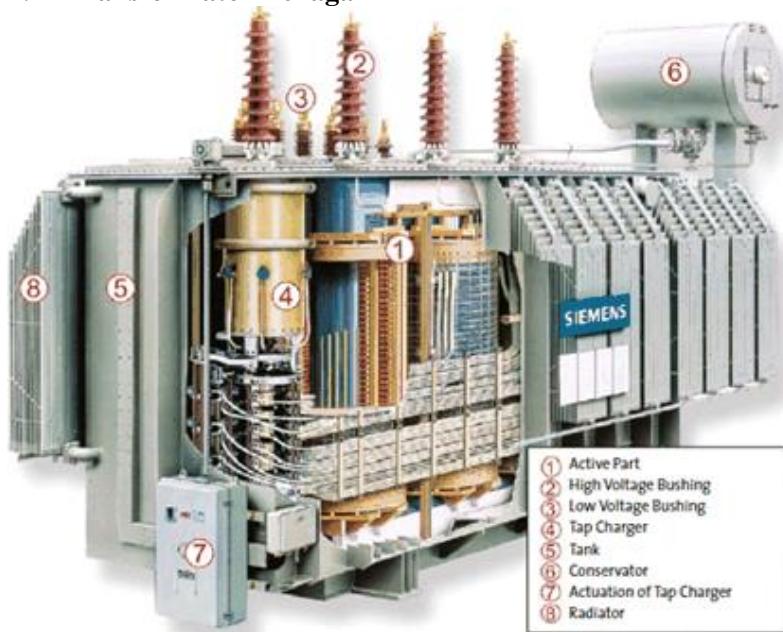
Berisi tentang hasil penelitian, kesimpulan dari penelitian yang telah dilakukan, serta saran sebagai bentuk keberlanjutan dan pengembangan penelitian selanjutnya.

*“Halaman ini sengaja dikosongkan”*

## BAB II

### TEORI PENUNJANG

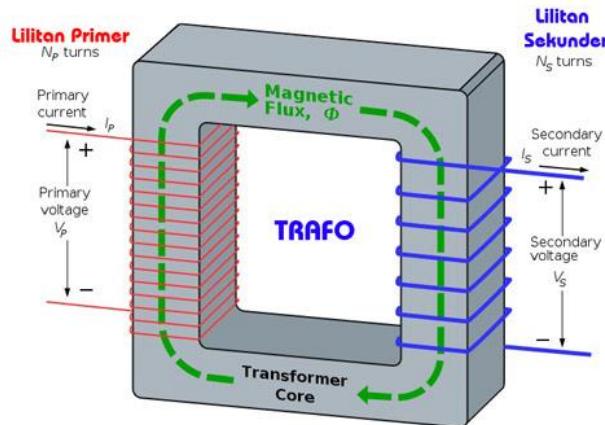
#### 2.1 Transformator Tenaga



Gambar 2.1. Transformator Siemens 40 MVA [20].

**Gambar 2.1.** menampilkan peralatan listrik yang memiliki rangkaian magnetik dan belitan yang terdiri dari dua atau lebih belitan yang berfungsi menransformasikan daya (arus dan tegangan) secara induksi elektromagnetik pada sistem AC ke sistem arus dan tegangan lain pada frekuensi yang sama disebut dengan transformator (trafo) (IEC 60076 -1 tahun 2011). Prinsip elektromagnetik yaitu hukum arus dan induksi faraday digunakan pada trafo dimana medan magnet dapat dibangkitkan oleh perubahan arus atau medan listrik dan tegangan induksi dapat dibangkitkan oleh perubahan medan magnet/fluk medan magnet [6]. Pada belitan primer yang dialiri arus AC, fluk magnet yang mengalir melalui inti besi yang terdapat diantara dua belitan

dibangkitkan. Belitan sekunder diinduksi oleh fluk magnet tersebut sehingga terdapat beda potensial / tegangan induksi pada ujung belitan sekunder [6]. **Gambar 2.2** menunjukkan prinsip kerja transformator.



**Gambar 2.2.** Prinsip Kerja Transformator [11].

Perubahan arus terhadap perubahan waktu yang melewati lilitan primer maupun lilitan sekunder pada **Gambar 2.2** menimbulkan terjadinya Gaya Gerak Listrik (GGL) Induksi diri dan Joseph Henry merumuskan persamaan GGL Induksi diri ditunjukkan pada persamaan (2-1). Dimana L adalah induktansi diri,  $\frac{di}{dt}$  adalah besarnya perubahan arus tiap satuan waktu (A/s) dan  $\mathcal{E}$  adalah GGL induksi diri.

Atau ketika terjadi perubahan arus pada lilitan maka terjadi perubahan fluk magnetik yang menyebabkan terjadinya perubahan tegangan induksi. Dimana  $N$  adalah jumlah lilitan kumparan dan  $\phi$  adalah fluk magnet.

Tegangan induksi total pada masing-masing lilitan sebanding dengan jumlah lilitan yang berkelok-kelok. Jika  $E_1$  adalah tegangan utama dan arus utama  $I_1$ , tegangan sekunder  $E_2$  dan  $I_2$  arus sekunder,  $N_1$  lilitan primer dan  $N_2$  lilitan sekunder, maka:

$$\frac{E_1}{E_2} = \frac{N_1}{N_2} = \frac{I_2}{I_1} \dots \dots \dots \quad (2-3)$$

## 2.2 Diagnosis Transformator

Peralatan dan pelatihan khusus diperlukan untuk mendiagnosa kondisi transformator tenaga [4]. Kondisi transformator pada PLN Transmisi JBTB diklasifikasikan ke dalam 3 kategori kondisi, antara lain: kondisi Buruk (bernilai 1), kondisi Normal (bernilai 6), atau kondisi Baik (bernilai 9). Penentuan nilai kategori kondisi transformator tenaga tersebut berasal dari hasil pengujian *Capacitance*, *Furan*, *Insulation resistance*, *RDC*, *Ratio*, *SFRA*, *Tan delta* belitan, *Dynamic resistance*, dan *Trafo life assessment*. Seluruh pengujian ini dilakukan oleh perusahaan berdasarkan Buku Pedoman Pemeliharaan Transformator Tenaga.

Langkah-langkah penentuan kondisi Buruk (bernilai 1), Normal (bernilai 6), atau Baik (bernilai 9) transformator tenaga adalah sebagai berikut pertama dilakukan pengukuran pada 7 subsistem transformator tenaga, yaitu *Bushing*, Dielektrik, *Electro Magnetic Circuit (EMC) & Current Carrying Unit (CCU)*, *OLTC*, Pendingin, Pernafasan, Struktur Mekanik transformator tenaga. Sarana penghubung antara belitan dengan jaringan luar pada trafo disebut *Bushing*. Daya hantar arus listrik yang sangat kecil atau bahkan tidak ada yang dimiliki suatu bahan disebut Dielektrik [18]. Bahan dielektrik merupakan isolator karena memiliki sifat diseluruh bahan oleh pengaruh medan listrik tidak terdapat elektron-elektron konduksi yang bebas bergerak. Induksi arus bolak-balik pada kumparan yang mengelilingi inti besi mengakibatkan timbulnya flux yang berjalan melalui media *Electro Magnetic Circuit* (Inti besi) yang terbentuk dari lempengan-lempengan besi tipis berisolasi sehingga kumparan yang lain dapat diinduksi kembali. Inti besi dikelilingi batang tembaga berisolasi membentuk belitan atau yang disebut *Current*

*Carrying Unit (Winding)*. Dimana saat belitan tembaga tersebut dialiri arus bolak balik, menginduksi inti besi dan flux magnetik ditimbulkan. Untuk mengubah sambungan penyadapan dari belitan, baik untuk trafo yang sedang *energize* atau berbeban digunakan perangkat yang disebut *On Load Tap Changer* (OLTC). Pendingin yang berupa minyak isolasi merupakan subsistem transformator tenaga yang berfungsi untuk mendinginkan suhu transformator tenaga yang sedang beroperasi agar tidak merusak isolasi kertas pada transformator tenaga. Peristiwa pernafasan pada transformator tenaga adalah ketika suhu operasi pada transformator mengalami kenaikan, volume minyak isolasi bertambah akibat dari memuainya minyak isolasi. Begitu juga ketika suhu operasi mengalami penurunan, maka terjadi penyusutan minyak isolasi dan penurunan volume minyak isolasi. Pada saat transformator mengalami kenaikan suhu, minyak isolasi ditampung menggunakan konservator. Bagian aktif transformator (inti dan belitan) disangga oleh struktur mekanik transformator agar tetap pada posisinya dan ketika terjadi arus hubung singkat, stres mekanik ditahan oleh struktur mekanik transformator [19].

Jika pada tiap subsistem tersebut tidak diisi (tidak diukur) maka bernilai 9. Jika pada tiap subsistem tersebut diisi (diukur) maka bernilai sesuai dengan hasil pengukuran pada tiap subsistem tersebut. Selanjutnya, dicari nilai minimal dari pengukuran di 7 subsistem tersebut. Setelah didapatkan nilai minimal tersebut, apabila nilai minimal itu bernilai 1 maka indeks kondisi transformator tenaga bernilai sementara 1 (Buruk). Apabila nilai minimal itu bernilai lebih dari 1 dan kurang dari 7 maka indeks kondisi transformator tenaga bernilai sementara 6 (Normal). Apabila nilai minimal itu bernilai lebih dari samadengan 7 dan kurang dari samadengan 9 maka indeks kondisi transformator tenaga bernilai sementara 9 (Baik). Nilai sementara kondisi transformator tenaga tersebut digunakan untuk menentukan kondisi akhir transformator tenaga yang disebut *Overall Condition* (sebagai target yang digunakan pada JST). *Overall Condition* ditentukan dari 0.5 kali nilai *trafo life assessment* ditambah dengan 0.5 kali nilai sementara kondisi transformator tenaga maka

didapatkan nilai *Overall Condition* transformator tenaga. Penjelasan mengenai tiap pengujian dijelaskan sebagai berikut.

Indeks kondisi transformator Baik (9) menandakan bahwa semua parameter diagnosis trafo menunjukkan kualitas baik tapi adakalanya beberapa parameter menunjukkan normal atau buruk, secara umum bisa dikategorikan baik. Indeks kondisi transformator Normal (6) menandakan bahwa semua parameter diagnosis trafo menunjukkan kualitas normal tapi adakalanya beberapa parameter menunjukkan baik atau buruk, secara umum bisa dikategorikan normal. Indeks kondisi transformator Buruk (1) menandakan bahwa semua parameter diagnosis trafo menunjukkan kualitas buruk tapi adakalanya beberapa parameter menunjukkan normal atau baik, secara umum bisa dikategorikan buruk.

### 2.2.1 Pengujian *Capacitance*

Pengujian nilai *capacitance* (kapasitansi) dilakukan pada *bushing*. Sarana penghubung antara belitan dengan jaringan luar pada trafo disebut *bushing* (ditampilkan pada **Gambar 2.3**) [6]. Sebuah konduktor yang diselubungi oleh isolator membentuk *bushing*. Penyekat antara *body main tank* transformator tenaga dengan konduktor *bushing* merupakan fungsi dari isolator. Diantara *ground flange* (bagian yang tidak bertegangan) dan konduktor (bagian yang bertegangan) terdapat nilai kapasitansi yang dimiliki *bushing* dengan inti konduktor yang dibelit oleh media isolasi kertas.

Pengujian kapasitansi (C) setiap trafo dimulai dari mengukur nilai C (uji) untuk fasa R, fasa S, fasa T, dan fasa N pada *Bushing Primer*, *Bushing Sekunder*, dan *Bushing Tersier*. Didapatkan nilai C (uji) untuk fasa R, fasa S, fasa T, dan fasa N pada *Bushing Primer*, *Bushing Sekunder*, dan *Bushing Tersier* kemudian dihitung persen selisih antara nilai C (uji) dengan nilai C (inisial) di setiap *Bushing* untuk fasa R, fasa S, fasa T, dan fasa N. Apabila nilai persen selisih C kurang dari 5% atau tidak diukur maka memiliki *score* 9. Apabila nilai persen selisih C lebih dari samadengan 5% dan kurang dari 10% maka memiliki *score* 6. Apabila nilai persen selisih C lebih dari samadengan 10% maka

memiliki *score* 1. Untuk setiap fasa jika *Availability tap test* menunjukkan “tidak” maka memiliki *score* 1 selain itu maka memiliki *score* 9. Nilai pengujian kapasitansi diperoleh dari *score* minimum persen selisih nilai sekaligus dengan *availability tap test*.



**Gambar 2.3. Bushing [6].**

### 2.2.2 Pengujian Furan

Sistem isolasi trafo memiliki bagian yang disebut isolasi kertas. Kekuatan mekanik dan *spacing* disediakan oleh media dielektrik yang merupakan fungsi dari isolasi kertas. Kualitas isolasi kertas dapat menurun akibat dari panas yang berlebih dan *by-product* dari oksidasi minyak. Proses depolimerisasi merupakan proses penurunan kualitas isolasi kertas. Pada proses depolimerisasi, kekuatan *tensile* dari isolasi kertas turun akibat dari rantai hidrokarbon yang panjang terputus / terpotong – potong. Gugus furan terbentuk selalu diiringi oleh proses depolimerisasi. Penurunan tingkat DP (*degree of polymerization*) sebanding dengan nilai furan yang terbentuk [6].

Perkiraan kondisi DP yang dialami isolasi kertas dan perkiraan sisa umur kertas isolasi tersebut (*Estimated percentage of remaining life – %Eprl*) dapat diketahui dari kandungan gugus furan 2Fal (2-Furaldehyde) yang terdeteksi.

Rumus perhitungan estimasi DP:

$$DP = \frac{[Log_{10}(2FaI_{ppb}*0.88)-4,51]}{-0.0035} \dots \dots \dots (2-4)$$

Dimana  $2\text{F}\alpha\text{l}_{\text{ppb}}$  merupakan kandungan gugus furan yang terdeteksi.

Rumus perhitungan estimasi %Eprl:

$$\%Eprl = 100 - \left\{ \frac{\lfloor Log_{10}(DP) - 2.903 \rfloor}{-0.00602} \right\} \dots \dots \dots (2-5)$$

Rata – rata kondisi DP isolasi kertas diindikasikan dari hasil pengujian furan. Pengujian DP secara langsung pada sampel isolasi kertas sebagai verifikasi kondisi isolasi kertas perlu dilakukan ketika hasil uji furan telah mendekati nilai *End of Expected Life* isolasi kertas [6].

Didapatkan nilai konsentrasi 2Fal mg/g/g dari kertas [17]. Apabila nilai konsentrasi 2Fal kurang dari 473 mg/g/g dari kertas, maka nilai pengujian Furan samadengan 9. Apabila nilai konsentrasi 2Fal lebih dari samadengan 473 dan kurang dari 2197 mg/g/g dari kertas, maka nilai pengujian Furan samadengan 6. Apabila nilai konsentrasi 2Fal lebih dari samadengan 2197 mg/g/g dari kertas, maka nilai pengujian Furan samadengan 1.

### 2.2.3 Pengujian *Insulation Resistance*

Tahanan yang digunakan untuk mengisolasi transformator supaya tidak terjadi panas yang dapat merusak transformator itu sendiri disebut dengan *insulation resistance* (tahanan isolasi). Tujuan dari pengujian tahanan isolasi adalah untuk mengukur nilai indeks polarisasi dan untuk mengetahui kondisi isolasi antara belitan dengan *ground* atau antara dua belitan. Nilai tahanan isolasi ditunjukkan oleh nilai indeks polarisasi. Indeks polarisasi adalah perbandingan hasil pengujian tahanan isolasi pada menit ke 10 dengan pada menit ke 1[6].

Pengujian tahanan isolasi ini menguji tahanan isolasi trafo di 6 titik yaitu primer-*ground*, sekunder-*ground*, tersier-*ground*, primer-*sekunder*, primer-*tertier*, sekunder-*tertier*. Di setiap titik diukur nilai tahanan (megaohm) pada menit ke 1 dan menit ke 10 kemudian nilai tahanan pada menit ke 10 dibandingkan dengan nilai tahanan pada menit ke 1 untuk mendapatkan nilai indeks polarisasi (IP). Untuk setiap titik jika indeks polarisasi kurang dari 1 maka memiliki *score* 1. Jika indeks polarisasi lebih dari samadengan 1 dan kurang dari 1.26 maka memiliki *score* 6. Jika indeks polarisasi lebih dari samadengan 1.26 maka memiliki *score* 9. Nilai pengujian tahanan isolasi sendiri merupakan *score* minimum dari 6 titik pengujian tersebut.

#### 2.2.4 Pengujian RDC (Tahanan DC)

Maksud dari pengujian tahanan DC adalah untuk mengukur nilai resistif (R) dari belitan dengan memberikan arus dc (*direct current*) pada belitan. Tujuan dilakukannya pengujian tahanan dc adalah untuk mengetahui kelayakan dari koneksi – koneksi yang ada di belitan dan untuk memperkirakan jika ada kemungkinan hubung singkat atau resistansi yang tinggi pada koneksi di belitan. Proses pengukuran dilakukan pada masing – masing belitan pada titik fasa ke netral pada trafo tiga fasa [6]. Pengukuran tahanan dc menggunakan *micro ohmmeter* atau jembatan *wheatstone*. *Micro ohmmeter* merupakan alat untuk mengukur nilai resistif dari sebuah tahanan dengan orde  $\mu\Omega$  (*micro ohm*) sampai dengan orde  $\Omega$  (*ohm*). Pada umumnya trafo – trafo berdaya rendah menggunakan jembatan *wheatstone* [6].

Pengujian tahanan DC dimulai dari menguji tahanan DC di fasa R, fasa S, dan fasa T. Ada 18 Tap pengujian fasa R, S, dan T, untuk setiap trafo. Disetiap Tap didapatkan tahanan DC di phasa R, S, dan T, lalu dicari berapa persen selisih terhadap fasa lain. Persen selisih fasa R terhadap fasa S, Persen selisih fasa S terhadap fasa T, dan Persen selisih fasa R terhadap fasa T. Apabila persen selisih lebih dari 3% maka memiliki *score* 1. Apabila persen selisih lebih dari 2% dan kurang dari samadengan 3% maka memiliki *score* 6. Apabila persen selisih kurang dari

samadengan 2% maka memiliki *score* 9. Nilai pengujian tahanan DC diperoleh dari *score* minimum persen selisih terhadap fasa lain seluruh Tap (18 Tap).

### 2.2.5 Pengujian Ratio

Pada dasarnya tujuan dari pengujian ratio belitan adalah untuk mengidentifikasi adanya masalah dalam antar belitan dan seksi – seksi sistem isolasi pada trafo. Adanya hubung singkat antar lilitan, putusnya lilitan, maupun ketidaknormalan pada *tap changer* dideteksi oleh pengujian ini. Metoda pengujian ratio belitan adalah sebagai berikut tegangan variabel diberikan pada sisi HV kemudian tegangan yang muncul pada sisi LV dilihat. Ratio perbandingan tegangan pada sisi LV dan sisi HV dapat diketahui dengan membandingkan tegangan sumber dengan tegangan yang muncul. Alat *Transformer Turn Ratio Test* digunakan untuk melakukan pengujian [6].

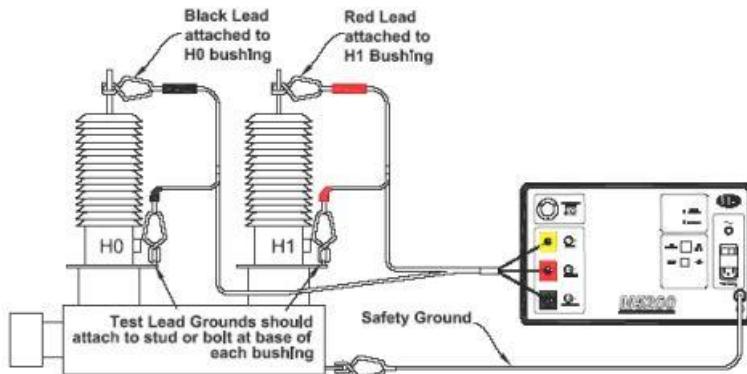
Pengujian ratio dimulai dari menguji ratio fasa R, fasa S, dan fasa T pada masing-masing Tap Atas, Tap Nominal, dan Tap Bawah untuk setiap trafo. Didapatkan nilai ratio fasa R, fasa S, dan fasa T pada masing-masing Tap Atas, Tap Nominal, dan Tap Bawah kemudian dihitung persen selisih antara ratio hasil uji dengan ratio *nameplate*. Apabila persen selisih kurang dari 0.3% maka memiliki *score* 9. Apabila persen selisih lebih dari samadengan 0.3% dan kurang dari 0.5% maka memiliki *score* 6. Apabila persen selisih lebih dari samadengan 0.5% maka memiliki *score* 1. Nilai pengujian ratio diperoleh dari *score* minimum persen selisih antara ratio hasil uji dengan ratio *nameplate*.

### 2.2.6 Pengujian SFRA

Informasi tentang adanya pergeseran pada inti dan belitan transformator dapat diberikan dengan menggunakan peralatan yang disebut SFRA (*Sweep Frequency Response Analyzer*). Bagaimana belitan memberikan sinyal bertegangan rendah dalam berbagai variasi frekuensi dapat diketahui dengan melakukan pengujian ini. Transformator merupakan

rangkaian impedansi dimana unsur – unsur kapasitif dan induktif berhubungan dengan konstruksi fisik transformator. Perubahan – perubahan dalam frekuensi respons terukur dalam teknik SFRA yang mengindikasikan perubahan fisik dalam suatu transformator yang harus didentifikasi dan diinvestigasi.

Kesatuan struktur mekanik dari inti, belitan dan struktur *clamping* pada trafo dievaluasi dengan SFRA yaitu dengan mengukur fungsi transfer elektrik terhadap sinyal bertegangan rendah dalam rentang frekuensi yang lebar. Skema *wiring* pengujian SFRA ditampilkan pada **Gambar 2.4**. Dengan membandingkan hasil pengukuran terbaru terhadap referensi, SFRA melakukan evaluasi kondisi trafo [6].



**Gambar 2.4.** Wiring pengujian SFRA [6].

Deformasi belitan (Axial dan Radial seperti *hoop buckling*, *tilting* dan *spiralling*), pergeseran antar belitan, *partial winding collapse*, lilitan yang terhubung singkat atau putus, kegagalan pentanahan pada inti atau *screen*, pergerakan inti, kerusakan struktur *clamping*, dan permasalahan pada koneksi internal dapat dideteksi oleh SFRA. Apabila terjadi hal-hal sebagai berikut, sebelum dan setelah transportasi, gempa dan gangguan hubung singkat yang besar maka dilakukan pengujian SFRA.

Pengujian SFRA pada setiap trafo dilakukan dengan menggunakan metode CCF (Cross Correlation Factor). Ada 2 CCF yaitu CCF dengan hasil uji *fingerprint*/sebelumnya dan CCF dengan fasa lain (jika tidak memiliki *fingerprint*). Pertama yaitu dengan metode CCF dengan hasil uji *fingerprint*/sebelumnya. CCF dengan hasil uji *fingerprint*/sebelumnya terdiri dari 4 Test yaitu Open Test High Voltage (HV), Open Test Low Voltage (LV), Open Test LV-HV dan Short Test LV. Setiap Test memiliki 3 rentang frekuensi pengujian yaitu Low Frekuensi (LF), Medium Frekuensi (MF), dan High Frekuensi (HF). Di setiap rentang frekuensi diukur fungsi transfer elektrik terhadap sinyal bertegangan rendah pada fasa R-N, fasa S-N, dan fasa T-N.

Pada LF setiap fasa dilakukan pengukuran. Apabila lebih dari samadengan 2 maka memiliki *score* 9. Bila tidak terdapat pengukuran juga memiliki *score* 9. Apabila lebih dari samadengan 1 dan kurang dari 2 maka memiliki *score* 6. Apabila kurang dari 1 maka memiliki *score* 1. Pada MF setiap fasa dilakukan pengukuran. Apabila lebih dari samadengan 1 maka memiliki *score* 9. Bila tidak terdapat pengukuran juga memiliki *score* 9. Apabila lebih dari samadengan 0.6 dan kurang dari 1 maka memiliki *score* 6. Apabila kurang dari 0.6 maka memiliki *score* 1. Pada HF setiap fasa dilakukan pengukuran. Apabila lebih dari samadengan 0.6 maka memiliki *score* 9. Bila tidak terdapat pengukuran juga memiliki *score* 9. Apabila tidak dua kondisi itu maka memiliki *score* 1.

Yang kedua dengan metode CCF dengan fasa lain (jika tidak memiliki *fingerprint*). Penjelasan dan cara pengukuran dengan metode ini sama halnya dengan penjelasan dan cara pengukuran metode CCF dengan hasil uji *fingerprint*/sebelumnya. Nilai pengujian SFRA diperoleh dari *score* minimum diseluruh pengukuran pada fasa R-N, fasa S-N, dan fasa T-N baik metode CCF dengan hasil uji *fingerprint*/sebelumnya maupun metode CCF dengan fasa lain (jika tidak memiliki *fingerprint*).

### 2.2.7 Pengujian Tan Delta Belitan

*Loss Angle* merupakan metoda diagnostik secara elektikal untuk mengetahui kondisi isolasi, sering disebut Tan Delta atau pengujian factor disipasi. Isolasi bersifat kapasitif sempurna seperti halnya sebuah isolator yang berada diantara dua elektroda pada sebuah kapasitor apabila isolasi bebas *defect*. Tegangan dan arus fasa bergeser  $90^\circ$  pada kapasitor sempurna dan isolasi yang dilewati arus merupakan kapasitif. Nilai tahanan dari isolasi berkurang apabila ada kontaminasi pada isolasi yang berdampak isolasi tersebut dilewati tingginya arus resistif dan mengakibatkan tidak lagi menjadi kapasitor sempurna. Tegangan dan arus bergeser kurang dari  $90^\circ$ . Tingkat kontaminasi pada isolasi direpresentasikan oleh besarnya selisih pergeseran dari  $90^\circ$  [6].

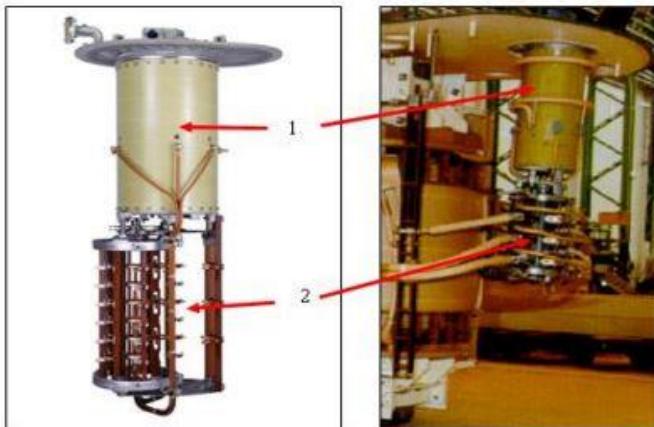
Secara garis besar sistem isolasi trafo terdiri dari isolasi antara belitan dengan *ground* dan isolasi antara dua belitan. Di lingkungan PT PLN terdapat tiga metode pengujian untuk trafo, yaitu metode trafo dua belitan, metode trafo tiga belitan dan metode *autotrafo*. Titik-titik pengujian trafo dua belitan yaitu Primer-*Ground* (CH), Sekunder-*Ground* (CL), dan Primer-Sekunder (CHL). Titik-titik untuk pengujian trafo tiga belitan yaitu, Primer-*Ground*, Sekunder-*Ground*, Tertier-*Ground*, Primer-Sekunder, Sekunder-Tertier, dan Primer-Tertier. Perbedaan dan beberapa pertimbangan dalam metode pengujian untuk autotrafo dilakukan samadengan metode trafo dua belitan antaralain; pada autotrafo sisi HV dan LV dirangkai menjadi satu belitan yang tidak dapat dipisahkan, sehingga *bushing* HV, LV dan Netral menjadi satu titik pengujian (Primer). Sisi Belitan TV menjadi satu titik pengujian (Sekunder) [6].

Pengujian Tan Delta Belitan dilakukan di 6 titik yaitu HL, LT, HT, HG, LG, dan TG. Pada setiap titik yaitu pada titik HL, LT, HT, HG, LG, dan TG, apabila nilai tan delta kurang dari samadengan 0.5 maka memiliki *score* 9. Apabila nilai tan delta lebih dari 0.5 dan kurang dari 1 maka memiliki *score* 6. Apabila nilai tan delta lebih dari samadengan 1 maka memiliki *score* 1.

Nilai pengujian Tan Delta Belitan diperoleh dari *score* minimum seluruh nilai tan delta di 6 titik tersebut.

### 2.2.8 Pengujian Dynamic Resistance

Ketidaknormalan kerja pada OLTC (*On Load Tap Changer*) khususnya yang berkaitan dengan kontak *diverter* maupun selektor *switch* diketahui dengan melakukan pengukuran *dynamic resistance*. Untuk mengubah sambungan penyadapan dari belitan, baik untuk trafo yang sedang *energize* atau berbeban digunakan perangkat yang disebut *On Load Tap Changer* (OLTC) ditampilkan pada **Gambar 2.5** [16].



Keterangan:

1. Kompartemen Diverter Switch

2. Selektor Switch

**Gambar 2.5.** OLTC pada Transformator [6].

### 2.2.9 Pengujian Trafo Life Assessment

Pengujian *trafo life assessment* dilakukan dengan mendapatkan nilai minimum dari hasil pengujian *Capacitance*, pengujian Furan, pengujian *Insulation Resistance*, pengujian

RDC, pengujian Ratio, pengujian SFRA, pengujian Tan Delta Belitan, dan pengujian *Dynamic Resistance*.

### 2.3 Korelasi Product Moment Pearson

Derajat hubungan searah bukan timbal balik antara dua variabel atau lebih dinyatakan dalam istilah statistik yaitu Korelasi [13]. Salah jenis teknik korelasi yaitu *Product Moment Pearson*. *Product Momentum Pearson* digunakan apabila kedua variabelnya berskala interval. Ada atau tidaknya hubungan antara variabel X dengan variabel Y dan besarnya sumbangan variabel satu terhadap yang lainnya yang dinyatakan dalam persen dapat dinyatakan dengan menggunakan korelasi *Product Moment Pearson*. Besarnya korelasi *Product Moment Pearson* ditunjukkan oleh nilai koefisien korelasi parsial ( $r$ ).

$$r_{xy} = \frac{n \sum XY - \sum X \sum Y}{\sqrt{n \sum X^2 - (\sum X)^2} \sqrt{n \sum Y^2 - (\sum Y)^2}}. \quad (2-6)$$

Keterangan:

$r_{xy}$  = nilai  $r$  dari perhitungan.

X = variabel yang dicari korelasinya.

Y = variabel yang dicari korelasinya.

n = jumlah data

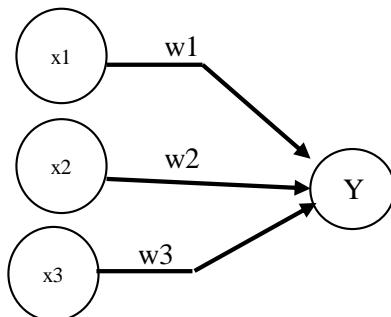
### 2.4 Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (JST) adalah sistem pemroses informasi yang memiliki karakteristik mirip dan dibentuk sebagai generalisasi model matematika jaringan syaraf biologi [3], dengan asumsi bahwa:

- Pada elemen-elemen sederhana (neuron) terjadi pemrosesan informasi.
- Sinyal-sinyal informasi melalui penghubung dikirimkan antar neuron-neuron.
- Bobot yang memperkuat atau memperlemah sinyal dimiliki oleh penghubung antar neuron.

- d. Setiap neuron menggunakan fungsi aktivasi yang dikenakan pada jumlahan *input* yang diterima untuk menentukan *output*. Besarnya *output* dibandingkan dengan batas ambang.

Tiga hal yang menentukan jaringan syaraf tiruan, yaitu: arsitektur jaringan (pola hubungan antar neuron), metode *training/learning/algoritma* untuk menentukan bobot penghubung, dan fungsi aktivasi [3].



**Gambar 2.6.** Neuron Y.

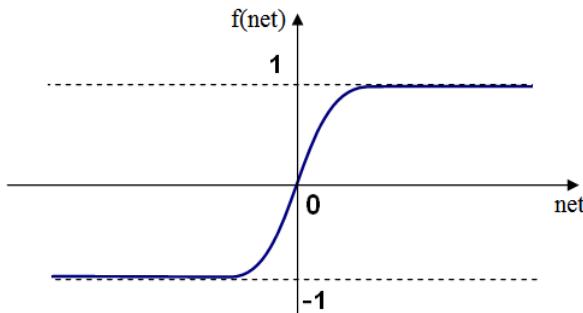
Pada **Gambar 2.6** Y menerima *input* dari neuron x1, x2, x3 dengan bobot hubungan masing-masing adalah w1, w2, dan w3. Tiga impuls neuron yang ada dijumlahkan menjadi persamaan (2-7).

$$\text{net} = x_1w_1 + x_2w_2 + x_3w_3 \dots \dots \dots \quad (2-7)$$

Y menerima besarnya impuls mengikuti fungsi aktivasi  $y = f(\text{net})$ . Sinyal diteruskan jika nilai fungsi aktivasi mencapai batas ambang. Dasar untuk mengubah bobot memakai nilai fungsi aktivasi (keluaran model jaringan) [3].

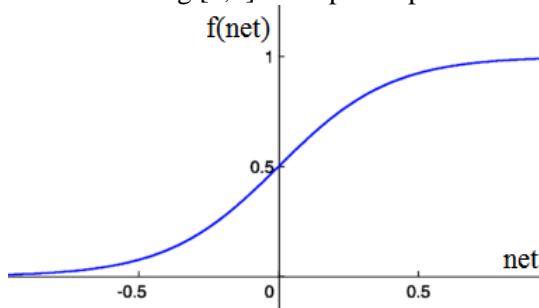
Fungsi yang menentukan apakah sinyal dari *input* neuron diteruskan ke neuron lain atau tidak disebut dengan fungsi aktivasi [14]. Macam-macam fungsi aktivasi adalah sebagai berikut:

- a. Tansig (*bipolar sigmoid*).  $f(\text{net}) = \frac{2}{1+e^{-\text{net}}} - 1$ . Fungsi sigmoid bipolar memiliki rentang [-1,1] ditampilkan pada **Gambar 2.7**.



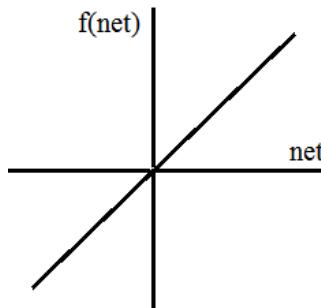
**Gambar 2.7.** Fungsi aktivasi *bipolar sigmoid*.

- b. Logsig (*binary sigmoid*).  $f(\text{net}) = \frac{1}{1+e^{-\text{net}}}$ . Fungsi sigmoid biner memiliki rentang [0,1] ditampilkan pada **Gambar 2.8**.



**Gambar 2.8.** Fungsi aktivasi *binary sigmoid*.

- c. Purelin (fungsi identitas).  $f(\text{net}) = \text{net}$ . Fungsi identitas memiliki nilai *output* yang samadengan nilai *input*-nya. Fungsi identitas ditampilkan pada **Gambar 2.9**.



**Gambar 2.9.** Fungsi aktivasi *Purelin*.

## 2.5 Metode Pelatihan *Levenberg-Marquardt*

Algoritma pelatihan *Levenberg Marquardt* digunakan dalam penelitian ini. Pendekatan matrik Hesian ( $H$ ) yang merupakan pengembangan algoritma *backpropagation* standar digunakan dalam algoritma *Levenberg-marquardt* [12]. Beberapa kekurangan yang dimiliki algoritma *backpropagation* diatasinya oleh algoritma *Levenberg-marquardt* dengan memanfaatkan teknik optimisasi numerik standar yaitu menggunakan pendekatan matrik Jacobian yang berada pada matrik Hesian. Pada dasarnya proses pelatihan ini meminimalkan *error function* dan penurunan algoritma ini dapat dijelaskan sebagai berikut ditampilkan pada **Gambar 2.10** [7].

Penjelasan algoritma pada **Gambar 2.10** dijelaskan sebagai berikut :

1. Inisialisasi bobot dan bias dengan bilangan acak, epoch maksimum, dan minimal *goal* (*performance* yang dihitung dengan MSE).
2. Menentukan parameter yang dibutuhkan, antara lain :
  - a. Parameter *Levenberg Marquardt* yang nilainya harus lebih besar dari nol.
  - b. Parameter faktor Beta ( $\beta$ ) yang digunakan sebagai parameter yang dikalikan atau dibagi dengan parameter *Levenberg Marquardt*. Penjelasan ada pada step berikutnya.

3. Menghitung maju (*feedforward*) pada *hidden* dan *output layer* seperti langkah-langkah pada algoritma error *backpropagation*.
  4. Menghitung nilai MSE
  5. Menghitung error dan total error jaringan
    - a. Rumus untuk error :

r merupakan *input* ke-r

- b. Rumus untuk menghitung total error :

e merupakan vektor kesalahan berukuran Nx1 yang terdiri dari  $e_r$  r = 1,2,3...N

6. Menghitung matriks Jacobian  $J(x)$  x merupakan matriks yang berisi nilai bobot dan bias dari keseluruhan jaringan.

$X = [v_{11}, v_{12}, \dots, v_{ij}; v_{01}, v_{02}, \dots, v_{0j}; w_{11}, w_{12}, \dots, w_{jk}; w_{01}, w_{02}, \dots, w_{0K}]$

Matriks Jacobian berisi turunan pertama *error* jaringan terhadap bobot dan bias jaringan. Rumus untuk mencari Jacobian Matriks antara lain.

$$J = \left[ \frac{\partial er}{\partial w} \right] \dots \quad (2-10)$$

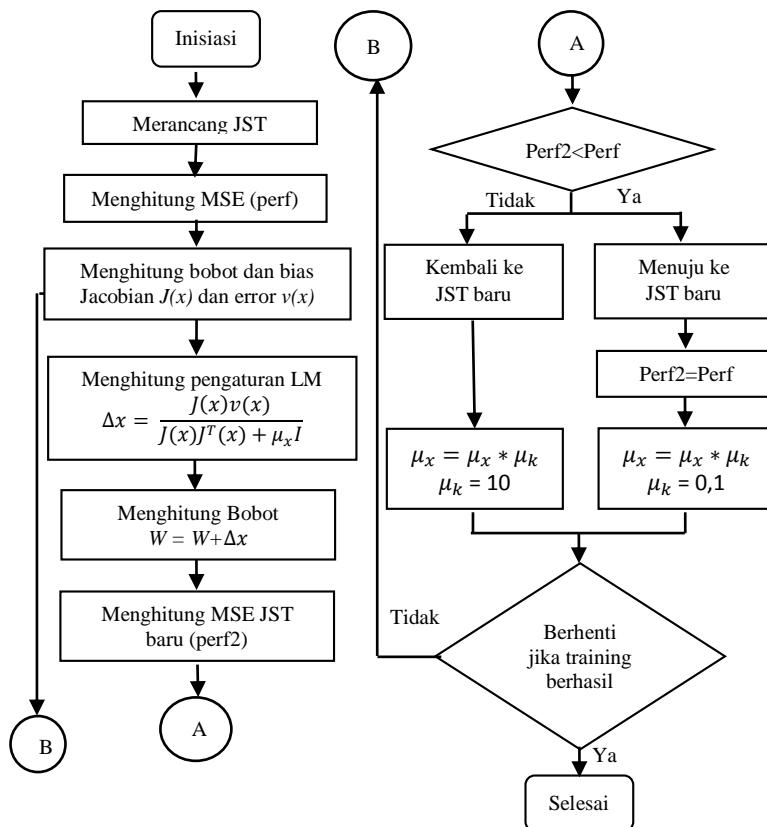
7. Setelah didapatkan nilai  $J(x)$  maka dapat dihitung perubahan koreksi bobot dan biasnya dengan rumus berikut :

8. Setelah didapatkan nilai  $\Delta X$  tahap selanjutnya adalah pengoreksian bobot dengan rumus yang sama seperti pada algoritma *error back propagation*.
  9. Menghitung maju (*feedforward*) dengan bobot dan bias yang baru.
  10. Menghitung MSE jaringan dengan bobot dan bias yang baru. Kemudian tes kondisi berhenti.

11. Jika epoch atau iterasi masih berlanjut maka akan terdapat 2 kemungkinan kondisi berikut :

- Jika MSE naik  $\beta\mu x$
- Jika MSE turun  $\beta\mu$

Kemudian melakukan kembali langkah 5 sampai langkah 8.



**Gambar 2.10.** Flowchart Algoritma Levenberg-Marquardt [7].

## 2.6 Root Mean Square Error (RMSE)

*Root Mean Square Error* (RMSE) merupakan parameter yang digunakan untuk mengevaluasi nilai hasil dari pengamatan / pengukuran terhadap nilai sebenarnya atau nilai yang dianggap benar [8]. RMSE didapatkan dari akar MSE hasil pelatihan JST.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2} = \sqrt{\frac{1}{n} \sum_{i=1}^n E_i^2} \dots \dots \dots \quad (2-13)$$

$$RMSE = \sqrt{MSE} \dots \dots \dots \quad (2-14)$$

Keterangan:

n = jumlah data.

$Y_i$  = output JST tiap data.

$\hat{Y}_i$  = target JST tiap data.

i = 1, 2, 3, ..., n

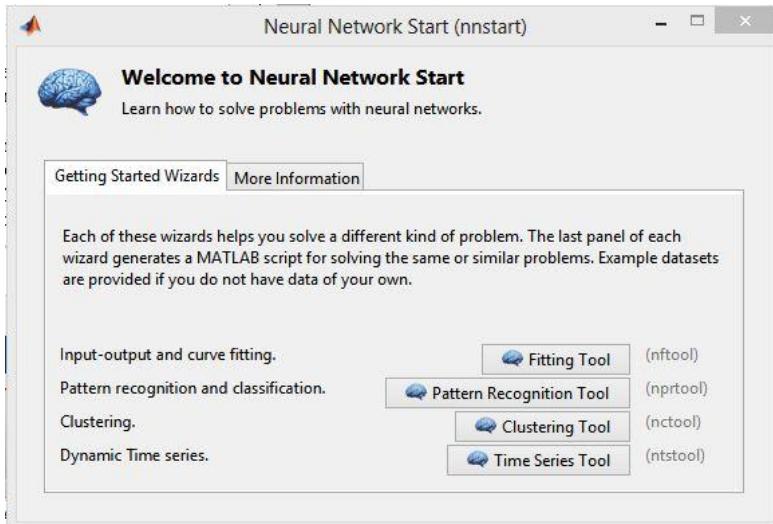
Ei = error tiap data.

MSE = Mean Square Error.

## 2.7 Aplikasi Jaringan Syaraf Tiruan pada Matlab

Salah satu fitur di Matlab adalah aplikasi jaringan syaraf tiruan (JST). *Neural Network Toolbox* (NNT) merupakan sebutan untuk aplikasi JST di Matlab. Tersedia berbagai fungsi dan aplikasi dengan menerapkan prinsip-prinsip JST untuk melakukan pemodelan sistem *nonlinear* yang kompleks di *Toolbox* ini. Pelatihan dengan supervisi dan tanpa supervisi didukung di NNT. Pengguna menggunakan NNT untuk merancang berbagai macam Jaringan Syaraf Tiruan, melatih, dan mensimulasikan JST tersebut. Berbagai aplikasi seperti *fitting* data, pengenalan pola, pengelompokan data, prediksi, dan pemodelan sistem dinamik dapat menggunakan NNT [9].

Pada Matlab, terdapat dua cara untuk merancang JST menggunakan NNT. Pertama yaitu *toolbox* berbasis *Graphical User Interface* (GUI) tampak pada **Gambar 2.10** dapat digunakan. Dengan mengetik perintah “nnstart” pada *command line*, *toolbox* berbasis *Graphical User Interface* (GUI) tampil.



**Gambar 2.11.** GUI *Neural Network Start* (nnstart) [9].

Dengan mengetik perintah “nnstart” pada *command line* Matlab, *toolbox* muncul dengan tampilan ditunjukkan pada **Gambar 2.11**. *Toolbox* itu terdapat dua tab. Tab pertama adalah *Getting Started Wizards* dan tab kedua adalah *More Information*. Tab *Getting Started Wizards* menyediakan 4 macam jenis JST untuk masalah yang berbeda. Yang pertama adalah “nftool” atau *Fitting Tool*. Jenis JST ini berguna untuk masalah pemasangan data. Yang kedua adalah “nprrtool” atau *Pattern Recognition Tool*. Jenis JST ini berguna untuk masalah pengenalan pola. Yang ketiga adalah “nctool” atau *Clustering Tool*. Jenis JST ini berguna untuk masalah pengelompokkan data. Yang keempat adalah “ntstool” atau *Time Series Tool*. Jenis JST ini berguna untuk masalah memprediksi data. Tab kedua adalah *More Information*. Tab ini berisi informasi tambahan dari “nnstart” *Toolbox*.

Cara kedua yaitu dengan *script* yang berisi fungsi-fungsi JST yang tersedia di *Neural Network Toolbox* dituliskan di Matlab yang ditunjukkan pada **Gambar 2.12**. Setiap baris di *script* (**Gambar 2.12**) memiliki fungsinya masing-masing. Baris pertama yaitu load house\_dataset berfungsi untuk memuat dataset kedalam jaringan

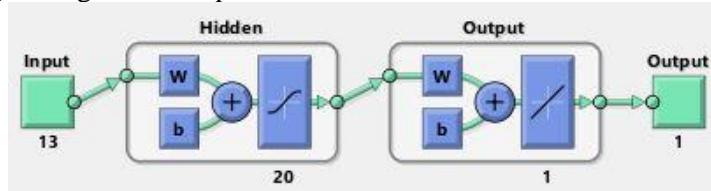
syaraf tiruan yang dibuat. Baris kedua yaitu `net = feedforwardnet(20)` berfungsi untuk membuat jaringan syaraf tiruan feedforward. Angka 20 menunjukkan jumlah node yang digunakan pada *hidden layer*. Baris ketiga yaitu `[net,tr] = train(net,houseInputs,houseTargets)` berfungsi untuk melakukan training pada jaringan dengan memasukkan *input* (`houseInputs`) dan target (`houseTargets`).

```

1 load house_dataset
2 net = feedforwardnet(20);
3 [net,tr] = train(net,houseInputs,houseTargets);
```

**Gambar 2.12.** Contoh pada Matlab baris perintah program Jaringan Syaraf Tiruan [9].

Tiga baris *script* pada **Gambar 2.12** mencakup inisialisasi sampai pelatihan JST. **Gambar 2.13** menampilkan arsitektur JST dengan konfigurasi 13-20-1 yang dibuat berdasarkan perintah pada **Gambar 2.12**. Arsitektur JST pada Matlab yang ditampilkan oleh **Gambar 2.13** memiliki satu *input layer* dengan jumlah input sebanyak 13 variabel, satu *hidden layer* yang memiliki 20 *hidden node* dengan fungsi aktivasi *bipolar sigmoid* dan 1 *output layer* dengan fungsi aktivasi *purelin*.

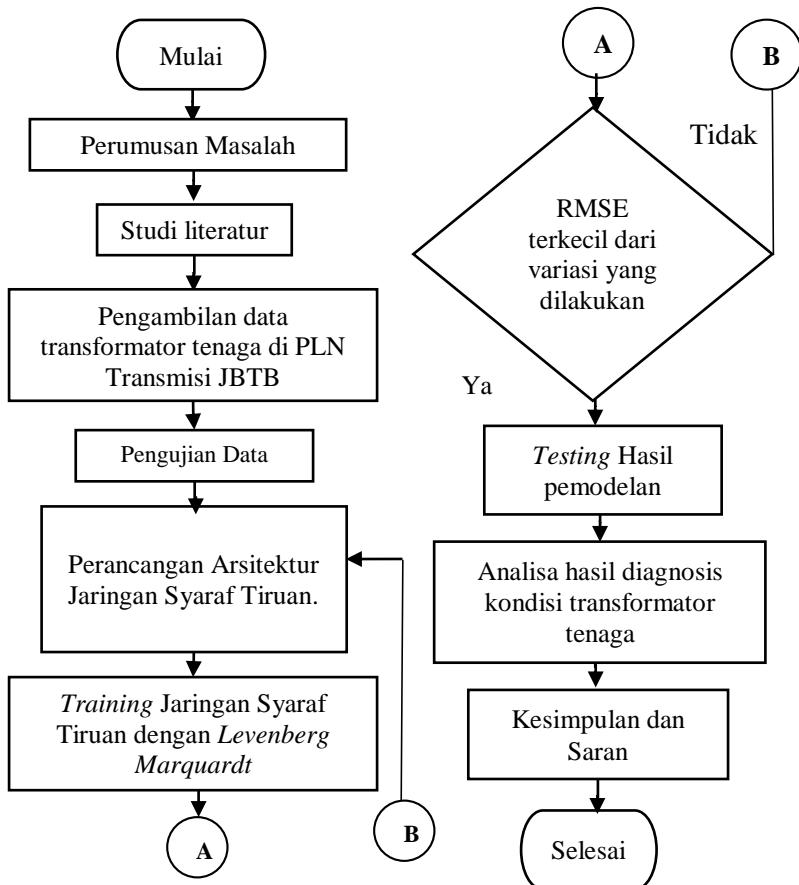


**Gambar 2.13.** Arsitektur JST [9].

### BAB III

## METODE PENELITIAN

Diagram alir dari penelitian ini ditunjukkan pada **Gambar 3.1.**



**Gambar 3.1.** Diagram alir penelitian.

### 3.1 Studi Literatur

Studi Literatur merupakan kegiatan mempelajari kepustakaan yang menunjang dalam pengerjaan penelitian. Kepustakaan yang dipelajari berasal dari buku, jurnal penelitian ilmiah, artikel ilmiah, laporan tugas akhir mengenai transformator tenaga, Jaringan Syaraf Tiruan, RMSE, dan *Levenberg-Marquardt* serta informasi-informasi yang diperoleh dari pihak PLN Transmisi Jawa Bagian Timur dan Bali (JBTB).

### 3.2 Pengumpulan Data *Input* dan Data Target Di PLN Transmisi JBTB

Data yang dibutuhkan untuk pengerjaan tugas akhir ini adalah sembilan hasil pengujian transformator tenaga yaitu pengujian *Capacitance*, Furan, *Insulation resistance*, RDC, Ratio, SFRA, Tan delta belitan, *Dynamic resistance*, dan *Trafo life assessment* yang berperan sebagai *input* jaringan syaraf tiruan untuk mendiagnosis kondisi transformator tenaga serta satu klasifikasi kondisi transformator tenaga (kondisi bernilai 1, kondisi bernilai 6, atau kondisi bernilai 9) yang berperan sebagai target. Data didapat dari PLN Transmisi JBTB.

### 3.3 Pengujian Data

Pengujian data bertujuan untuk memastikan terdapat hubungan antara variabel-variabel *input* dengan variabel-variabel target. *Software* statistika digunakan untuk pengujian data. Pengujian data yang dilakukan adalah Uji Korelasi *Product Moment Pearson*. Uji korelasi *Product Moment Pearson* dilakukan untuk mengetahui nilai koefisien korelasi parsial antara 9 variabel *input* terhadap 1 variabel target. Koefisien korelasi parsial dilambangkan dengan huruf r. Interpretasi nilai r yang digunakan adalah pada **Tabel 3.1**.

**Tabel 3.1.** Interpretasi Koefisien Korelasi [13].

R	Interpretasi
0	Tidak korelasi
0,01-0,20	Sangat rendah
0,21-0,40	Rendah
0,41-0,60	Agak rendah
0,61-0,80	Cukup
0,81-0,99	Tinggi
1	Sangat tinggi

Langkah-langkah menghitung koefisien korelasi parsial ( $r$ ) adalah sebagai berikut [13]:

1. Tulis  $H_0$  dan  $H_a$  dalam bentuk kalimat.  $H_0$  adalah hipotesis tidak terdapat hubungan antara dua variabel yang diuji (misal X dan Y).  $H_a$  adalah hipotesis terdapat hubungan antara dua variabel yang diuji. Ada dua macam  $H_a$ ;  $H_a$  *two-tailed* dan  $H_a$  *one-tailed*.  $H_a$  *two-tailed* memiliki pengertian terdapat hubungan antara dua variabel yang diuji.  $H_a$  *one-tailed* memiliki pengertian semakin besar nilai variabel X maka semakin besar nilai variabel Y atau semakin besar nilai variabel X maka semakin kecil nilai variabel Y.
2. Tulis  $H_0$  dan  $H_a$  dalam bentuk statistik.
3. Hitung  $r$  dengan rumus persamaan (2-4).
4. Tentukan taraf signifikansinya ( $\alpha$ ). Tingkat signifikansi ( $\alpha$ ) menunjukkan probabilitas (peluang) kesalahan yang ditetapkan peneliti dalam mengambil keputusan untuk menolak atau mendukung hipotesis nol, atau dapat diartikan juga sebagai tingkat kesalahan atau tingkat kekeliruan yang ditolerir oleh peneliti, yang diakibatkan oleh kemungkinan adanya kesalahan dalam pengambilan sampel (sampling error) [15].
5. Cari  $r$  tabel dengan derajat kebebasan (dk),  $dk = n-2$ .
6. Tentukan kriteria pengujian. Jika  $-r_{tabel} \leq r_{hitung} \leq +r_{tabel}$ ,  $H_0$  diterima.
7. Bandingkan  $r_{hitung}$  dengan  $r_{tabel}$

8. Buat kesimpulan.

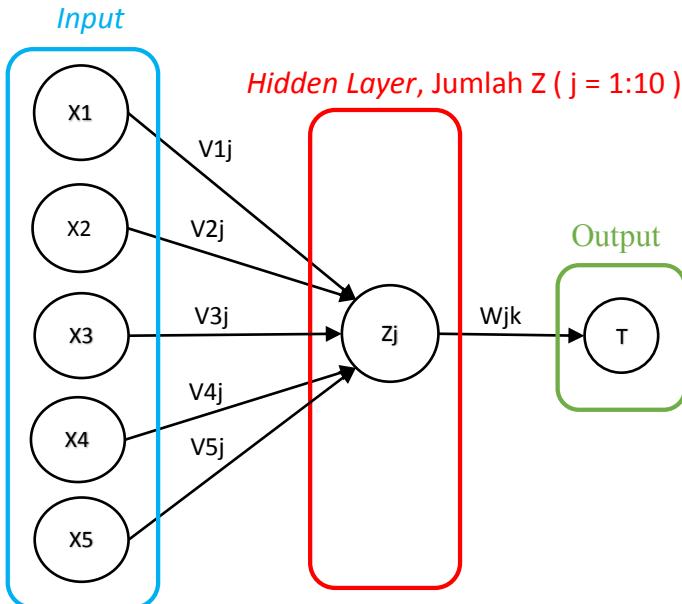
Dari pengujian korelasi *Product Moment Pearson* 9 variabel *input* terhadap 1 variabel target didapatkan 5 variabel *input* yang memiliki interpretasi r mulai dari Rendah hingga Tinggi terhadap 1 variabel target.

### 3.4 Perancangan Arsitektur Jaringan Syaraf Tiruan

Arsitektur JST yang dirancang adalah *Multilayer* dengan algoritma *Levenberg-Marquardt*. **Gambar 3.2** merupakan arsitektur JST untuk variasi *hidden node* mulai dari 1 *hidden node* hingga 10 *hidden node* dengan 5 buah *input* dan 1 buah target. **Tabel 3.2** menampilkan beberapa perancangan arsitektur JST yang dilakukan pada penelitian ini dengan jumlah *input* hasil dari uji korelasi *input-target* dan 1 buah target, yaitu:

**Tabel 3.2.** Perancangan Arsitektur JST

No	Jumlah			Fungsi aktivasi
	<i>Hidden layer</i>	<i>Hidden node</i>	data	
a	1	1 s/d 10	170	Purelin
b		1 s/d 20		
c		1 s/d 10, & 1		
d		1 s/d 10, & 2		
e		1 s/d 10, & 3		
f		1 s/d 10, & 4		
g		1 s/d 10, & 5		
h		1 s/d 10, & 6		
i		1 s/d 10, & 7		
j		1 s/d 10, & 8		
k	2	1 s/d 10, & 9	170	Logsig & Logsig
l		1 s/d 10, & 10		
m		1 s/d 10	320	Logsig
n				
o				
p				
q	1	960	170	Tansig
r		1 s/d 20		Logsig



Keterangan sebagai berikut:

T = Target

X1 = *Capacitance*

X2 = RDC

X3 = SFRA

X4 = Tan Delta Belitan

X5 = *Trafo Life Assesment*.

Z<sub>j</sub> = *hidden node*, jumlah *hidden node* divariasikan mulai dari *hidden node* berjumlah 1 buah, 2 buah, 3 buah, hingga 10 buah. Variasi tersebut dinotasikan dengan notasi perulangan (j = 1:10)

V<sub>1j</sub> s/d V<sub>5j</sub> = bobot *hidden node* (j = 1:10).

W<sub>jk</sub> = bobot keluaran, (j = 1:10) & (k = 1).

**Gambar 3.2.** Arsitektur Jaringan Saraf Tiruan

Dari pengujian korelasi *Product Moment Pearson* 9 variabel *input* terhadap 1 variabel target didapatkan 5 variabel *input* yang memiliki interpretasi r mulai dari Rendah hingga Tinggi terhadap

1 variabel target. Pada **Gambar 3.2** X1 sampai X5 merupakan variabel *input* yang dinormalisasi menjadi 0 hingga 1 pada jaringan syaraf tiruan. V1j s/d V5j dan Wk bertindak sebagai bobot yang bernilai acak yang menghubungkan masing-masing *node*. Jumlah *hidden node* divariasikan mulai 1 sampai 10 buah *node* dengan jumlah *hidden layer* yang tidak divariasikan yaitu 1 buah.

### 3.5 Training Jaringan Syaraf Tiruan

Data yang dimiliki dibagi menjadi 75% untuk *training* dan 25% untuk *testing*. *Training* dilakukan pada tiap variasi arsitektur yang disebutkan pada subbab perancangan arsitektur JST. Algoritma *Levenberg-Marquadt* digunakan untuk *training* JST ini. Parameter JST yang diatur adalah `net.trainParam.epochs`. Parameter `epochs` diatur pada 50 epochs dan ada yang diatur 100 epochs. Parameter `net.trainParam.lr` atau *learning rate* ( $\alpha$ ) diatur pada nilai *default*-nya yaitu sebesar 0,01. Nilai  $\alpha$  semakin besar maka proses pelatihan semakin cepat tetapi jika terlalu besar  $\alpha$ , maka algoritma menjadi tidak stabil dan mencapai titik minimum lokal. Dari tiap *training* tersebut kemudian dipilih satu variasi *hidden node* yang memiliki nilai RMSE terkecil. Nilai RMSE terkecil ini menunjukkan bahwa arsitektur JST dengan variasi *hidden node* tersebut mendapatkan diagnosis dengan menghasilkan nilai RMSE terkecil pada tahap *training* dibandingkan dengan 9 variasi *hidden node* yang lain. Arsitektur JST dengan nilai RMSE terkecil ini kemudian disimpan untuk selanjutnya digunakan pada tahap *testing* dengan data (data uji) yang berbeda dari data yang digunakan pada tahap *training* (data latih).

### 3.6 Testing Jaringan Syaraf Tiruan

Dari 25% data, 12.5% data digunakan untuk *testing* dengan cara yang pertama yaitu memasukkannya kedalam Ms.Excel dan terhubung dengan *Software* sehingga proses *testing* tidak perlu memasukkan *input* secara manual ke JST. 12.5% data yang lain digunakan untuk *testing* dengan cara yang kedua yaitu dengan memasukkan *input* secara manual ke GUI JST. *Testing* dilakukan untuk mendapatkan nilai RMSE yang dihasilkan dari arsitektur

JST yang memiliki nilai RMSE terkecil pada tahap *training*. Nilai RMSE yang dihasilkan dari tahap *testing* dengan menggunakan data uji ini menunjukkan bahwa arsitektur JST yang didapat dari tahap *training* mendiagnosis kondisi transformator tenaga disertai dengan nilai RMSE yang dihasilkan.

### **3.7 Analisa Hasil dan Pelaporan**

Pada tahap ini dilakukan analisis mengenai hasil dari jaringan syaraf tiruan yang telah dirancang pada tahap sebelumnya. Hasil percobaan dan analisis selanjutnya dituliskan dalam bentuk laporan.

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

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

Dalam penelitian ini telah dilakukan eksperimen dari perancangan arsitektur JST **Tabel 3.2**. Eksperimen ini menggunakan perancangan **Tabel 3.2** untuk 5 *input* dan menggunakan perancangan **Tabel 3.2** untuk 2 *input*. 5 *input* diperoleh dari hasil pengujian korelasi data dengan koefisien korelasi (R) yang dipilih untuk digunakan bernilai mulai dari 0.2 hingga 0.6, sedangkan 2 *input* diperoleh dari hasil pengujian korelasi data dengan koefisien korelasi (R) yang dipilih untuk digunakan bernilai mulai dari 0.4 hingga 0.6.

Didapatkan satu rancangan arsitektur yang memiliki RMSE paling kecil diantara beberapa perancangan arsitektur JST yang dilakukan, yaitu rancangan arsitektur (urutan yang ke 0) 1 *hidden layer* dengan fungsi aktivasi *logsig (binary sigmoid)* di *hidden layer*. Fungsi aktivasi *purelin* (identitas) di *output layer*. *Input* berjumlah 5 buah. *Hidden node* di *hidden layer* berjumlah 8 buah dan 1 buah *hidden node* target (5-8-1) serta menggunakan 640 data *input-target*. Hasilnya antara lain: hasil pengujian korelasi data, hasil *training* arsitektur jaringan syaraf tiruan, dan hasil *testing* pemodelan jaringan syaraf dari arsitektur

#### **4.1 Analisa Hasil Pengujian Korelasi Data**

Hasil pengujian korelasi data ditampilkan pada **Tabel 4.1**.

Keterangan:

$X_1$  sampai  $X_9$  merupakan *input* dan T merupakan target. b menunjukkan tidak adanya korelasi (hubungan).

$X_1 = Capacitance$

$X_2 = Furam$

$X_3 = Insulations Resistance$

$X_4 = RDC$

$X_5 = Ratio$

$X_6 = SFRA$

$X_7 = Tan Delta Belitan$

$X_8 = Dynamic Resistance$

$$X_9 = \text{Trafo Life Assesment}$$

**Tabel 4.1.** Koefisien korelasi (R) antara Xn (n= 1 s/d 9) dengan T

Pearson Correlation	
R	T
X <sub>1</sub>	0.292
X <sub>2</sub>	0.032
X <sub>3</sub>	0.020
X <sub>4</sub>	0.320
X <sub>5</sub>	0.138
X <sub>6</sub>	0.412
X <sub>7</sub>	0.298
X <sub>8</sub>	b
X <sub>9</sub>	0.677
T	1

Langkah pertama dalam membuat jaringan syaraf tiruan adalah menguji korelasi (hubungan) antara *input* dengan target yang digunakan untuk membuat jaringan syaraf tiruan. Hal ini dilakukan untuk menyatakan adanya hubungan antara *input* dengan target. Adanya hubungan antara *input* dengan target ditunjukkan oleh nilai koefisien korelasi yang dilambangkan dengan huruf r. Hasil uji korelasi dari 9 buah *input* (X<sub>1</sub> s/d X<sub>9</sub>) dengan masing-masing *input* berjumlah 640 data dan 1 buah target (T) dengan jumlah 640 data yang ditampilkan pada **Tabel 4.1** menunjukkan bahwa terdapat 5 buah *input* yang mempunyai nilai koefisien korelasi dengan interpretasi r Rendah hingga Tinggi berdasarkan pada **Tabel 3.1** terhadap target. 5 buah *input* tersebut adalah *Capacitance* (X<sub>1</sub>), RDC (X<sub>4</sub>), SFRA (X<sub>6</sub>), Tan Delta Belitan (X<sub>7</sub>), *Trafo Life Assesment* (X<sub>9</sub>). Nilai koefisien korelasi masing-masing *input* antara lain: *Capacitance* (X<sub>1</sub>) memiliki nilai koefisien korelasi sebesar 0.292, RDC (X<sub>4</sub>) memiliki nilai koefisien korelasi sebesar 0.320, SFRA (X<sub>6</sub>) memiliki nilai koefisien korelasi sebesar 0.412, Tan Delta Belitan (X<sub>7</sub>) memiliki nilai koefisien korelasi sebesar 0.298, *Trafo Life Assesment* (X<sub>9</sub>) memiliki nilai koefisien korelasi sebesar 0,677. Nilai koefisien korelasi yang dihasilkan

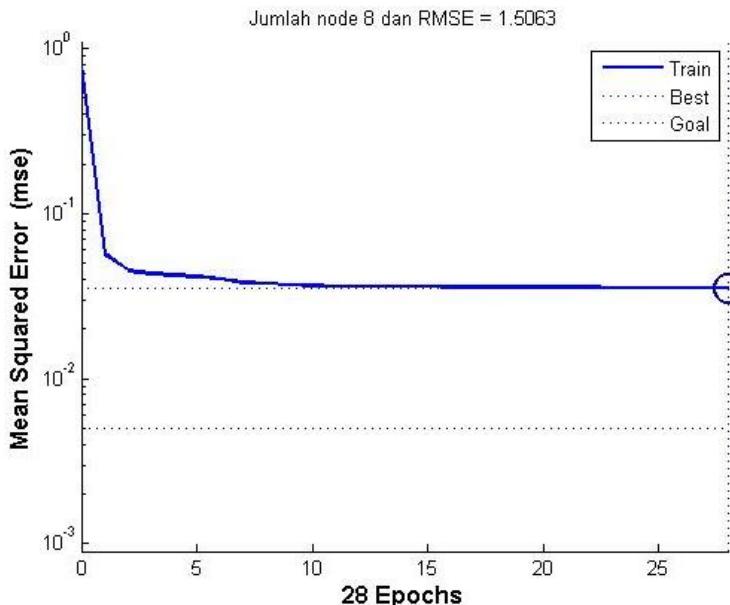
dari uji dipengaruhi oleh keragaman nilai didalam skala interval dari data *input-target* yang diuji.

Uji korelasi *Product Moment Pearson* memiliki ketentuan 2 variabel yang diuji memiliki skala interval. Skala interval dalam penelitian ini yaitu berskala nilai 1 hingga nilai 9. Dari 9 variabel *input* setiap *input* memiliki 640 data yang diuji terhadap 1 variabel target berjumlah 640 data, terdapat 4 variabel *input* yang tidak memiliki keragaman nilai pada interval nilai 1 hingga nilai 9 dalam 640 data tersebut. 4 varibel input itu adalah *Furan*, *Insulation Resistance*, *Ratio*, dan *Dynamic Resistance*. Ketidakadanya keragaman nilai pada interval nilai 1 hingga nilai 9 di 640 data pada 4 variabel tersebut menyebabkan nilai koefisien korelasi tidak mendekati 1 yang ditunjukkan pada **Tabel 4.1**. Ketidakadanya keragaman nilai yang dimaksud adalah sebagai berikut, terdapat 640 data di setiap variabel *input*, dari 640 data itu hanya memiliki 1 nilai yang sama untuk seluruh 640 data, maka dari itu variabel *input* tersebut tidak memiliki keragaman nilai pada skala interval nilai 1 hingga 9 di 640 data dan karena hal itulah maka 4 variabel *input* itu tidak dipilih menjadi variabel input jaringan syaraf tiruan. Nilai koefisien korelasi semakin mendekati 1 maka menunjukkan semakin terdapat korelasi (hubungan) antara *input* dan target. Hal tersebut dikarenakan terdapatnya keragaman nilai pada skala interval nilai 1 hingga nilai 9 untuk 640 data di tiap variabel *input-target* yang memiliki nilai koefisien korelasi mendekati 1. Oleh sebab itulah dipilih 5 buah *input* tersebut yaitu *Capacitance*, *RDC*, *SFRA*, *Tan Delta Belitan*, dan *Trafo Life Assesment* untuk digunakan dalam membuat jaringan syaraf tiruan dengan satu target.

## 4.2 Analisa Hasil *Training* Arsitektur Jaringan Syaraf Tiruan

Pada tahap *training* ini menggunakan sebanyak 480 pasang data *input-target* (75% dari total data) didapatkan hasil performansi *training* JST dengan RMSE paling kecil diantara beberapa perancangan arsitektur yang disebutkan pada subbab 3.4. Hasil performansi tersebut ditunjukkan pada **Gambar 4.1**. Arsitektur JST ini memiliki nilai bobot *hidden layer*, bobot keluaran, bias

*hidden layer*, dan bias keluaran yang ditampilkan di bagian lampiran. Bobot-bobot dan bias-bias tersebut diinisialisasi secara acak oleh *Software* saat proses *training* dimulai. Oleh karena itu, bobot inisial dan bobot akhir selalu berbeda setiap kali program dieksekusi.



**Gambar 4.1.** Hasil *training* arsitektur JST (5-8-1)

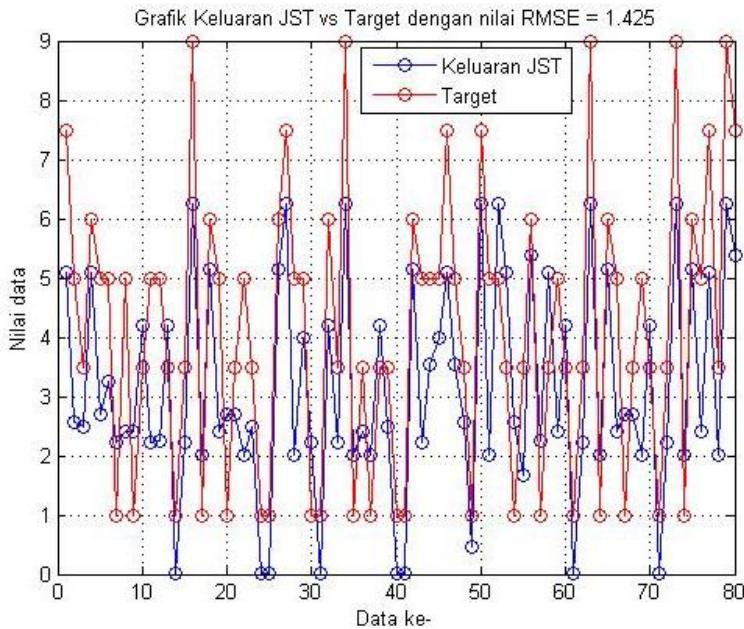
Beberapa perancangan arsitektur yang disebutkan pada subbab 3.4 dilakukan untuk mengetahui apa pengaruh fungsi aktivasi, apa pengaruh jumlah *hidden layer*, apa pengaruh jumlah *hidden node* di *hidden layer*, apa pengaruh jumlah data yang digunakan pada menurunnya nilai RMSE yang dihasilkan dari tahap *training* dan tahap *testing* masing-masing rancangan arsitektur. Berdasarkan pada penelitian yang dilakukan ini fungsi aktivasi tidak berpengaruh untuk menurunkan nilai RMSE tetapi berpengaruh pada peningkatan nilai RMSE tahap *training*. Jumlah *hidden layer* tidak berpengaruh untuk menurunkan nilai RMSE

tahap *training* tetapi berpengaruh pada penurunan nilai RMSE tahap *testing*. Jumlah *hidden node* di *hidden layer* berpengaruh untuk menurunkan nilai RMSE tahap *training* dan tahap *testing*. Jumlah data yang digunakan juga berpengaruh untuk menurunkan nilai RMSE tahap *training* dan tahap *testing*. Pengaruh jumlah *hidden node* di *hidden layer* dan pengaruh jumlah data yang digunakan ditunjukkan dari didapatkannya nilai RMSE pada tahap *training* (**Gambar 4.1**) untuk rancangan arsitektur (urutan yang ke 0 pada subbab **3.4**) 1 *hidden layer* dengan fungsi aktivasi *logsig* (*binary sigmoid*) di *hidden layer*. Fungsi aktivasi *purelin* (identitas) di *output layer*. *Input* berjumlah 5 buah. *Hidden node* di *hidden layer* berjumlah 8 buah dan 1 buah *hidden node* target (5-8-1) serta menggunakan 640 data *input-target*. Maka dari itu digunakanlah arsitektur JST ini untuk tahap *testing* karena memiliki RMSE terkecil diantara variasi perancangan yang dilakukan lainnya.

#### 4.3 Analisa Hasil *Testing* Jaringan Syaraf Tiruan Dengan Secara Otomatis Di Dalam Software

Digunakan arsitektur JST hasil *training* dengan 8 *hidden node* pada *hidden layer* disimpan untuk tahap *testing*. Tahap *testing* dengan cara pertama ini menggunakan sebanyak 80 pasang data *input-target* baru yang berbeda dengan yang digunakan pada tahap *training*. **Gambar 4.2** menampilkan grafik keluaran JST dan grafik target yang seharusnya dicapai. Nilai RMSE yang dihasilkan adalah sebesar 1.425. Nilai RMSE ini menunjukkan bahwa jaringan syaraf tiruan dengan arsitektur 8 buah *hidden node* di *hidden layer* menghasilkan keluaran (mendiagnosis kondisi transformator tenaga) dengan selisih sebesar 1.425 terhadap target data baru. Perbedaan data yang digunakan pada tahap *testing* dengan data yang digunakan pada tahap *training* menyebabkan RMSE yang dicapai oleh JST ini sebesar 1.425. Data yang digunakan pada tahap *testing* dengan cara yang pertama ini tidak digunakan untuk tahap *training* sehingga jaringan syaraf tiruan ini mengeluarkan keluaran berdasarkan dari hasil *training* menggunakan data *training* yang telah dilakukan. Hasil *testing* yang ditunjukkan pada **Gambar 4.2** merupakan hasil *testing*

dengan RMSE terkecil dari hasil *testing* beberapa perancangan arsitektur yang lainnya.

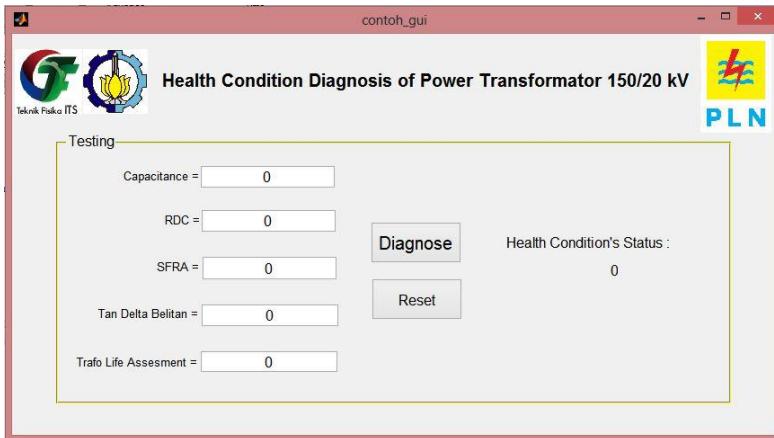


**Gambar 4.2.** Hasil *testing* arsitektur JST (5-8-1) dengan secara otomatis di dalam *software*.

#### 4.4 Analisa Hasil *Testing* Jaringan Syaraf Tiruan Dengan Secara Manual Menggunakan GUI

Gambar 4.3 merupakan tampilan *user interface* dari jaringan JST yang digunakan untuk mendiagnosa kondisi transformator tenaga 150/20 kV. Terdapat 5 buah kolom *input* yang berguna untuk memasukkan 5 hasil pengujian transformator tenaga. Pengujian yang dimaksud adalah pengujian *Capacitance*, pengujian RDC, pengujian SFRA, pengujian Tan Delta Belitan, dan pengujian *Trafo Life Assesment*. Ketika semua kolom telah diisi lalu ditekan tombol Diagnose maka status kondisi transformator tenaga muncul. Arsitektur JST yang digunakan pada *user interface* ini merupakan arsitektur yang samadengen arsitektur

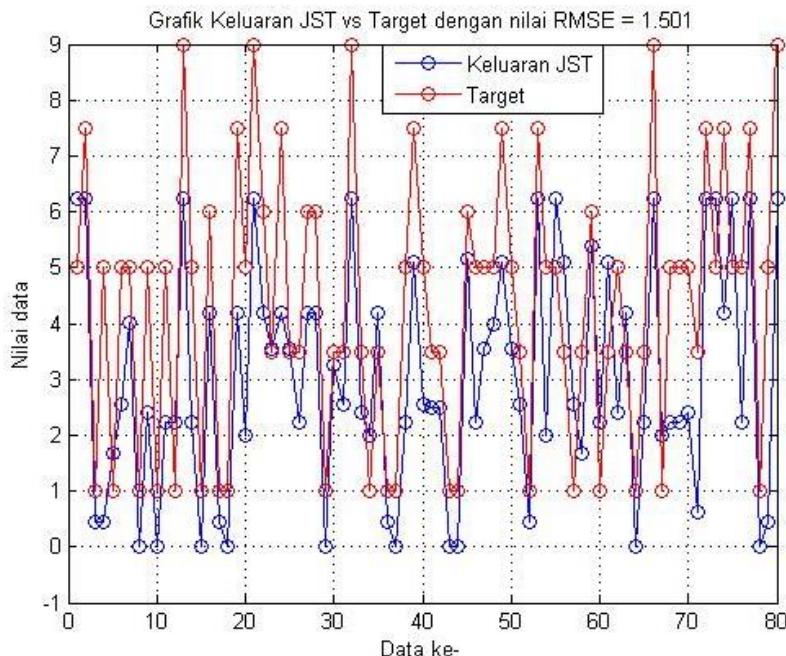
yang memiliki RMSE terkecil pada tahap *training* yaitu arsitektur dengan 8 buah *hidden node* di *hidden layer*. Hal itu dilakukan agar diharapkan *error* yang dihasilkan dari *user interface* ini samadengan *error* pada tahap *training* yaitu ditunjukkan dari nilai RMSE *training*.



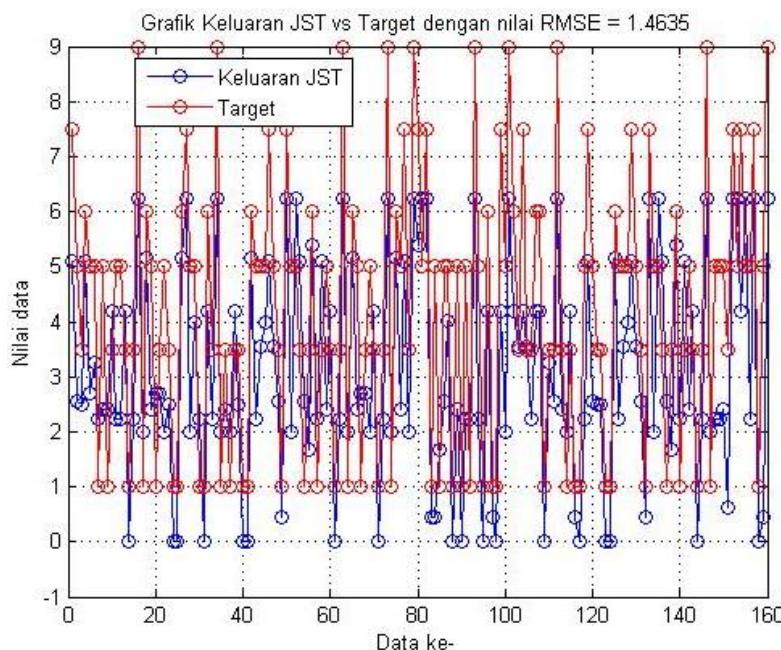
**Gambar 4.3.** Tampilan *user interface*.

Tahap *testing* dengan cara kedua ini menggunakan sebanyak 80 pasang data *input-target*. **Gambar 4.4** menampilkan grafik keluaran JST dan grafik target yang seharusnya dicapai. Nilai RMSE yang dihasilkan adalah sebesar 1.501. Nilai RMSE ini menunjukkan bahwa jaringan syaraf tiruan dengan arsitektur 8 *hidden node* di *hidden layer* menghasilkan keluaran (mendiagnosis kondisi transformator tenaga) dengan selisih sebesar 1.501 terhadap target data baru. Perbedaan data yang digunakan pada tahap *testing* cara kedua dengan data yang digunakan pada tahap *training* menyebabkan nilai RMSE yang dihasilkan pun berbeda. Nilai RMSE yang didapat dari tahap *testing* cara pertama berbeda dengan nilai RMSE yang didapat dari tahap *testing* cara kedua, yaitu 1.425 untuk nilai RMSE *testing* cara pertama dan 1.501 untuk nilai RMSE *testing* cara kedua. Perbedaan ini disebabkan oleh data yang digunakan berbeda antara *testing* cara pertama dengan *testing* cara kedua. Data yang digunakan berbeda tersebut menyebabkan

jaringan yang telah dilatih menggunakan data latih (75% dari total data) mengeluarkan jawaban (*output*) yang berbeda juga. Hal itu ditunjukkan dengan nilai RMSE pada *testing* cara pertama berbeda dengan nilai RMSE pada *testing* cara kedua.



**Gambar 4.4.** Hasil *testing* arsitektur JST (5-8-1) dengan secara manual menggunakan GUI.



**Gambar 4.5.** Hasil *testing* arsitektur JST (5-8-1) menggunakan 25% dari data total dan dengan 5 *Input*.

**Gambar 4.5** menampilkan hasil *testing* menggunakan 25% dari data total yaitu sebanyak 160 data. *Testing* menghasilkan nilai RMSE sebesar 1.4635. Hal ini menunjukkan selisih *output* JST dengan target untuk data yang berbeda dari data yang digunakan untuk tahap *training* adalah sebesar 1.4635. Nilai RMSE yang dihasilkan pada tahap *training*, pada tahap *testing* cara pertama, pada tahap *testing* cara kedua dan pada tahap *testing* dengan 25% data memiliki nilai RMSE adalah sebagai berikut, 1.5063 untuk nilai RMSE *training*, 1.425 untuk nilai RMSE *testing* cara pertama, 1.501 untuk nilai RMSE *testing* cara kedua dan 1.4635 untuk nilai RMSE *testing* dengan 25% data. Dari seluruh nilai RMSE yang dihasilkan tersebut, maka dipilihlah rancangan arsitektur 1 *hidden layer* dengan fungsi aktivasi *logsig* (*binary sigmoid*) di *hidden layer*, fungsi aktivasi *purelin* (*identitas*) di *output layer*, *input*

berjumlah 5 buah, *hidden node* di *hidden layer* berjumlah 8 buah dan 1 buah *hidden node* target (5-8-1) serta menggunakan 640 data *input-target* sebagai arsitektur JST untuk mendiagnosis kondisi transformator tenaga.

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Dari hasil yang diperoleh maka kesimpulan yang didapat adalah sebagai berikut:

- a. Rancangan arsitektur jaringan syaraf tiruan yang menghasilkan kesalahan minimum (RMSE terkecil dari variasi yang dilakukan) dalam mendiagnosis kondisi transformator tenaga adalah rancangan arsitektur 1 *hidden layer* dengan fungsi aktivasi *logsig (binary sigmoid)* di *hidden layer*, fungsi aktivasi *purelin* (identitas) di *output layer*, *input* berjumlah 5 buah. *Hidden node* di *hidden layer* berjumlah 8 buah dan 1 buah *hidden node* target (5-8-1) serta menggunakan 640 data *input-target* menghasilkan 1.5063 untuk nilai RMSE *training*, 1.425 untuk nilai RMSE *testing* dengan secara otomatis di dalam *software*, 1.501 untuk nilai RMSE *testing* dengan secara manual menggunakan GUI dan 1.4635 untuk nilai RMSE *testing* dengan 25% data.

#### 5.2 Saran

Saran yang dapat diberikan untuk penelitian selanjutnya adalah mempelajari terlebih dahulu karakteristik data yang akan digunakan pada jaringan syaraf tiruan sehingga dapat menentukan parameter apa saja yang diperlukan agar mencapai nilai RMSE yang ditentukan karena berbeda kasus berbeda juga cara penanganannya.

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

## DAFTAR PUSTAKA

- [1] A. Jahromi, R. Piercy, S. Cress, J. Service and W. Fan, "An Approach to Power Transformer Asset Management Using Health Index," *Electrical Insulation Magazine*, vol. 25, no. 2, pp. 20-34, 2009.
- [2] P. H. Mukti, F. A. Pamuji and B. S. Munir, "Implementation of Artificial Neural Networks for Determining Power Transformer Condition," *ADCONP*, 2014.
- [3] J. J. Siang, Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB, Yogyakarta: ANDI, 2005.
- [4] U. D. o. t. I. B. o. R. T. S. C. I. S. D. H. R. a. T. S. Group, "Transformers: Basics, Maintenance, and Diagnostics," April 2005.
- [5] <http://www.youtube.com/watch?v=g-RvTjUOENQ>. [Accessed 17 October 2017].
- [6] Buku Pedoman Pemeliharaan Transformator Tenaga, Jakarta: PT PLN (Persero), 2014.
- [7] D. Baptista, "Comparing Different Implementations for the Levenberg-Marquardt," *10th Portuguese Conference on Automatic Control*, 16-18 July 2012.
- [8] <http://digilib.itb.ac.id/files/disk1/555/jbptitbpp-gdl-alfurqonni-27740-3-2007ta-2.pdf>. [Accessed 17 October 2017].
- [9] H. Demuth, M. Beale and M. Hagan, Neural Network Toolbox 6 User's Guide, The MathWorks, Inc., 2004.
- [10] F. Hartono, Prediksi Pencampuran Batubara Menggunakan Pendekatan Jaringan Syaraf Tiruan pada Proses Pembakaran di Pembangkit Listrik Tenaga Uap, Surabaya: Departemen Teknik Fisika FTI ITS, 2017.
- [11] <https://www.skemaku.com/wp-content/uploads/2015/08/prinsip-kerja-transformator.jpg> . [Accessed 17 October 2017].

- [12] R. R. Setiawan and M. H. Purnomo, "Perbandingan Algoritma Levenberg-Marquardt dengan Metoda Backpropagation pada Proses Learning Jaringan Saraf Tiruan untuk Pengenalan Pola Sinyal Elektrokardiograf," *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, 17 June 2006.
- [13] <http://repository.usu.ac.id/bitstream/handle/123456789/54681/Chapter%20II.pdf;jsessionid=98D2629A7D93B2990B84601BCD656BD3?sequence=3>. [Accessed 7 Februari 2018].
- [14] [http://share.its.ac.id/pluginfile.php/14161/mod\\_resource/content/1/JST.ppt](http://share.its.ac.id/pluginfile.php/14161/mod_resource/content/1/JST.ppt). [Accessed 27 April 2018].
- [15] <http://sambas.staf.upi.edu/2013/01/22/tingkat-signifikansi-dan-tingkat-kepercayaan/>. [Accessed 28 April 2018].
- [16] F. and M. Ismanto, "Pemeliharaan On-Load Tap Changer (OLTC) Tranformator Daya PT PLN (Persero) Unit Pelayanan Transmisi (UPT) Banda Aceh," *Seminar Nasional dan Expo Teknik Elektro*, 2015.
- [17] B. and R. M. Hutabarat, "Penilaian Kondisi Transformator Daya pada PT.X," *SINERGI*, vol. 20, no. 3, pp. 175-185, Oktober 2016.
- [18] <http://repository.usu.ac.id/bitstream/handle/123456789/55788/Chapter%20II.pdf;jsessionid=06F8846E8B2AB01C86B7F45D1B746DD7?sequence=3>. [Accessed 27 April 2018].
- [19] <http://digilib.umg.ac.id/download.php?id=5834>. [Accessed 27 April 2018].
- [20] <https://www.energy.siemens.com/us/pool/us/powertransmission/transformers/downloads/TRANSFORMERS-PowerTransfers10to100MVA-1.pdf>. [Accessed 3 Oktober 2017].

# **LAMPIRAN**

*“Halaman ini sengaja dikosongkan”*

Bobot dan bias pada rancangan arsitektur 1 *hidden layer* dengan fungsi aktivasi *logsig* (*binary sigmoid*) di *hidden layer*, fungsi aktivasi *purelin* (identitas) di *output layer*, *input* berjumlah 5 buah. *Hidden node* di *hiddenlayer* berjumlah 8 buah dan 1 buah *hidden node* target (5-8-1) serta menggunakan 640 data *input-target*.

Hidden node (j)	Bobot <i>hidden layer</i> (Vij)					Bias <i>hidden layer</i> (j)	
	Input (i)						
	V1	V2	V3	V4	V5		
1	-2.8903	1.4821	- 5.5714	- 3.3607	- 1.0511	12.0032	
2	3.4508	- 5.4260	- 1.2718	- 4.1039	- 6.0548	-1.3096	
3	-4.7351	- 4.7255	7.1711	- 8.5630	- 0.3374	6.4313	
4	-5.0950	4.9785	2.5231	- 2.6721	- 3.5434	-1.3316	
5	-4.5919	5.7022	2.5550	- 4.9950	- 0.3590	3.0487	
6	4.5629	- 4.7651	- 7.0187	- 3.1918	- 4.6126	8.2908	
7	3.5666	- 5.5816	1.0814	- 3.2414	- 5.6451	3.6941	
8	3.0792	- 5.2938	- 5.3432	0.4165	2.8648	-0.1197	
Bobot <i>output layer</i> (Wjk)							
Hidden node (j)	Output node (k)					Bias <i>output layer</i>	
	Y					-0.1117	
1	1.0584						
2	1.2746						
3	-0.9869						
4	1.1837						
5	-1.1191						
6	-0.2477						
7	0.7620						
8	-1.6533						

Hasil dari rancangan arsitektur lainnya sebagai berikut (warna kuning menandakan RMSE terkecil, *testing 1* = *test* dengan cara pertama, *testing 2* = *test* dengan cara kedua, *testing all* = *test* dengan 25% dari data total):

- a. 1 *hidden layer* dengan fungsi aktivasi purelin (identitas). Jumlah *hidden node* mulai dari 1 hingga 10, menggunakan 170 data.

Arsitektur	RMSE <i>training</i>	Epoch	RMSE		
			<i>testing 1</i>	<i>testing 2</i>	<i>testing all</i>
5-1-1	1.5358	5			
5-2-1	1.5358	6			
5-3-1	1.5358	5			
5-4-1	1.5358	4			
5-5-1	1.5358	6			
5-6-1	1.5358	4			
5-7-1	1.5358	3	2.0249	2.9837	2.1516
5-8-1	1.5358	3			
5-9-1	1.5358	4			
5-10-1	1.5358	3			

- b. 1 *hidden layer* dengan fungsi aktivasi purelin (identitas). Jumlah *hidden node* mulai dari 1 hingga 20, menggunakan 170 data.

Arsitektur	RMSE <i>training</i>	Epoch	RMSE		
			<i>testing 1</i>	<i>testing 2</i>	<i>testing all</i>
5-1-1	1.5358	5			
5-2-1	1.5358	5			
5-3-1	1.5358	5			
5-4-1	1.5358	5			
5-5-1	1.5358	5			
5-6-1	1.5358	4			
5-7-1	1.5358	4			
5-8-1	1.5358	3	2.0786	2.9608	2.0669
5-9-1	1.5358	4			
5-10-1	1.5358	4			
5-11-1	1.5358	4			
5-12-1	1.5358	3			
5-13-1	1.5358	4			
5-14-1	1.5358	4			
5-15-1	1.5358	5			
5-16-1	1.5358	3			
5-17-1	1.5358	4			
5-18-1	1.5358	3			
5-19-1	1.5358	3			
5-20-1	1.5358	3			

- c. 2 *hidden layer* dengan fungsi aktivasi *hidden layer* pertama dan kedua adalah *logsig (binary sigmoid)*. Jumlah *hidden node* di *hidden layer* pertama mulai dari 1 hingga 10 sedangkan jumlah *hidden node* di *hidden layer* kedua tetap 1 buah, menggunakan 170 data.

Arsitektur	RMSE <i>training</i>	Epoch	RMSE <i>testing all</i>
5-1-1-1	5.1855	0	5.1855
5-2-1-1	1.6947	15	1.6947
5-3-1-1	1.6467	8	1.6467
5-4-1-1	1.6099	15	1.6099
5-5-1-1	1.8101	9	1.8101
5-6-1-1	1.6573	15	1.6573
5-7-1-1	1.7776	8	1.7776
5-8-1-1	2.3881	8	2.3881
5-9-1-1	1.6395	11	1.6395
5-10-1-1	1.7426	12	1.7426

- d. 2 *hidden layer* dengan fungsi aktivasi *hidden layer* pertama dan kedua adalah *logsig (binary sigmoid)*. Jumlah *hidden node* di *hidden layer* pertama mulai dari 1 hingga 10 sedangkan jumlah *hidden node* di *hidden layer* kedua tetap 2 buah, menggunakan 170 data.

Arsitektur	RMSE <i>training</i>	Epoch	RMSE <i>testing all</i>
5-1-2-1	5.1855	0	5.1855
5-2-2-1	1.6824	34	1.6824
5-3-2-1	1.5918	12	1.5918
5-4-2-1	1.8092	11	1.8092
5-5-2-1	1.6539	10	1.6539
5-6-2-1	1.6289	9	1.6289
5-7-2-1	1.5809	35	1.5809
5-8-2-1	2.2418	17	2.2418
5-9-2-1	1.6791	19	1.6791
5-10-2-1	1.9523	12	1.9523

- e. 2 *hidden layer* dengan fungsi aktivasi *hidden layer* pertama dan kedua adalah *logsig (binary sigmoid)*. Jumlah *hidden node* di *hidden layer* pertama mulai dari 1 hingga 10 sedangkan jumlah *hidden node* di *hidden layer* kedua tetap 3 buah, menggunakan 170 data.

Arsitektur	RMSE <i>training</i>	Epoch	RMSE <i>testing all</i>
5-1-3-1	5.1855	0	5.1855
5-2-3-1	1.6826	100	1.6826
5-3-3-1	1.6851	17	1.6851
5-4-3-1	1.6866	11	1.6866
5-5-3-1	1.687	9	1.687
<b>5-6-3-1</b>	<b>1.5864</b>	<b>21</b>	<b>1.5864</b>
5-7-3-1	1.7261	9	1.7261
5-8-3-1	1.7174	9	1.7174
5-9-3-1	1.7611	9	1.7611
5-10-3-1	1.647	10	1.647

- f. 2 *hidden layer* dengan fungsi aktivasi *hidden layer* pertama dan kedua adalah *logsig (binary sigmoid)*. Jumlah *hidden node* di *hidden layer* pertama mulai dari 1 hingga 10 sedangkan jumlah *hidden node* di *hidden layer* kedua tetap 4 buah, menggunakan 170 data.

Arsitektur	RMSE <i>training</i>	Epoch	RMSE <i>testing all</i>
5-1-4-1	5.1855	0	5.1855
5-2-4-1	1.6669	15	1.6669
5-3-4-1	1.6652	15	1.6652
5-4-4-1	1.6855	13	1.6855
5-5-4-1	1.6248	16	1.6248
5-6-4-1	1.7126	10	1.7126
<b>5-7-4-1</b>	<b>1.5543</b>	<b>17</b>	<b>1.5543</b>
5-8-4-1	1.8343	8	1.8343
5-9-4-1	1.5654	14	1.5654
5-10-4-1	1.5819	12	1.5819

- g. 2 *hidden layer* dengan fungsi aktivasi *hidden layer* pertama dan kedua adalah *logsig (binary sigmoid)*. Jumlah *hidden node* di *hidden layer* pertama mulai dari 1 hingga 10

sedangkan jumlah *hidden node* di *hidden layer* kedua tetap 5 buah, menggunakan 170 data.

Arsitektur	RMSE <i>training</i>	Epoch	RMSE <i>testing all</i>
5-1-5-1	5.1855	0	5.1855
5-2-5-1	1.7956	9	1.7956
5-3-5-1	1.6784	11	1.6784
5-4-5-1	1.85	7	1.85
5-5-5-1	1.7154	13	1.7154
5-6-5-1	1.6631	9	1.6631
5-7-5-1	1.5827	12	1.5827
5-8-5-1	1.6127	12	1.6127
5-9-5-1	1.6505	9	1.6505
5-10-5-1	1.7139	10	1.7139

- h. 2 *hidden layer* dengan fungsi aktivasi *hidden layer* pertama dan kedua adalah *logsig (binary sigmoid)*. Jumlah *hidden node* di *hidden layer* pertama mulai dari 1 hingga 10 sedangkan jumlah *hidden node* di *hidden layer* kedua tetap 6 buah, menggunakan 170 data.

Arsitektur	RMSE <i>training</i>	Epoch	RMSE <i>testing all</i>
5-1-6-1	5.1855	0	5.1855
5-2-6-1	1.6441	21	1.6441
5-3-6-1	1.6292	10	1.6292
5-4-6-1	1.699	8	1.699
5-5-6-1	1.8122	9	1.8122
5-6-6-1	1.7587	9	1.7587
5-7-6-1	1.5491	12	1.5491
5-8-6-1	1.6458	9	1.6458
5-9-6-1	2.0836	8	2.0836
5-10-6-1	1.5969	14	1.5969

- i. 2 *hidden layer* dengan fungsi aktivasi *hidden layer* pertama dan kedua adalah *logsig (binary sigmoid)*. Jumlah *hidden node* di *hidden layer* pertama mulai dari 1 hingga 10 sedangkan jumlah *hidden node* di *hidden layer* kedua tetap 7 buah, menggunakan 170 data.

Arsitektur	RMSE <i>training</i>	Epoch	RMSE <i>testing all</i>
5-1-7-1	5.1855	0	5.1855
5-2-7-1	1.6865	13	1.6865
5-3-7-1	1.7076	11	1.7076
5-4-7-1	1.6317	13	1.6317
5-5-7-1	1.7142	11	1.7142
5-6-7-1	1.6889	9	1.6889
<b>5-7-7-1</b>	<b>1.5397</b>	<b>7</b>	<b>1.5397</b>
5-8-7-1	1.7108	9	1.7108
5-9-7-1	1.6528	12	1.6528
5-10-7-1	1.8375	8	1.8375

- j. 2 *hidden layer* dengan fungsi aktivasi *hidden layer* pertama dan kedua adalah *logsig (binary sigmoid)*. Jumlah *hidden node* di *hidden layer* pertama mulai dari 1 hingga 10 sedangkan jumlah *hidden node* di *hidden layer* kedua tetap 8 buah, menggunakan 170 data.

Arsitektur	RMSE <i>training</i>	Epoch	RMSE <i>testing all</i>
5-1-8-1	5.1855	0	5.1855
5-2-8-1	1.698	14	1.698
<b>5-3-8-1</b>	<b>1.5593</b>	<b>33</b>	<b>1.5593</b>
5-4-8-1	1.7076	9	1.7076
5-5-8-1	1.5966	11	1.5966
5-6-8-1	1.6478	10	1.6478
5-7-8-1	1.5992	10	1.5992
5-8-8-1	1.6476	9	1.6476
5-9-8-1	1.6803	12	1.6803
5-10-8-1	1.6344	12	1.6344

- k. 2 *hidden layer* dengan fungsi aktivasi *hidden layer* pertama dan kedua adalah *logsig (binary sigmoid)*. Jumlah *hidden node* di *hidden layer* pertama mulai dari 1 hingga 10 sedangkan jumlah *hidden node* di *hidden layer* kedua tetap 9 buah, menggunakan 170 data.

Arsitektur	RMSE <i>training</i>	Epoch	RMSE <i>testing all</i>
5-1-9-1	5.1855	0	5.1855
5-2-9-1	1.6978	8	1.6978
5-3-9-1	1.8007	9	1.8007
5-4-9-1	2.2068	8	2.2068
5-5-9-1	1.6916	10	1.6916
5-6-9-1	1.6458	12	1.6458
5-7-9-1	1.5837	8	1.5837
5-8-9-1	1.6142	14	1.6142
5-9-9-1	1.5847	8	1.5847
5-10-9-1	1.6152	9	1.6152

- l. 2 *hidden layer* dengan fungsi aktivasi *hidden layer* pertama dan kedua adalah *logsig (binary sigmoid)*. Jumlah *hidden node* di *hidden layer* pertama mulai dari 1 hingga 10 sedangkan jumlah *hidden node* di *hidden layer* kedua tetap 10 buah, menggunakan 170 data.

Arsitektur	RMSE <i>training</i>	Epoch	RMSE <i>testing all</i>
5-1-10-1	5.1855	0	5.1855
5-2-10-1	1.5736	34	1.5736
5-3-10-1	1.6767	9	1.6767
5-4-10-1	1.7352	10	1.7352
5-5-10-1	1.8884	10	1.8884
5-6-10-1	1.6596	9	1.6596
5-7-10-1	1.6506	12	1.6506
5-8-10-1	1.5591	15	1.5591
5-9-10-1	1.75	9	1.75
5-10-10-1	2.0002	8	2.0002

- m. 1 *hidden layer* dengan fungsi aktivasi *logsig (binary sigmoid)*. Jumlah *hidden node* mulai dari 1 hingga 10, menggunakan 170 data.

Arsitektur	RMSE training	Epoch	RMSE		
			testing 1	testing 2	testing all
5-1-1	1.5338	50			
5-2-1	1.4466	50			
5-3-1	1.4246	50			
5-4-1	1.4136	50			
5-5-1	1.4082	50			
5-6-1	1.4071	50			
5-7-1	1.3995	27	2.0399	2.2359	2.5782
5-8-1	1.3995	48			
5-9-1	1.3995	21			
5-10-1	1.3995	21			

- n. 1 *hidden layer* dengan fungsi aktivasi *logsig (binary sigmoid)*. Jumlah *hidden node* mulai dari 1 hingga 10, menggunakan 320 data.

Arsitektur	RMSE training	Epoch	RMSE		
			testing 1	testing 2	testing all
5-1-1	1.6805	50			
5-2-1	1.6356	50			
5-3-1	1.5714	50			
5-4-1	1.5491	50			
5-5-1	1.5377	50			
5-6-1	1.4964	50			
5-7-1	1.4959	50			
5-8-1	1.4956	19	1.5289	1.7131	1.6236
5-9-1	1.4956	44			
5-10-1	1.4956	26			

- o. 1 *hidden layer* dengan fungsi aktivasi *logsig (binary sigmoid)*. Jumlah *hidden node* mulai dari 1 hingga 10, menggunakan 640 data.

Arsitektur	RMSE <i>training</i>	Epoch	RMSE		
			<i>testing 1</i>	<i>testing 2</i>	<i>testing all</i>
5-1-1	1.7103	50			
5-2-1	1.6218	50			
5-3-1	1.6016	50			
5-4-1	1.5175	50			
5-5-1	1.5338	50			
5-6-1	1.5203	50			
5-7-1	1.5063	50			
5-8-1	1.5063	28	1.425	1.501	1.4635
5-9-1	1.5074	50			
5-10-1	1.5063	19			

- p. 1 *hiddenlayer* dengan fungsi aktivasi *logsig* (*binary sigmoid*). Jumlah *hidden node* mulai dari 1 hingga 10, menggunakan 960 data.

Arsitektur	RMSE <i>training</i>	Epoch	RMSE		
			<i>testing 1</i>	<i>testing 2</i>	<i>testing all</i>
5-1-1	1.7005	100			
5-2-1	1.6761	100			
5-3-1	1.5482	82			
5-4-1	1.5297	100			
5-5-1	1.5024	100			
5-6-1	1.5038	100			
5-7-1	1.5024	38	1.4464	1.4901	1.4684
5-8-1	1.5024	41			
5-9-1	1.5024	24			
5-10-1	1.5024	24			

- q. 1 *hidden layer* dengan fungsi aktivasi *tansig* (*bipolar sigmoid*). Jumlah *hidden node* mulai dari 1 hingga 10, menggunakan 170 data.

Arsitektur	RMSE training	Epoch	RMSE		
			testing 1	testing 2	testing all
5-1-1	1.5336	100			
5-2-1	1.4799	100			
5-3-1	1.4474	100			
5-4-1	1.4047	100			
5-5-1	1.4025	100			
5-6-1	1.3995	34	2.0249	2.9837	2.1516
5-7-1	1.3995	15			
5-8-1	1.3995	54			
5-9-1	1.3995	20			
5-10-1	1.3995	18			

- r. 1 hidden layer dengan fungsi aktivasi *logsig (binary sigmoid)*. Jumlah hidden node mulai dari 1 hingga 20, menggunakan 170 data.

Arsitektur	RMSE training	Epoch	RMSE		
			testing 1	testing 2	testing all
5-1-1	1.5335	100			
5-2-1	1.4771	100			
5-3-1	1.4098	100			
5-4-1	1.4208	100			
5-5-1	1.404	100			
5-6-1	1.3995	50	3.0814	2.9272	2.7669
5-7-1	1.3995	79			
5-8-1	1.3995	40			
5-9-1	1.3995	15			
5-10-1	1.3995	33			
5-11-1	1.3995	15			
5-12-1	1.3995	11			
5-13-1	1.3995	9			
5-14-1	1.3995	31			
5-15-1	1.3995	13			
5-16-1	1.3995	8			
5-17-1	1.3995	9			
5-18-1	1.3995	7			
5-19-1	1.3995	6			
5-20-1	1.3995	7			

Hasil dari rancangan arsitektur dengan 2 *input* menggunakan perancangan arsitektur JST **Tabel 3.2** sebagai berikut ditampilkan RMSE terkecil, *testing 1 = test* dengan cara pertama, *testing 2 = test* dengan cara kedua, *testing all = test* dengan 25% dari data total):

No	Epoch	Arsitektur	RMSE			
			Training	Testing-1	Testing-2	Testing-all
a	3	2-9-1	1.616	2.1129	2.9115	2.068
b	6	2-6-1	1.616	2.1129	2.9115	2.068
c	9	2-3-1-1		1.7142		
d	10	2-7-2-1		1.7137		
e	2	2-2-3-1		1.714		
f	7	2-7-4-1		1.722		
g	7	2-5-5-1		1.7199		
h	5	2-9-6-1		1.7178		
i	4	2-7-7-1		1.7132		
j	5	2-8-8-1		1.7211		
k	5	2-6-9-1		1.7181		
l	4	2-8-10-1		1.734		
m	16	2-2-1	1.6065	2.1487	2.978	2.1135
n	38	2-5-1	1.7149	1.7752	1.8871	1.832
o	4	2-8-1	1.7328	1.6749	1.744	1.7098
p	25	2-4-1	1.7282	1.7005	1.7254	1.713
q	91	2-2-1	1.6065	2.1487	2.978	2.1135
r	4	2-11-1	1.6065	2.1487	2.978	2.1135

### Script software JST

#### A. Script training JST

```
clc;clear;close all;warning off;
```

```
% Proses membaca data latih dari excel
filename = 'DATA-TA.xlsx';
sheet = 1;
xlRange = 'O4:T131';
```

```
Data = xlsread(filename, sheet, xlRange);
```

```
Cpt = Data(:,1);
```

```

Rdc = Data(:,2);
Sfra = Data(:,3);
Tdb = Data(:,4);
Tla = Data(:,5);
Trgt = Data(:,6);

%normalisasi data dibuat agar dalam rentang 0 sampai 1
Cptnew = (Cpt-min(Cpt))/(max(Cpt)-min(Cpt));
Rdcnew = (Rdc-min(Rdc))/(max(Rdc)-min(Rdc));
Sfranew = (Sfra-min(Sfra))/(max(Sfra)-
min(Rdc));
Tdbnew = (Tdb-min(Tdb))/(max(Tdb)-min(Tdb));
Tlanew = (Tla-min(Tla))/(max(Tla)-min(Tla));
Trgtnew = (Trgt-min(Trgt))/(max(Trgt)-
min(Trgt));
Data = [Cptnew Rdcnew Sfranew Tdbnew Tlanew
Trgtnew];

%inisialisasi data input dan target
data_latih = Data(:,1:5)';
target_latih = Data(:,6)';
[m,n] = size(data_latih);

for jumlahnode=1:10
% Pembuatan JST
net = newff(minmax(data_latih),[jumlahnode
1],{'logsig','purelin'},'trainlm');
% Fungsi aktivasi HiddenLayer memakai logsig
karena range data [0,1]
% Fungsi aktivasi OutputLayer memakai purelin
karena supaya nilainya
% tidak berubah

% Memberikan nilai untuk mempengaruhi proses pelatihan
net.performFcn = 'mse';
net.trainParam.goal = 0.005;
%rmse==sqrt(mse)
net.trainParam.lr = 0.01; %default 0.01

```

```

net.trainParam.epochs = 100;

% Proses training
[net_keluaran,tr,Y,E] =
train(net,data_latih,target_latih);

% Hasil setelah pelatihan
bobot_hidden = net_keluaran.IW{1,1};
bobot_keluaran = net_keluaran.LW{2,1};
bias_hidden = net_keluaran.b{1,1};
bias_keluaran = net_keluaran.b{2,1};
jumlah_iterasi = tr.num_epochs;
nilai_keluaran = Y;
nilai_error = E;
error_MSE = (1/n)*sum(nilai_error.^2);
error_RMSE = sqrt(error_MSE)*(9-1);

if(jumlahnode==1)
    RMSE_best=error_RMSE;
    node_best=1;
else
    if(error_RMSE<RMSE_best)
        RMSE_best=error_RMSE;
        node_best=jumlahnode;
    end
end
filename=sprintf('net%d.mat',
jumlahnode);
save(filename,'net_keluaran');

%menampilkan plot grafik
performansi (RMSE)
figure,
plotperform(tr)
title(strcat(['Jumlah node
',num2str(jumlahnode), ' dan RMSE =
',num2str(error_RMSE)]))

```

```
    end
```

## B. Script testing JST

```
clc;clear;

% load jaringan yang sudah dibuat pada proses
% pelatihan
load netbest.mat

% Proses membaca data uji dari excel
filename = 'DATA-TA.xlsx';
sheet = 1;
xlRange = 'O132:T173';

Data = xlsread(filename, sheet, xlRange);

Cpt = Data(:,1);
Rdc = Data(:,2);
Sfra = Data(:,3);
Tdb = Data(:,4);
Tla = Data(:,5);
Trgt = Data(:,6);

%normalisasi data
Cptnew = (Cpt-min(Cpt))/(max(Cpt)-min(Cpt));
Rdcnew = (Rdc-min(Rdc))/(max(Rdc)-min(Rdc));
Sfranew = (Sfra-min(Sfra))/(max(Sfra)-
min(Sfra));
Tdbnew = (Tdb-min(Tdb))/(max(Tdb)-min(Tdb));
Tlanew = (Tla-min(Tla))/(max(Tla)-min(Tla));
Trgtnew = (Trgt-min(Trgt))/(max(Trgt)-
min(Trgt));
Data = [Cptnew Rdcnew Sfranew Tdbnew Tlanew
Trgtnew];

data_uji = Data(:,1:5)';
target_uji = Data(:,6)';
```

```

[m,n] = size(data_uji);

% Hasil Diagnosis
hasil_uji = sim(net_keluaran,data_uji);
nilai_error = hasil_uji-target_uji;

hasil_ujibaru = hasil_uji*(9-1);

% Performansi hasil diagnosis
error_MSE = (1/n)*sum(nilai_error.^2);
error_RMSE = sqrt(error_MSE)*(9-1);

filename = 'DATA-TA.xlsx';
sheet = 1;
xlRange = 'T132:T173';

target_uji_asli = xlsread(filename, sheet,
xlRange);

figure,
plot(hasil_ujibaru,'bo-')
hold on
plot(target_uji_asli,'ro-')
hold off
grid on
title(strcat(['Grafik Keluaran JST vs Target
dengan nilai RMSE = ',...
num2str(error_RMSE)]))
xlabel('Data ke-')
ylabel('Nilai data')
legend('Keluaran
JST', 'Target', 'Location', 'Best')

```

### C. Script GUI JST

```

function varargout = contoh_gui(varargin)
% CONTOH_GUI MATLAB code for contoh_gui.fig
%           CONTOH_GUI, by itself, creates a new
CONTOH_GUI or raises the existing
%           singleton*.

```

```

%
%      H = CONTOH_GUI returns the handle to a
new CONTOH_GUI or the handle to
%      the existing singleton*.
%
%
CONTOH_GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in
CONTOH_GUI.M with the given input arguments.
%
%      CONTOH_GUI('Property','Value',...) creates a new CONTOH_GUI or raises
%      the existing singleton*. Starting from the left, property value pairs are
%      applied to the GUI before contoh_gui_OpeningFcn gets called. An
%      unrecognized property name or invalid value makes property application
%      stop. All inputs are passed to contoh_gui_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help contoh_gui

% Last Modified by GUIDE v2.5 30-Apr-2018
20:14:49

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',
filename, ...
%           'gui_Singleton',
gui_Singleton, ...

```

```

                'gui_OpeningFcn',
@contoh_gui_OpeningFcn, ...
                'gui_OutputFcn',
@contoh_gui_OutputFcn, ...
                'gui_LayoutFcn', [], ...
                'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback =
str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] =
gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before contoh_gui is made
% visible.
function contoh_gui_OpeningFcn(hObject,
 eventdata, handles, varargin)
% This function has no output args, see
OutputFcn.
% hObject    handle to figure
% eventdata   reserved - to be defined in a
future version of MATLAB
% handles    structure with handles and user
data (see GUIDATA)
% varargin    command line arguments to
contoh_gui (see VARARGIN)

% Choose default command line output for
contoh_gui
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

```

```
myImage = imread('Logo_PLN.png');
axes(handles.axes4);
imshow(myImage);

myImage = imread('logo-tf-web-fix.jpg');
axes(handles.axes5);
imshow(myImage);

myImage = imread('lambang-its-png-v1.png');
axes(handles.axes6);
imshow(myImage);

initialize_gui(hObject, handles, false);

% UIWAIT makes contoh_gui wait for user
response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned
to the command line.
function varargout =
contoh_gui_OutputFcn(hObject, eventdata,
handles)
% varargout cell array for returning output
args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a
future version of MATLAB
% handles structure with handles and user
data (see GUIDATA)

% Get default command line output from
handles structure
varargout{1} = handles.output;
```

```

% --- Executes during object creation, after
% setting all properties.
function data1_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to data1 (see GCBO)
% eventdata   reserved - to be defined in a
future version of MATLAB
% handles    empty - handles not created
until after all CreateFcns called

% Hint: popupmenu controls usually have a
white background on Windows.
%         See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function data1_Callback(hObject, eventdata,
handles)
% hObject    handle to data1 (see GCBO)
% eventdata   reserved - to be defined in a
future version of MATLAB
% handles    structure with handles and user
data (see GUIDATA)

% Hints: get(hObject,'String') returns
contents of data1 as text
%         str2double(get(hObject,'String'))
returns contents of data1 as a double

data1 = str2double(get(hObject, 'String'));
if isnan(data1)
    set(hObject, 'String', 0);
    errordlg('Input must be a
number', 'Error');

```

```

end

% Save the new data1 value
handles.dt.data1 = data1;
guidata(hObject, handles)

% --- Executes during object creation, after
% setting all properties.
function data2_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to data2 (see GCBO)
% eventdata   reserved - to be defined in a
future version of MATLAB
% handles    empty - handles not created
until after all CreateFcns called

% Hint: popupmenu controls usually have a
white background on Windows.
%         See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end


function data2_Callback(hObject, eventdata,
handles)
% hObject    handle to data2 (see GCBO)
% eventdata   reserved - to be defined in a
future version of MATLAB
% handles    structure with handles and user
data (see GUIDATA)

% Hints: get(hObject,'String') returns
contents of data2 as text
%         str2double(get(hObject, 'String'))
returns contents of data2 as a double

```

```

data2 = str2double(get(hObject, 'String'));
if isnan(data2)
    set(hObject, 'String', 0);
    errordlg('Input must be a
number','Error');
end

% Save the new data2 value
handles.dt.data2 = data2;
guidata(hObject,handles)

% --- Executes on button press in calculate.
function calculate_Callback(hObject,
 eventdata, handles)
% hObject    handle to calculate (see GCBO)
% eventdata   reserved - to be defined in a
future version of MATLAB
% handles    structure with handles and user
data (see GUIDATA)

%hasil = handles.metricdata.data1 *
handles.metricdata.data2;

load netbest.mat;

data_uji = [ (handles.dt.data1)/(9-
1);(handles.dt.data2)/(9-1);
(handles.dt.data3)/(9-1);
(handles.dt.data4)/(9-1);
(handles.dt.data5)/(9-1) ];

% Hasil diagnosis
hasil_uji = sim(net_keluaran,data_uji);

%dikali disik
out = hasil_uji*(9-1);

%mlebu if
if (out == 1)

```

```

        status = 'Buruk';
elseif ((out >= 7)&&(out <= 9))
    status = 'Baik';
elseif ((out > 1)&&(out < 7))
    status = 'Normal';

end

%mlebu ngisor
set(handles.hasil_uji, 'String', out);
set(handles.text19, 'string', status);

% --- Executes on button press in reset.
function reset_Callback(hObject, eventdata,
handles)
% hObject      handle to reset (see GCBO)
% eventdata     reserved - to be defined in a
future version of MATLAB
% handles       structure with handles and user
data (see GUIDATA)

initialize_gui(gcbf, handles, true);

% -----
-----
function initialize_gui(fig_handle, handles,
isreset)
% If the metricdata field is present and the
reset flag is false, it means
% we are we are just re-initializing a GUI by
calling it from the cmd line
% while it is up. So, bail out as we dont
want to reset the data.
if isfield(handles, 'dt') && ~isreset
    return;
end

handles.dt.data1 = 0; % Capacitance

```

```

handles.dt.data2 = 0; % RDC
handles.dt.data3 = 0; % SFRA
handles.dt.data4 = 0; % Tan Delta Belitan
handles.dt.data5 = 0; % Trafo Life Assesment

set(handles.data1, 'String',
handles.dt.data1); % Capacitance
set(handles.data2, 'String',
handles.dt.data2); % RDC
set(handles.data3, 'String',
handles.dt.data3); % SFRA
set(handles.data4, 'String',
handles.dt.data4); % Tan Delta Belitan
set(handles.data5, 'String',
handles.dt.data5); % Trafo LIfe Assesment

set(handles.hasil_uji, 'String',0);
set(handles.text19, 'String','');

% Update handles structure
guidata(handles.figure1, handles);

function data3_Callback(hObject, eventdata,
handles)
% hObject    handle to data3 (see GCBO)
% eventdata   reserved - to be defined in a
future version of MATLAB
% handles    structure with handles and user
data (see GUIDATA)

% Hints: get(hObject,'String') returns
contents of data3 as text
%         str2double(get(hObject,'String'))
returns contents of data3 as a double
data3 = str2double(get(hObject, 'String'));
if isnan(data3)

```

```
    set(hObject, 'String', 0);
    errordlg('Input must be a
number','Error');
end

% Save the new data1 value
handles.dt.data3 = data3;
guidata(hObject,handles)

% --- Executes during object creation, after
% setting all properties.
function data3_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to data3 (see GCBO)
% eventdata   reserved - to be defined in a
future version of MATLAB
% handles    empty - handles not created
until after all CreateFcns called

% Hint: edit controls usually have a white
background on Windows.
%         See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function data4_Callback(hObject, eventdata,
handles)
% hObject    handle to data4 (see GCBO)
% eventdata   reserved - to be defined in a
future version of MATLAB
% handles    structure with handles and user
data (see GUIDATA)
```

```

% Hints: get(hObject,'String') returns
contents of data4 as text
%           str2double(get(hObject,'String'))
returns contents of data4 as a double
data4 = str2double(get(hObject, 'String'));
if isnan(data4)
    set(hObject, 'String', 0);
    errordlg('Input must be a
number','Error');
end

% Save the new data1 value
handles.dt.data4 = data4;
guidata(hObject,handles)

% --- Executes during object creation, after
setting all properties.
function data4_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to data4 (see GCBO)
% eventdata   reserved - to be defined in a
future version of MATLAB
% handles      empty - handles not created
until after all CreateFcns called

% Hint: edit controls usually have a white
background on Windows.
%         See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function data5_Callback(hObject, eventdata,
handles)

```

```

% hObject      handle to data5 (see GCBO)
% eventdata    reserved - to be defined in a
% future version of MATLAB
% handles      structure with handles and user
% data (see GUIDATA)

% Hints: get(hObject,'String') returns
% contents of data5 as text
%         str2double(get(hObject,'String'))
% returns contents of data5 as a double
data5 = str2double(get(hObject, 'String'));
if isnan(data5)
    set(hObject, 'String', 0);
    errordlg('Input must be a
              number','Error');
end

% Save the new data1 value
handles.dt.data5 = data5;
guidata(hObject,handles)

% --- Executes during object creation, after
% setting all properties.
function data5_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to data5 (see GCBO)
% eventdata    reserved - to be defined in a
% future version of MATLAB
% handles      empty - handles not created
% until after all CreateFcns called

% Hint: edit controls usually have a white
% background on Windows.
%         See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');


```

```
end
```

```
% --- Executes during object creation, after
% setting all properties.
function hasil_uji_CreateFcn(hObject,
 eventdata, handles)
% hObject    handle to hasil_uji (see GCBO)
% eventdata   reserved - to be defined in a
future version of MATLAB
% handles    empty - handles not created
until after all CreateFcns called
```

```
% --- Executes during object creation, after
% setting all properties.
function uipanel1_CreateFcn(hObject,
 eventdata, handles)
% hObject    handle to uipanel1 (see GCBO)
% eventdata   reserved - to be defined in a
future version of MATLAB
% handles    empty - handles not created
until after all CreateFcns called
```

## BIODATA PENULIS



Nama lengkap penulis adalah **Guntur Putra Pamungkas**, lahir pada tanggal 30 Mei 1995 di Karanganyar, merupakan anak ke dua dari dua bersaudara. Penulis menempuh pendidikan sekolah dasar di MI Muhammadiyah Karanganyar, pendidikan menengah pertama di SMP Negeri 1 Karanganyar dan melanjutkan pendidikan menengah atas di SMA Negeri 4 Surakarta kemudian menempuh pendidikan Diploma Tiga pada program DIII Metrologi dan Instrumentasi FTI ITB. Selanjutnya menempuh pendidikan Sarjana Satu Lintas Jalur di ITS Departemen Teknik Fisika pada tahun 2016 dengan NRP 02311645000010. Pada tahun 2018 penulis akhirnya melaksanakan Tugas Akhir sebagai syarat untuk menyelesaikan pendidikan S1 dengan judul "**Diagnosis Kondisi Transformator Tenaga Menggunakan Pendekatan Jaringan Syaraf Tiruan di Perusahaan Listrik Negara Transmisi Jawa Bagian Timur dan Bali**". Bagi pembaca yang memiliki saran, kritik, atau ingin berdiskusi lebih lanjut tentang Tugas Akhir ini, maka dapat menghubungi penulis melalui [gunturppamungkas@gmail.com](mailto:gunturppamungkas@gmail.com).