



**PERAMALAN BEBAN LISTRIK JANGKA PANJANG PADA PT. PLN  
(PERSERO) APJ JEMBER DENGAN MENGGUNAKAN METODE  
*RECURRENT NEURAL NETWORK* DENGAN OPTIMASI  
*LEVENBERG MARQUARDT***

**SKRIPSI**

Oleh  
**Muhammad Haidlir Zulkarnain**  
**NIM 141910201044**

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO  
JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS JEMBER  
2018**



**PERAMALAN BEBAN LISTRIK JANGKA PANJANG PADA PT. PLN  
(PERSERO) APJ JEMBER DENGAN MENGGUNAKAN METODE  
*RECURRENT NEURAL NETWORK* DENGAN OPTIMASI  
*LEVENBERG MARQUARDT***

**SKRIPSI**

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat  
untuk menyelesaikan studi di Jurusan Teknik Elektro (S1)  
dan mencapai gelar Sarjana Teknik

Oleh

**Muhammad Haidlir Zulkarnain**  
**NIM 141910201044**

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO  
JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS JEMBER  
2018**

## PERSEMBAHAN

Alhamdulillah, segala puji bagi Allah SWT atas segala karunia dan rahmat-Nya kepada penulis sehingga penulis dapat menyelesaikan skripsi yang berjudul “Peramalan Beban Listrik Jangka Panjang pada PT. PLN (Persero) APJ Jember dengan Menggunakan Metode *Recurrent Neural Network* dengan Optimasi *Levenberg Marquardt*” dan skripsi ini merupakan gerbang awal dalam mencapai kesuksesan yang lebih baik lagi. Maka dari itu, saya ingin mempersembahkan karya ini kepada :

1. Allah SWT, Tuhan yang bergantung kepada-Nya segala sesuatu.
2. Nabi Muhammad SAW, utusan Allah SWT untuk menjadi pedoman manusia.
3. Alm. Ibu Wardatul Hasanah dan Ayah Popong Hadi Prayogo sebagai orang tua yang telah mendidik sampai dewasa.
4. Semua Dosen Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember yang telah memberikan ilmu dan membimbing kami selama dibangku perkuliahan. Terutama Dosen Pembimbing Bapak Dr. Bambang Sri Kaloko, S.T., M.T. dan Dr. Triwahju Hardianto, S.T., M.T. yang telah membimbing dan meluangkan waktunya untuk membantu hingga terselesaikannya skripsi ini.
5. Saudaraku seperjuangan Teknik Elektro angkatan 2014 “KETEK UJ”.
6. Almamater Fakultas Teknik Universitas Jember.
7. Luna Ivanka D. E. yang telah membantu dan menyemangati sampai selesai.
8. Serta semua teman – teman yang telah menemani dalam suka maupun duka.

Kepada semua pihak yang telah membimbing, membantu, maupun sekedar menemani dalam menyelesaikan penelitian ini saya ucapkan banyak terima kasih.

**MOTTO**

“Sebaik – baik manusia adalah yang paling bermanfaat bagi manusia lain”

**(HR. Ahmad, ath – Thabrani, ad – Daruqutni. Di dalam shahihul Jami’ no. 3289)**

“Karunia Allah yang paling lengkap adalah kehidupan yang didasarkan pada ilmu pengetahuan”

**(Ali bin Abi Thalib)**

“Janganlah kamu takut dan janganlah kamu bersedih hati. Sesungguhnya Allah bersama kita”

**(Q.S. At-Taubah : 40)**

“Selalu berdoa dan berharap kepada Allah atas segala masalahmu, InsyaAllah hatimu akan diberi ketenangan”

**(Muhammad Haidlir Zulkarnain)**

**PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

Nama : Muhammad Haidlir Zulkarnain

NIM : 141910201044

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul : “Peramalan Beban Listrik Jangka Panjang pada PT. PLN (Persero) APJ Jember dengan Menggunakan Metode *Recurrent Neural Network* dengan Optimasi *Levenberg Marquardt*” adalah benar-benar karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak mana pun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 19 Juli 2018

Yang menyatakan,

Muhammad Haidlir Z.

NIM 141910201044

**SKRIPSI**

**PERAMALAN BEBAN LISTRIK JANGKA PANJANG PADA PT. PLN  
(PERSERO) APJ JEMBER DENGAN MENGGUNAKAN METODE  
*RECURRENT NEURAL NETWORK* DENGAN OPTIMASI  
*LEVENBERG MARQUARDT***

Oleh  
Muhammad Haidlir Zulkarnain  
NIM 141910201044

Pembimbing :

Dosen Pembimbing Utama : Dr. Bambang Sri Kaloko, S.T., M.T.

Dosen Pembimbing Anggota : Dr. Triwahju Hardianto, S.T., M.T.

PENGESAHAN

Skripsi berjudul “Peramalan Beban Listrik Jangka Panjang pada PT. PLN (Persero) APJ Jember dengan Menggunakan Metode *Recurrent Neural Network* dengan Optimasi *Levenberg Marquardt*” Karya Muhammad Haidlir Zulkarnain telah diuji dan disahkan pada :

Hari : Senin

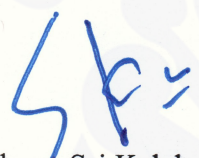
Tanggal : 23 Juli 2018

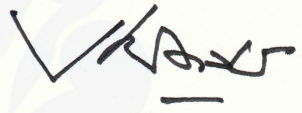
Tempat : Fakultas Teknik Universitas Jember

Tim Penguji,

Ketua,


Anggota I,


  
Dr. Bambang Sri Kaloko, S.T., M.T.  
NIP 197104022003121001

  
Dr. Triwahju Hardianto, S.T., M.T.  
NIP 197008261997021001

Anggota II,

Anggota III,

  
Prof. Dr. Ir. Bambang Sujanarko, M.M.  
NIP 196312011994021002

  
H. Samsul Bachri, M, S.T, M.MT.  
NIP 196403171998021001

Mengesahkan,  
Dekan Fakultas Teknik

Dr. Ir. Entin Hidayah M.U.M  
NIP 196612151995032001

## RINGKASAN

**Peramalan Beban Listrik Jangka Panjang pada PT. PLN (Persero) APJ Jember dengan Menggunakan Metode *Recurrent Neural Network* dengan Optimasi *Levenberg Marquardt***; Muhammad Haidlir Zulkarnain, 141910201044; 2018; 85 halaman; Program Studi Strata 1 (S1) Teknik, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember.

Pesatnya perkembangan teknologi pada era saat ini terus mengalami peningkatan yang tinggi, sehingga akan membuat pemakaian terhadap energi listrik pun juga mengalami peningkatan yang tinggi, seperti pada tempat-tempat industri, perkantoran, dan perumahan. Seiring berjalannya waktu, beban listrik yang digunakan juga akan semakin besar. Untuk mengantisipasi hal tersebut, dibutuhkan metode agar dapat memprediksikan jumlah beban listrik yang akan didistribusikan kepada konsumen agar dapat menghindari kerugian seperti pemadaman secara bergilir. Hal tersebut justru akan membuat konsumen menjadi khawatir dan sangat tidak nyaman dalam menggunakan energi listrik. Maka dari itu dibutuhkan suatu metode peramalan beban listrik untuk memprediksi jumlah beban yang dibutuhkan di dalam beberapa waktu ke depan.

Penelitian untuk meramalkan beban listrik ini menggunakan simulasi dengan *software* Matlab. Data yang digunakan pada penelitian kali ini merupakan data beban puncak listrik setiap bulan selama 8 tahun yang terbagi menjadi 2 data, yaitu data *input* dan data target. Metode yang diterapkan pada penelitian kali ini yaitu metode *recurrent neural network* dengan optimasi *levenberg marquardt*. Data yang dipakai untuk pelatihan penelitian ini yaitu data pada pertengahan tahun 2010 sampai dengan pada tahun 2017, dimana dari data-data tersebut akan dibagi menjadi data *input* dan data target pelatihan *recurrent neural network*. Data tersebut disusun menjadi 42 data *input* dan 12 data target, serta menggunakan 13 pola yang berbeda, dimana pergantian pada setiap pola merupakan pergeseran dari data *input* dan data target yang bertujuan untuk meminimalkan jumlah nilai *error* dari suatu metode peramalan.



Hasil dari peramalan beban listrik yang didapatkan yaitu nilai MAPE (*Mean Absolute Percentage Error*) yang merupakan perhitungan akurasi peramalan terhadap target, pada pengujian peramalan beban dengan menggunakan metode *recurrent neural network* dengan optimasi *levenberg marquardt* didapatkan nilai MAPE sebesar 4,455%. Nilai MAPE tersebut sudah memiliki tingkat akurasi yang tinggi, dimana C. D. Lewis pada tahun 1982 pada bukunya yang berjudul *Industrial and Business Forecasting Methods: A Radical Guide to Exponential Smoothing and Curve Fitting*, mengatakan bahwa nilai MAPE yang kurang dari 10% sudah dapat dikategorikan memiliki tingkat akurasi peramalan yang tinggi. Data peramalan yang didapatkan pada penelitian ini yaitu data peramalan beban yang akan terjadi di wilayah Jember sejak awal Januari tahun 2018 sampai dengan akhir Desember tahun 2023.

Nilai data beban listrik di wilayah Jember mengalami kenaikan yang cukup tinggi setiap tahunnya meskipun data beban listriknya masih bersifat fluktuatif pada bulan-bulan tertentu. Nilai *error* terkecil pada perbandingan antara data beban sekunder dengan data hasil peramalan beban menggunakan metode *recurrent neural network* dengan optimasi *levenberg marquardt* yaitu sebesar 0,097% pada bulan April 2016 dengan selisih data beban sebesar 0,0637 MW dan nilai *error* terbesar yaitu sebesar 21,619% pada bulan November 2016 dengan selisih data beban sebesar 14,3149 MW.

## PRAKATA

Puji syukur kehadiran Allah SWT, atas segala limpahan karunia dan rahmat-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Peramalan Beban Listrik Jangka Panjang pada PT. PLN (Persero) APJ Jember dengan Menggunakan Metode *Recurrent Neural Network* dengan Optimasi *Levenberg Marquardt*”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, disampaikan terima kasih kepada:

1. Allah SWT, yang telah melimpahkan rizki dan rahmat-Nya serta memberi kelancaran dan kemudahan atas segalanya, sehingga terselesaikannya proyek akhir ini.
2. Nabi Muhammad SAW, yang telah menuntun kita kepada peradaban manusia yang lebih baik.
3. Ibu Dr. Ir. Entin Hidayah M.U.M selaku Dekan Fakultas Teknik Universitas Jember.
4. Bapak Dr. Bambang Sri Kaloko, S.T., M.T., selaku Dosen Pembimbing Utama dan Bapak Dr. Triwahju Hardianto, S.T., M.T., selaku Dosen Pembimbing Anggota yang telah membantu dan meluangkan waktunya, serta memberikan bimbingan dalam penyusunan penelitian ini.
5. Bapak Prof. Dr. Ir. Bambang Sujanarko, M.M., selaku dosen penguji utama dan Bapak H. Samsul Bachri M, S.T., M.MT., selaku dosen penguji anggota yang telah membantu memberikan kritik dan saran yang dapat membangun sehingga sangat membantu terhadap penyempurnaan penyusunan skripsi ini.
6. Bapak Dr. Bambang Sri Kaloko, S.T., M.T., selaku dosen pembimbing akademik yang telah membimbing dan menanamkan rasa disiplin dan tanggung jawab dengan apa yang dilakukakan selama penulis menjadi mahasiswa.
7. Para dosen beserta seluruh staf karyawan Fakultas Teknik Universitas Jember, terima kasih atas segala bantuan dan dukungannya selama ini.

8. Alm. Ibu Wardatul Hasanah dan Ayah Popong Hadi Prayogo tercinta, yang telah mendidik, memberi kasih sayang dan mendoakan.
9. Luna Ivanka D. E., yang selalu ada untuk membantu dan memberi semangat untuk penulis setiap waktu dan tanpa mengenal lelah.
10. Saudaraku elektro angkatan 2014 “KETEK UJ” terima kasih kalian adalah inspirasi dan penyemangatku.
11. Sahabat di kala senang dan susah, terimakasih Aldi, Bowok, Dwie, Kevin, dan Neni.
12. Semua pihak yang terlibat dalam membantu penyelesaian penelitian ini.

Semoga skripsi ini dapat bermanfaat dan memberikan pengetahuan dalam mengembangkan ilmu pengetahuan khususnya untuk disiplin ilmu teknik elektro, kritik dan saran diharapkan terus mengalir agar dapat lebih menyempurnakan skripsi ini dan diharapkan dapat dikembangkan kedepannya untuk penelitian-penelitian selanjutnya.

Jember, 17 Juli 2018

Penulis

**DAFTAR ISI**

	Halaman
<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>HALAMAN PERSEMBAHAN .....</b>	<b>ii</b>
<b>HALAMAN MOTTO .....</b>	<b>iii</b>
<b>HALAMAN PERNYATAAN.....</b>	<b>iv</b>
<b>HALAMAN PEMBIMBING .....</b>	<b>v</b>
<b>HALAMAN PENGESAHAN.....</b>	<b>vi</b>
<b>HALAMAN RINGKASAN .....</b>	<b>vii</b>
<b>HALAMAN PRAKATA.....</b>	<b>ix</b>
<b>DAFTAR ISI.....</b>	<b>xi</b>
<b>DAFTAR GAMBAR.....</b>	<b>xiv</b>
<b>DAFTAR TABEL .....</b>	<b>xvi</b>
<b>DAFTAR LAMPIRAN.....</b>	<b>xvii</b>
<b>BAB 1. PENDAHULUAN .....</b>	<b>1</b>
<b>1.1 Latar Belakang.....</b>	<b>1</b>
<b>1.2 Rumusan Masalah .....</b>	<b>3</b>
<b>1.3 Batasan Masalah .....</b>	<b>4</b>
<b>1.4 Tujuan.....</b>	<b>4</b>
<b>1.5 Manfaat Penelitian.....</b>	<b>4</b>
<b>BAB 2. TINJAUAN PUSTAKA.....</b>	<b>5</b>
<b>2.1 Sistem Tenaga Listrik.....</b>	<b>5</b>
2.1.1 Sistem Pembangkit .....	5
2.1.2 Sistem Transmisi.....	6
2.1.3 Sistem Distribusi.....	7
<b>2.2 Sistem Pendistribusian Tenaga Listrik.....</b>	<b>9</b>
2.2.1 Sistem Pendistribusian Langsung .....	9
2.2.2 Sistem Pendistribusian Tidak Langsung.....	10
<b>2.3 Klasifikasi dan Karakteristik Umum Beban.....</b>	<b>10</b>
2.3.1 Klasifikasi Beban.....	10

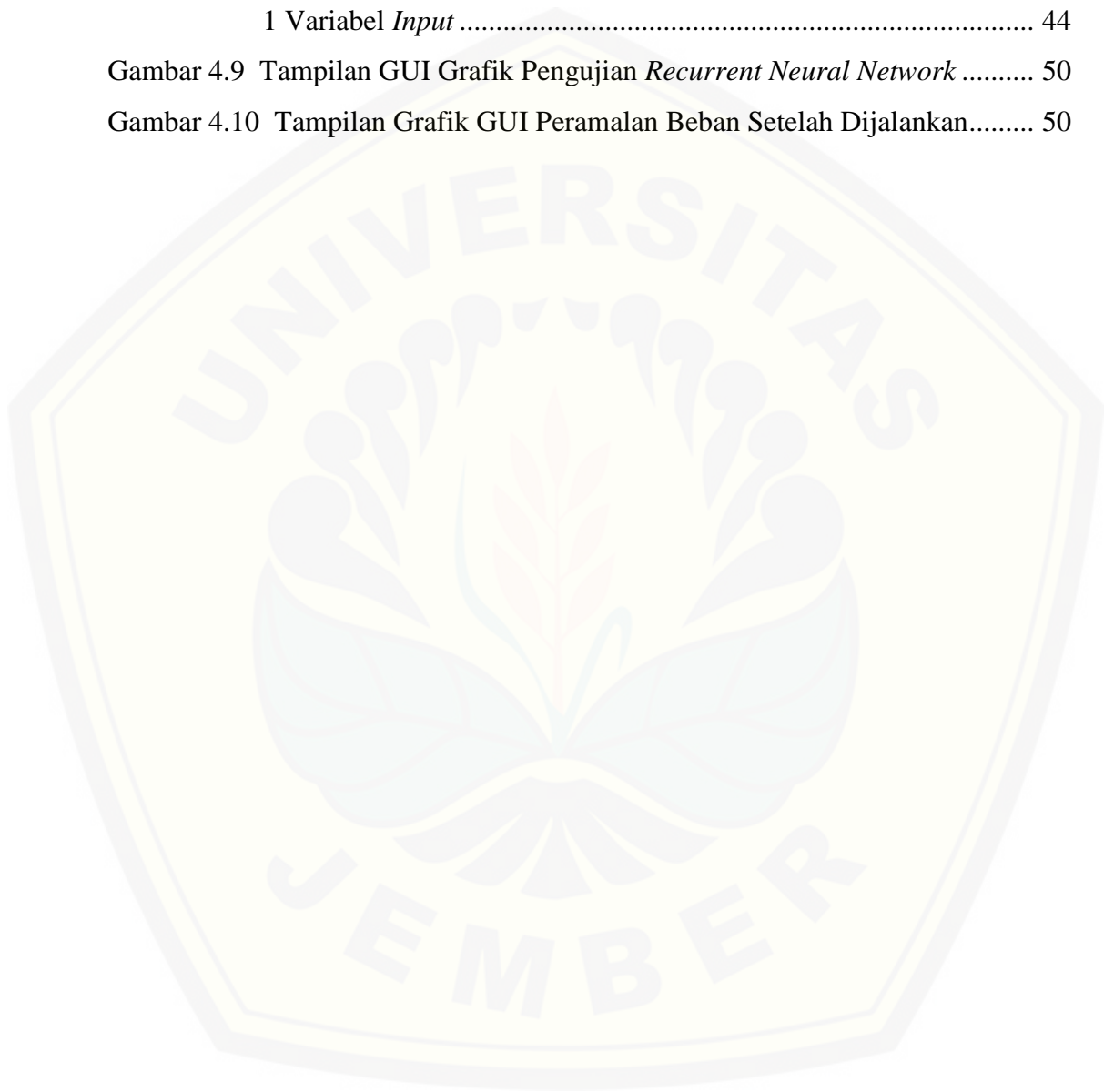
2.3.2 Karakteristik Beban .....	11
2.3.3 Peramalan di Bidang Tenaga Listrik .....	11
<b>2.4 Parameter Peramalan Beban Listrik .....</b>	<b>13</b>
2.4.1 Jumlah Penduduk.....	14
2.4.2 Pertumbuhan Ekonomi .....	14
2.4.3 Pendapatan Perkapita.....	15
2.4.4 Pertumbuhan Industri.....	15
<b>2.5 Jenis Peramalan Beban Listrik.....</b>	<b>13</b>
2.5.1 Jenis dan Akibat Kesalahan Peramalan .....	14
2.5.2 Jenis Kesalahan Peramalan.....	14
2.5.3 Akibat Kesalahan Peramalan .....	14
<b>2.6 Analisa Data Berkala.....</b>	<b>15</b>
<b>2.7 Neural Network (NN) .....</b>	<b>17</b>
2.7.1 Arsitektur <i>Neural Network</i> .....	19
2.7.2 Fungsi Transfer .....	21
2.7.3 Aplikasi <i>Neural Network</i> .....	22
<b>2.8 Recurrent Neural Network .....</b>	<b>22</b>
<b>2.9 Levenberg Marquardt .....</b>	<b>24</b>
<b>BAB 3. METODE PENELITIAN.....</b>	<b>27</b>
<b>3.1 Prosedur Penelitian.....</b>	<b>27</b>
<b>3.2 Diagram Alir Penelitian .....</b>	<b>29</b>
<b>3.3 Diagram Alir Peramalan Beban.....</b>	<b>30</b>
<b>3.4 Peramalan Beban Menggunakan <i>Software Matlab</i> .....</b>	<b>32</b>
<b>3.5 Arsitektur <i>Training dan Testing Recurrent Neural Network</i>.....</b>	<b>32</b>
3.5.1 <i>Preprocessing</i> .....	33
3.5.2 Model Jaringan Syaraf .....	33
3.5.3 <i>Training dan Testing</i> .....	34
<b>BAB 4. HASIL DAN PEMBAHASAN.....</b>	<b>36</b>
<b>4.1 Data Beban Wilayah Jember .....</b>	<b>36</b>
<b>4.2 Data Pelatihan <i>Recurrent Neural Network</i>.....</b>	<b>37</b>

4.3 Hasil dan Analisis Peramalan Beban Menggunakan Metode <i>Recurrent Neural Network</i> dengan Optimasi <i>Levenberg Marquardt</i> .....	38
4.4 Hasil dan Analisa Pengujian Data Peramalan Beban pada <i>Recurrent Neural Network</i> .....	42
4.5 Hasil dan Pengujian Data Peramalan Beban pada <i>Recurrent Neural Network</i> pada <i>Graphical User Interface (GUI)</i> .....	48
<b>BAB 5. PENUTUP</b> .....	<b>51</b>
5.1 Kesimpulan.....	51
5.2 Saran .....	52
<b>DAFTAR PUSTAKA</b> .....	<b>53</b>
<b>LAMPIRAN</b> .....	<b>56</b>

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Sistem Tenaga Listrik.....	5
Gambar 2.2 Blok Diagram Pembangkit Listrik .....	6
Gambar 2.3 Konfigurasi Transmisi Tenaga Listrik .....	7
Gambar 2.4 Sistem Distribusi Tenaga Listrik .....	8
Gambar 2.5 Jaringan Distribusi .....	8
Gambar 2.6 Pola Variasi <i>Trend</i> Permintaan Daya Listrik .....	16
Gambar 2.7 Bentuk Dasar <i>Neuron</i> .....	17
Gambar 2.8 Arsitektur <i>Single Layer Network</i> .....	20
Gambar 2.9 <i>Multi Layer Network</i> .....	20
Gambar 2.10 Struktur <i>Recurrent Network</i> .....	21
Gambar 2.11 Struktur <i>Recurrent Neural Network</i> .....	23
Gambar 3.1 Diagram Alir Penelitian .....	29
Gambar 3.2 Diagram Alir Peramalan Beban .....	31
Gambar 4.1 Grafik Perubahan Data Beban Puncak di Wilayah Jember dari Tahun 2010 – 2017 (Data Sekunder PT. PLN (Persero) APJ Jember).....	36
Gambar 4.2 Grafik <i>Performance</i> Pelatihan <i>Recurrent Neural Network</i> dengan 1 Variabel <i>Input</i> .....	39
Gambar 4.3 Grafik <i>Performance</i> Pelatihan <i>Recurrent Neural Network</i> dengan 5 Variabel <i>Input</i> .....	40
Gambar 4.4 Grafik <i>Regression</i> Pelatihan <i>Recurrent Neural Network</i> dengan 1 Variabel <i>Input</i> .....	40
Gambar 4.5 Grafik <i>Regression</i> Pelatihan <i>Recurrent Neural Network</i> dengan 5 Variabel <i>Input</i> .....	41
Gambar 4.6 Grafik <i>Training State</i> Pelatihan <i>Recurrent Neural Network</i> dengan 1 Variabel <i>Input</i> .....	41

Gambar 4.7 Grafik <i>Training State</i> Pelatihan <i>Recurrent Neural Network</i> dengan 5 Variabel <i>Input</i> .....	42
Gambar 4.8 Grafik Hubungan Data Beban Sekunder, Data Beban RNN dengan 5 Variabel <i>Input</i> , dan Data Beban RNN dengan 1 Variabel <i>Input</i> .....	44
Gambar 4.9 Tampilan GUI Grafik Pengujian <i>Recurrent Neural Network</i> .....	50
Gambar 4.10 Tampilan Grafik GUI Peramalan Beban Setelah Dijalankan.....	50



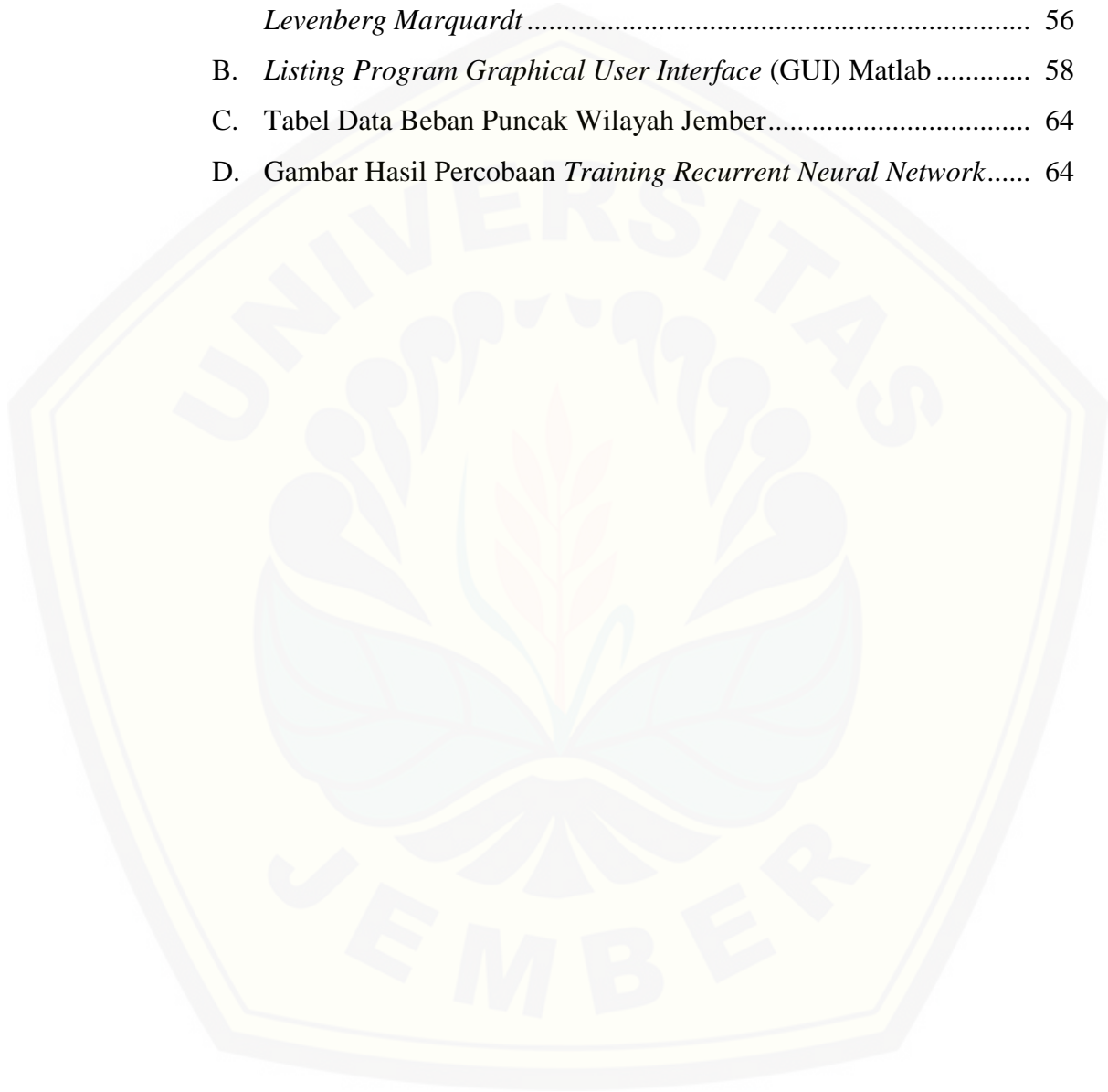


**DAFTAR TABEL**

	Halaman
Tabel 2.1 Karakteristik Jaringan Distribusi Tenaga Listrik .....	9
Tabel 4.1 Hasil <i>Error</i> dari Pelatihan <i>Recurrent Neural Network</i> .....	38
Tabel 4.2 Perbandingan Data Nilai Beban Sekunder dengan Beban Peramalan RNN 5 Variabel <i>Input</i> dan Beban Peramalan RNN 1 Variabel <i>Input</i> .....	43
Tabel 4.3 Peramalan Beban Puncak Wilayah Jember Tahun 2018-2023 dengan Menggunakan Metode RNN .....	46

**DAFTAR LAMPIRAN**

	Halaman
A. <i>Listing Program Recurrent Neural Network</i> dengan Optimasi <i>Levenberg Marquardt</i> .....	56
B. <i>Listing Program Graphical User Interface (GUI) Matlab</i> .....	58
C. Tabel Data Beban Puncak Wilayah Jember.....	64
D. Gambar Hasil Percobaan <i>Training Recurrent Neural Network</i> .....	64



## BAB 1. PENDAHULUAN

### 1.1 Latar Belakang

Teknologi yang kian berkembang pada era dewasa ini terus mengalami peningkatan, kebutuhan akan energi berkembang pesat terutama energi listrik yang sangat dibutuhkan seperti pada industri, perkantoran, dan perumahan. Seiring berjalannya waktu, beban listrik yang digunakan juga akan bertambah besar. Maka dari itu, dibutuhkan suatu prediksi atau peramalan beban untuk mengetahui jumlah daya listrik yang didistribusikan ke konsumen. Hal ini dilakukan untuk menghindari terjadinya pemadaman secara bergilir. Peningkatan kebutuhan energi listrik tersebut harus diimbangi dengan penyediaan energi listrik yang memadai, sehingga diperlukan sistem tenaga listrik yang handal namun tetap ekonomis (Marlia, 2016).

Sidemen (2013) mengatakan bahwa untuk memenuhi kebutuhan tersebut secara kuantitas dan kualitas maka dibutuhkan perencanaan sistem tenaga listrik yang tepat. Agar dapat berjalan dengan baik dan sesuai dengan perencanaan yang ditentukan, maka yang harus diutamakan adalah dengan mengetahui jumlah beban listrik yang dibutuhkan pada jangka waktu tertentu. Prediksi atau peramalan beban yang digunakan untuk penelitian ini meliputi beberapa variabel, seperti beban puncak historis di wilayah Jember, jumlah perkembangan penduduk di wilayah Jember, dan lain-lain. Pendistribusian optimal dalam mendistribusikan energi listrik kepada para konsumen seharusnya sesuai dengan kebutuhan konsumen itu sendiri. Hal tersebut memiliki tujuan yaitu agar energi listrik yang disalurkan kepada konsumen dari pihak PLN tidak berlebihan, yang nantinya akan membuat pemborosan energi listrik yang menimbulkan kerugian pada pihak PLN.

Jumlah energi listrik yang diperlukan konsumen di wilayah Jember dapat diprediksikan atau diramalkan selama beberapa waktu ke depan dengan menggunakan suatu metode peramalan beban listrik dengan cara penyesuaian kapasitas energi listrik supaya sesuai dengan kebutuhan permintaan para konsumen. Salah satu cara yang bisa dilakukan memprediksi permintaan beban listrik yang dibutuhkan oleh konsumen dalam beberapa jangka waktu kedepan (Zulfa dan

Suhartono, 2015). Nasution (2005), mengatakan bahwa peramalan dalam hubungannya dengan horizon waktu, dapat diklasifikasikan kedalam 3 kelompok, yaitu: yang pertama, peramalan beban jangka pendek; kedua, peramalan beban jangka menengah; dan ketiga peramalan beban jangka panjang. Dengan mengkategorikan suatu peramalan yang berhubungan dengan waktu tersebut, dapat diketahui pula data *input* yang akan dijadikan sebagai masukan pada pelatihan *recurrent neural network* dengan optimasi *levenberg marquardt*. Data *input* merupakan data suatu variabel yang telah terjadi beberapa waktu sebelumnya, dan akan dipolakan lagi oleh *recurrent neural network* untuk mendapatkan suatu pola baru yang akan berguna sebagai peramalan beban listrik.

Daya yang akan dibangkitkan harus selalu sama dengan daya yang dikonsumsi oleh para pemakai tenaga listrik yang secara teknis umumnya dikatakan sebagai beban sistem (Djiteng, 2006). Menurut Suhartono (2009) perkiraan besarnya penggunaan daya listrik digunakan untuk mengoptimalkan penggunaan daya listrik di masyarakat, sehingga tidak akan terjadi pemborosan ataupun pemadaman listrik. Jika besarnya energi listrik yang diproduksi atau dibangkitkan lebih kecil ataupun tidak cukup untuk kebutuhan energi listrik konsumen, maka dapat menyebabkan *overload* yang akan berdampak pada terjadinya pemadaman yang merugikan pihak konsumen. Sebaliknya, jika besarnya pembangkitan energi listrik lebih tinggi daripada kebutuhan konsumen, maka dapat mengakibatkan terjadinya pemborosan energi listrik sehingga pihak penyedia tenaga listrik dapat mengalami kerugian. Untuk mencegah hal tersebut terjadi, dapat menggunakan metode peramalan beban listrik agar dapat diketahui tingkat kesesuaian antara pembangkitan dan kebutuhan konsumen. Serta agar tercapai penyesuaian antara pembangkitan dan permintaan daya, maka pihak penyedia listrik harus mengetahui beban atau permintaan daya listrik untuk beberapa waktu ke depan dengan melakukan perkiraan beban listrik (Sarjon, 2013). Dari uraian yang telah dijelaskan, penelitian ini akan dilaksanakan dengan menggunakan metode *recurrent neural network* dengan optimasi *levenberg marquardt*. Dimana penelitian peramalan beban serupa dengan menggunakan metode tersebut pernah dilakukan oleh Mambak Udin. Metode yang digunakan adalah metode *recurrent neural network*

dengan optimasi *levenberg marquardt* tentang meramalkan kapasitas baterai *lead acid* pada mobil listrik yang menggunakan metode *levenberg marquardt neural network*. Penelitian yang lain yaitu dilakukan oleh Lori Kusuma Dewi dengan judul penelitian Penentuan Kapasitas *Lead Acid Battery* dengan *Pulse Voltammetry Cyclic* Berbasis *Levenberg Marquardt Neural Network*. Serta penelitian yang dilakukan oleh Ratna Kusuma Wardhani dengan judul penelitian Pengendalian Kedip Tegangan pada Rumah Sakit Menggunakan *Dynamic Voltage Restorer* dengan Metode *Recurrent Neural Network*. Dengan begitu, diharapkan dalam penelitian ini mampu memberi masukan kepada PT. PLN (Persero) APJ Jember, tentang pengoptimalan pendistribusian yang dibantu dengan menggunakan suatu metode peramalan beban, yaitu *recurrent neural network* dengan optimasi *levenberg marquardt* yang akan digunakan.

## 1.2 Rumusan Masalah

Rumusan masalah yang akan dibahas pada penelitian kali ini ialah seperti di bawah ini:

1. Bagaimana membangun jaringan berbasis *recurrent neural network* dengan optimasi *levenberg marquardt* untuk peramalan beban pada PT. PLN (Persero) APJ Jember?
2. Bagaimana tingkat akurasi hasil peramalan beban menggunakan metode *recurrent neural network* dengan optimasi *levenberg marquardt* dengan data asli pada PT. PLN (Persero) APJ Jember?
3. Bagaimana perbedaan tingkat akurasi pengujian beban antara data peramalan beban dengan menggunakan variabel *input* berupa beban saja dan data peramalan beban dengan menggunakan beberapa variabel *input* berupa data beban, jumlah penduduk, pertumbuhan ekonomi, pendapatan perkapita, dan pertumbuhan industri?
4. Bagaimana hasil peramalan beban selama enam tahun ke depan, yaitu dari tahun 2018 sampai dengan tahun 2023?

### 1.3 Batasan Masalah

Adapun beberapa batasan masalah yang digunakan pada penelitian ini yaitu:

1. Peramalan beban dilakukan sebagai peramalan beban puncak bulanan, dengan data input menggunakan data beban historis selama 8 tahun yaitu dari tahun 2010 sampai dengan tahun 2017, dengan membaginya menjadi dua data, yaitu data pelatihan (*training*) dan data pengujian (*testing*).
2. Data beban sekunder yang digunakan sebagai pengujian (*testing*) dibatasi dari tahun 2016 sampai dengan tahun 2017.
3. Data peramalan beban yang dihasilkan yaitu dari tahun 2018 sampai dengan tahun 2023.
4. Penelitian ini hanya menggunakan data-data numerik dari beban-beban historis, tanpa memperhitungkan faktor-faktor eksternal seperti pengaruh keadaan cuaca, curah hujan, pertumbuhan ekonomi, keadaan politik negara, dan faktor-faktor lainnya yang dapat mempengaruhi hasil peramalan beban (Triwulan, 2013).
5. Simulasi peramalan beban yang digunakan menggunakan *software Matlab*.

### 1.4 Tujuan

Tujuan yang akan dicapai dalam penelitian ini ialah:

1. Untuk menganalisis dan mengamati kebutuhan energi listrik pada wilayah Jember dalam kurun waktu 6 tahun ke depan dengan menggunakan metode *recurrent neural network* dengan optimasi *levenberg marquardt*.
2. Untuk mengetahui tingkat akurasi kemampuan peramalan beban menggunakan metode *recurrent neural network* dengan optimasi *levenberg marquardt*.

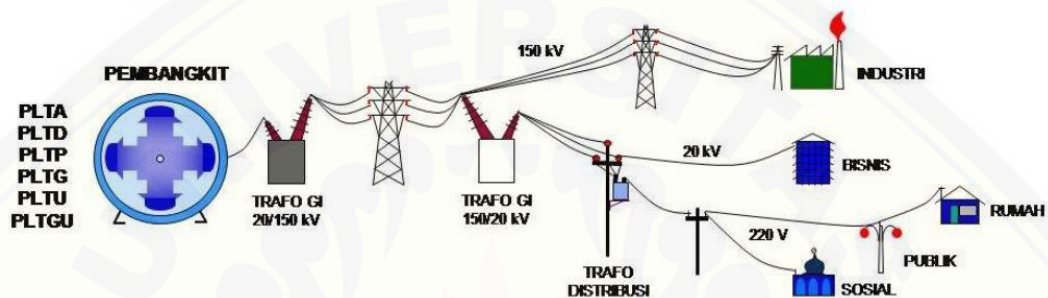
### 1.5 Manfaat Penelitian

Manfaat daripada penelitian ini yaitu agar peramalan beban jangka menengah menggunakan metode *recurrent neural network* dengan optimasi *levenberg marquardt* penerapannya dapat menjadi salah satu metode untuk PT. PLN (Persero) APJ Jember, tentang pengoptimalan pendistribusian yang dibantu dengan menggunakan suatu metode peramalan beban, yaitu *recurrent neural network* dengan optimasi *levenberg marquardt*.

## BAB 2. TINJAUAN PUSTAKA

### 2.1 Sistem Tenaga Listrik

Sistem tenaga listrik ialah suatu sistem kesatuan instalasi listrik yang terbagi atas unit pembangkit, transmisi, dan distribusi dimana tiga hal tersebut terintegrasi satu sama lain untuk menyalurkan listrik kepada konsumen.



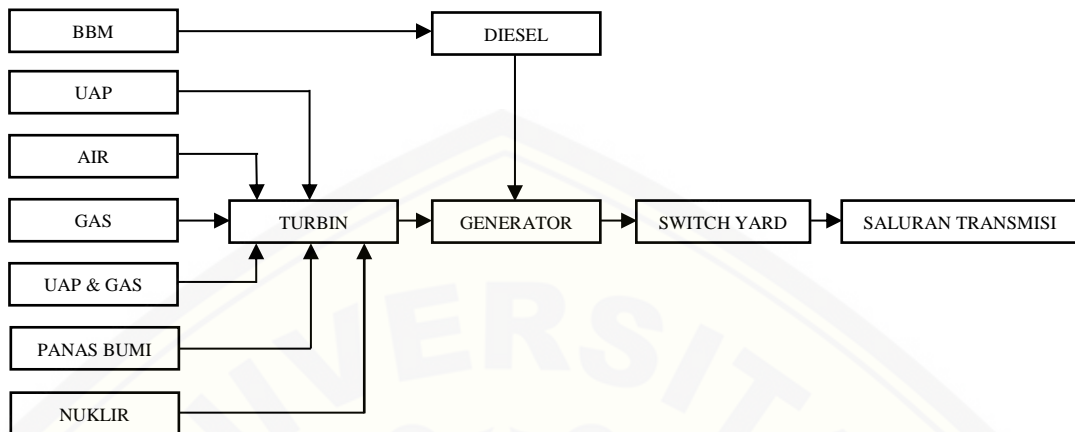
Gambar 2.1 Sistem Tenaga Listrik  
(Satu Energi, diakses pada 7 Juni 2018)

Seperti pada gambar 2.1 tersebut, maka sistem tenaga listrik dapat dijelaskan sebagai berikut.

#### 2.1.1 Sistem Pembangkit

Sistem pembangkit yang merupakan suatu proses mengkonversi sumber energi yang berasal dari alam menjadi energi mekanis, dan kemudian generator mengubah kembali energi mekanis tersebut menjadi energi listrik. Seperti pada gambar 2.1, dapat diamati bahwa pada pusat pembangkit listrik terdapat beberapa jenis pembangkit yang sering digunakan yaitu antara lain PLTU (Pembangkit Listrik Tenaga Uap), PLTG (Pembangkit Listrik Tenaga Gas), PLTGU (Pembangkit Listrik Tenaga Gas dan Uap), PLTP (Pembangkit Listrik Tenaga Panas Bumi), PLTA (Pembangkit Listrik Tenaga Air), dan PLTD (Pembangkit Listrik Tenaga Diesel). Pada gambar 2.2 berikut terdapat gambar blok diagram pembangkit listrik yang mana sumber yang dijadikan untuk membangkitkan listrik

bermacam-macam yaitu seperti BBM (bahan bakar minyak), air, uap, gas, uap dan gas, panas bumi, dan nuklir.

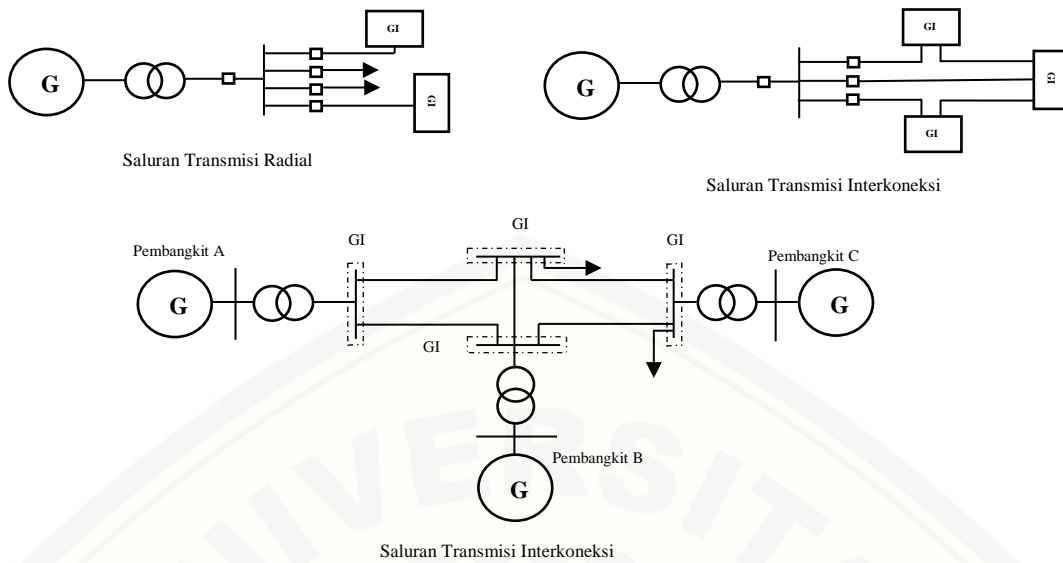


Gambar 2.2 Blok Diagram Pembangkit Listrik  
(Kharis, diakses pada 3 Juni 2017)

### 2.1.2 Sistem Transmisi

Sistem transmisi ialah penyaluran energi listrik dari pembangkit listrik ke gardu induk. Untuk mengurangi rugi dayanya dan agar mengimbangi jarak transmisi yang jauh, maka sebelum energi listrik mulai ditransmisikan, dinaikkan tegangannya terlebih dahulu. Yang perlu dilakukan yaitu menaikkan tegangan yang disuplai dari generator menjadi 70 kV, 150 kV atau 500 kV, dikarenakan tegangan yang keluar dari generator hanya sekitar antara 6,6 kV sampai 24 kV. Setelah tegangannya dinaikkan, selanjutnya listrik ditransmisikan melalui Saluran Udara Tegangan Tinggi (SUTT) atau melalui Saluran Udara Tegangan Ekstra Tinggi (SUTET). Pada gambar 2.3 berikut ini terdapat beberapa macam konfigurasi transmisi tenaga listrik yang terdiri dari saluran transmisi radial, saluran transmisi interkoneksi *ring*, dan saluran transmisi interkoneksi *grid*.





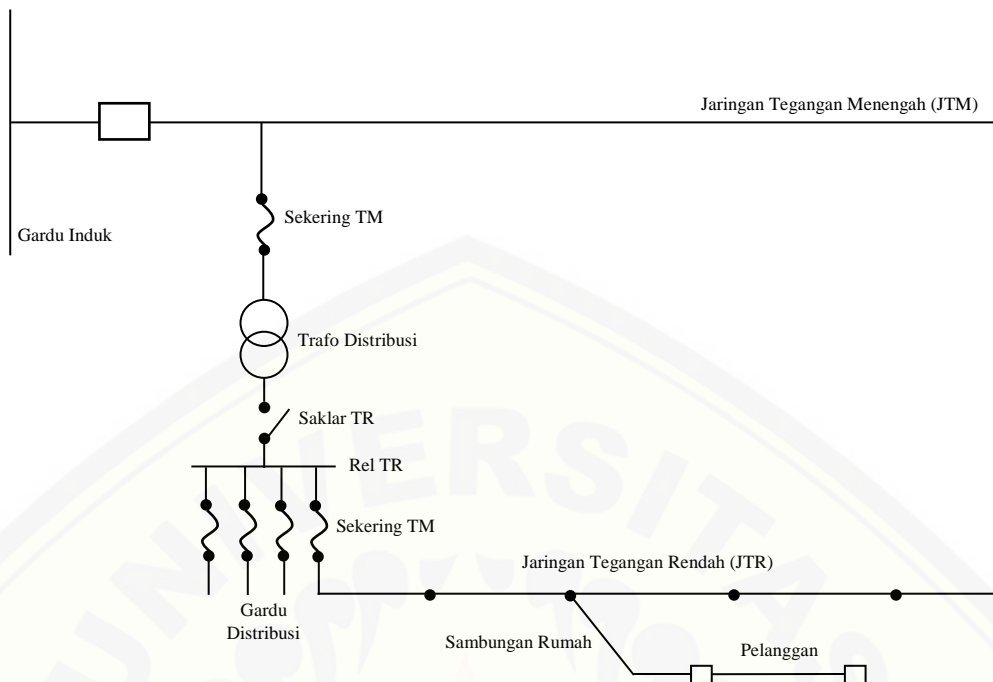
Gambar 2.3 Konfigurasi Transmisi Tenaga Listrik  
(Kharis, diakses pada 7 Juni 2018)

Gardu induk sendiri ialah bagian (sub sistem) dari sistem transmisi, yang berfungsi untuk (Kharis, 2013):

- Mentransformasikan tenaga listrik tegangan tinggi yang satu ketegangan tinggi lainnya (500 kV/150kV, 150 kV/70 kV) atau dari tegangan tinggi ke tegangan menengah (150 kV/20kV, 70 kV/20 kV).
- Pengukuran, pengawas operasi serta pengaturan pengamanan dari sistem tenaga listrik.
- Pengaturan pelayanan beban (daya) ke gardu-gardu induk lain melalui tegangan tinggi dan ke gardu-gardu distribusi setelah melalui proses penurunan tegangan melalui penyulang (*feeder*) tegangan menengah.

### 2.1.3 Sistem Distribusi

Sistem distribusi merupakan suatu proses dalam menyalurkan energi listrik dari gardu induk kepada konsumen. Dimana pada distribusi, tegangan yang tinggi akan diturunkan menjadi tegangan menengah sebesar 20 KV, hal ini disebut juga dengan distribusi primer. Dan tegangan yang disalurkan kepada konsumen diturunkan menjadi tegangan rendah sebesar 380 V untuk kawasan pabrik dan industri, dan 220 V untuk kawasan perumahan penduduk.



Gambar 2.4 Sistem Distribusi Tenaga Listrik  
(Kharis, diakses pada 3 Juni 2017)



Gambar 2.5 Jaringan Distribusi (Daman, 2010)

Pada gambar 2.4 di atas merupakan gambar sistem distribusi energi listrik yang menggambarkan pendistribusian listrik daripada gardu induk hingga sampai kepada konsumen. Contoh gambar jaringan distribusi dapat diamati seperti pada gambar 2.5 di atas.

Dari penjelasan sebelumnya dapat diambil suatu kesimpulan bahwa sistem distribusi tenaga listrik memiliki beberapa karakteristik. Hal ini dapat diamati pada tabel berikut:

Tabel 2.1 Karakteristik Jaringan Distribusi Tenaga Listrik (Daman, 2010)

No	Dari Segi	Distribusi
1	Letak Lokasi Jaringan	Dalam kota
2	Tegangan Sistem	< 30 kV
3	Bentuk Jaringan	Radial, loop, paralel interkoneksi
4	Sistem Penyaluran	Saluran udara dan saluran bawah tanah
5	Konstruksi Jaringan	Lebih rumit dan beragam
6	Analisis Jaringan	Lebih kompleks
7	Komponen Rangkaian Yang Diperhitungkan	Komponen R dan L
8	Penyangga Jaringan	Tiang Jaringan
9	Tinggi Penyangga Jaringan	Kurang dari 20 m
10	Kawat Penghantar	BCC, SAC, AAC, dan AAAC
11	Kawat Tarikan	Dengan kawat tarikan
12	Isolator Jaringan	Jenis pasak (pin), Jenis post (batang), Jenis gantung, Jenis cincin
13	Besarnya Andongan	0 – 1 m
14	Fungsinya	Menyalurkan daya ke konsumen
15	Bahan Penyangga	Baja, besi, kayu
16	Jarak Antar Tiang	40 – 100 m

## 2.2 Sistem Pendistribusian Tenaga Listrik

Sistem pendistribusian tenaga listrik ialah penyaluran energi listrik dari pembangkit ke konsumen. Pada sistem pendistribusian listrik dibagi menjadi dua macam, yaitu sistem pendistribusian langsung dan sistem pendistribusian tidak langsung (Daman, 2010).

### 2.2.1 Sistem Pendistribusian Langsung

Sistem pendistribusian langsung ialah suatu sistem untuk menyalurkan tenaga listrik yang dilakukan tanpa perantara dari pusat pembangkit tenaga listrik, dan tidak melalui jaringan transmisi terlebih dahulu. Sistem pendistribusian jenis ini digunakan jika pusat pembangkit tenaga listrik memiliki jarak yang tidak begitu jauh dari pusat-pusat beban listrik, dan biasanya terdapat pada daerah pelayanan beban ataupun di pinggiran kota.

### 2.2.2 Sistem Pendistribusian Tidak Langsung

Sistem pendistribusian tak langsung ialah suatu sistem untuk menyalurkan energi listrik yang dilakukan jika pusat pembangkit tenaga listrik memiliki jarak yang jauh dari pusat-pusat beban, sehingga untuk penyaluran tenaga listrik jenis ini membutuhkan sebuah jaringan transmisi sebagai jaringan perantara sebelum kemudian dihubungkan dengan jaringan distribusi yang langsung menyalurkan energi listrik kepada konsumen.

Hal-hal yang harus diamati dalam pendistribusian tenaga listrik ialah:

1. Faktor keterandalan sistem.
2. Faktor kualitas sistem.
3. Faktor keselamatan sistem dan publik.
4. Faktor pemeliharaan sistem.
5. Faktor perencanaan sistem.

## 2.3 Klasifikasi dan Karakteristik Umum Beban

### 2.3.1 Klasifikasi Beban

Secara umum, beban listrik dapat diklasifikasikan menjadi 3, yaitu antara lain:

1. Beban Rumah Tangga

Beban rumah tangga hampir secara keseluruhan didominasi oleh peralatan elektronik yang menunjang pada pemakaian sehari-hari seperti televisi, kipas angin, setrika, lemari es, lampu, *handphone*, dan lain-lain.

2. Beban Komersial atau Bisnis

Beban komersial atau bisnis secara umum dikategorikan sebagai peralatan listrik yang dikhususkan digunakan pada tempat-tempat bisnis atau tempat umum seperti lampu untuk penerangan *mall*, hotel, toko, dan lain-lain.

3. Beban Industri

Beban industri dibagi menjadi dua rentang atau skala yang digunakan sesuai dengan faktor permintaan konsumen yang bersangkutan, yaitu skala besar dan skala kecil. Dimana untuk industri skala besar sendiri faktor beban

dapat diambil 70 – 80% dan untuk industri skala kecil faktor beban berkisar 60 - 65%.

### 2.3.2 Karakteristik Beban

Karakteristik yang harus diperhatikan dalam melakukan peramalan beban sendiri yaitu pembagian peralamalan beban yang didasarkan pada suatu periode penggunaan beban tersebut pada suatu sistem, yaitu menggunakan periode harian, bulanan, atau tahunan. Karakteristika beban yang biasanya digunakan sebagai dasar perhitungan dibagi menjadi tiga, antara lain:

#### 1. Karakteristik Beban Harian

Karakteristik beban harian ini adalah dasar perhitungan tentang kebutuhan pembebanan. Ada beberapa faktor yang mempengaruhi karakteristik beban harian ini seperti cuaca pada suatu daerah, maupun kondisi daerah beban tersebut.

#### 2. Karakteristik Beban Bulanan

Karakteristik beban bulanan merupakan data yang berupa beban dengan pengambilan datanya didasarkan pada beban puncak harian selama satu bulan penuh.

#### 3. Karakteristik Beban Tahunan

Karakteristik beban tahunan mempunyai konsep yang menyerupai karakteristik beban bulanan, yaitu pada karakteristik beban tahunan diperoleh dari beban puncak dari beban bulanan selama satu tahun penuh.

### 2.3.3 Peramalan di Bidang Tenaga Listrik

Peramalan (*forecasting*) adalah suatu kegiatan untuk memproyeksikan hal yang telah terjadi di masa lampau ke masa depan. Peramalan di bidang tenaga listrik dilakukan dengan meramalkan beban listrik pada data historis atau data beban listrik yang telah terjadi sebelumnya untuk kemudian diolah kembali menggunakan suatu metode sehingga akan menghasilkan suatu data yang merupakan prediksi atau peramalan beban yang pada beberapa waktu yang akan datang. Peramalan beban ini berperan penting untuk suatu perencanaan pembangkitan tenaga listrik, sehingga kebutuhan listrik pada suatu periode tertentu dapat diprediksikan.

Peramalan berdasarkan dengan sifatnya dibagi menjadi dua jenis, yang pertama peramalan kualitatif, yaitu peramalan ini hanya cenderung menggunakan intuisi atau pertimbangan secara subjektif, tidak menggunakan angka atau data yang sudah ada sebelumnya (historis), contohnya pada suatu peramalan cuaca. Yang kedua peramalan kuantitatif, yaitu peramalan ini didasarkan pada angka atau data yang sudah ada sebelumnya (historis), contohnya pada suatu peramalan beban listrik.

Peramalan berdasarkan jangka waktunya dibagi atas 3 macam, yaitu peramalan jangka pendek, peramalan jangka menengah, dan peramalan jangka panjang.

#### **2.4 Parameter Peramalan Beban Listrik**

Beberapa parameter yang digunakan dalam peramalan beban listrik dengan menggunakan metode *recurrent neural network* dengan optimasi *levenberg marquardt* yang didapat dari publikasi Badan Pusat Statistik Jember dalam angka dengan kurun waktu selama 8 tahun sejak tahun 2010 hingga tahun 2017 ialah seperti di bawah ini:

##### **2.4.1 Jumlah Penduduk**

Kenaikan jumlah penduduk termasuk salah satu parameter dalam penelitian ini, dimana pada setiap bulannya akan diwakilkan menjadi 1 data *input* pada pelatihan *recurrent neural network*. Dari tahun ke tahun, jumlah penduduk di wilayah Jember mengalami peningkatan signifikan. Pada bulan Januari tahun 2010 jumlah penduduk di wilayah Jember sebanyak 1447287 jiwa, dan pada bulan Desember tahun 2017 jumlah penduduk di wilayah Jember menyentuh angka sebanyak 1199380 jiwa.

##### **2.4.2 Pertumbuhan Ekonomi**

Pertumbuhan ekonomi di wilayah Jember juga berkembang pesat pada setiap bulannya, dan sejak awal tahun 2010 hingga akhir tahun 2017 peningkatannya sangat besar. Pada bulan Januari tahun 2010 jumlah pertumbuhan ekonomi di wilayah Jember sebesar Rp. 1.338.734.086.915,-, dan pada bulan Desember tahun 2017 jumlah pertumbuhan ekonomi di wilayah Jember menyentuh angka sebesar Rp. 3.426.455.456.281,-.

### 2.4.3 Pendapatan Perkapita

Pendapatan perkapita di wilayah Jember cukup signifikan dalam perkembangannya dari tahun ke tahun. Pada bulan Januari tahun 2010 jumlah pendapatan perkapita di wilayah Jember sebesar Rp. 10.831,-, dan pada bulan Desember tahun 2017 jumlah pertumbuhan ekonomi di wilayah Jember menyentuh angka sebesar Rp. 24.170,-.

### 2.4.4 Pertumbuhan Industri

Industri di wilayah Jember juga merupakan salah satu variabel yang digunakan dalam *input* data pelatihan, karena pertumbuhan industri juga berbanding lurus dengan semakin tingginya kebutuhan beban listrik yang dibutuhkan. Jenis industri ini meliputi industri makanan, minuman, tembakau, tekstil, barang kulit, alas kaki, barang kayu dan hasil hutan lainnya, kertas dan barang cetakan, pupuk, kimia, barang dari karet, semen dan barang galian nonlogam, logam dasar, besi, baja, alat angkutan, mesin dan peralatannya, dan barang lainnya. Pada bulan Januari tahun 2010 jumlah industri di wilayah Jember sebanyak 37583 unit, dan pada bulan Desember tahun 2017 jumlah pertumbuhan ekonomi di wilayah Jember menyentuh angka sebanyak 44348 unit.

## 2.5 Jenis Peramalan Beban Listrik

Berdasarkan waktunya, peramalan beban listrik terbagi menjadi 2 jenis, yaitu peramalan beban jangka pendek, peramalan beban jangka menengah, dan peramalan beban jangka panjang. Menurut Djiteng Marsudi (2006), pembagian kelompok peramalan/perkiraan beban terdiri atas:

### a. Peramalan Beban Jangka Panjang

Peramalan beban jangka panjang ialah suatu kegiatan yang dilakukan untuk memperkirakan besar keluaran suatu beban listrik dalam jangka waktu diatas satu tahun. Peramalan memiliki peran penting dalam konteks pembangkitan, transmisi dan perencanaan jaringan distribusi di sistem tenaga. Tujuan utama dari perencanaan sistem tenaga adalah untuk menentukan ekspansi ekonomi dari peralatan dan fasilitas untuk memenuhi kebutuhan listrik masa depan

pelanggan dengan tingkat yang dapat diterima keandalan dan kualitas daya. (Minaye dan Matewose, 2013).

Dalam perencanaan peramalan beban jangka panjang membutuhkan data-data yang diambil tiap triwulan ataupun tiap tahun. Signifikansi adalah untuk membantu menentukan instalasi baru unit pembangkit (termasuk kapasitas ukuran, jenis, lokasi dan waktu) dan perencanaan jaringan, peningkatan kapasitas dan konversi.

Beberapa metode yang dapat digunakan untuk meramalkan beban jangka panjang terdapat beberapa macam, antara lain regresi linear, jaringan syaraf tiruan (*neural network*), *fuzzy*, *moving average*, koefisien korelasi, koefisien beban, *exponential smoting* dan lain-lain.

#### b. Peramalan Beban Jangka Menengah

Peramalan beban jangka menengah ialah suatu kegiatan yang dilakukan untuk memperkirakan besar keluaran suatu beban listrik dalam jangka waktu dari 1 bulan hingga 1 tahun.

#### c. Peramalan Beban Jangka Pendek

Peramalan beban jangka pendek adalah suatu kegiatan yang dilakukan untuk memperkirakan besar keluaran suatu beban listrik dalam jangka waktu beberapa jam, hari, hingga 1 minggu.

### 2.5.1 Jenis dan Akibat Kesalahan Peramalan

Pada sistem distribusi energi listrik sendiri terdapat gangguan yang nantinya akan menghambat dalam proses pendistribusian tenaga listrik. Misalnya gangguan yang disebabkan oleh sambaran petir yang akan berpengaruh buruk pada kawat saluran transmisi maupun distribusi. Untuk menangani gangguan-gangguan tersebut maka perusahaan tenaga listrik dapat menyediakan cadangan listrik atau *back up* agar masalah tersebut dapat diatasi.

### 2.5.2 Jenis Kesalahan Peramalan

Kesalahan dalam peramalan dikelompokkan menjadi beberapa jenis, yaitu:



### 1. Kesalahan Tidak Terduga

Kesalahan jenis ini terjadi karena adanya hal-hal yang sebelumnya tidak pernah kita duga. Contohnya faktor kekacauan ekonomi dan faktor kekacauan politik yang terus berlanjut.

### 2. Kesalahan Sistematis

Kesalahan ini terjadi akibat adanya gejala sistematis dari kebutuhan. Secara prinsip kesalahan ini dapat dikurangi tetapi tidak dapat dihilangkan, misalnya pengaruh penggunaan variabel bebas, penentuan parameter persamaan regresi dan pengambilan asumsi dari metode peramalan yang akan digunakan pada perhitungan.

#### 2.5.3 Akibat Kesalahan Peramalan

Pada suatu peramalan, dapat diamati bahwa hasil dari peramalan itu sendiri tidak akan pernah tepat 100%. Penggunaan dari peramalan ini juga masih membutuhkan suatu pertimbangan dari para pemakainya sebelum menggunakan. Pengaruh yang dapat diamati pada kesalahan suatu peramalan peramalan dapat berbeda sesuai dengan metode yang digunakan dalam dan juga periode tertentu yang digunakan dalam peramalan tersebut.

Pentingnya peramalan dalam hal peramalan beban listrik adalah jika peramalan yang dapat dihasilkan itu mempunyai nilai yang besar maka dapat mengakibatkan lonjakan unit pembangkit yang dioperasikan dan juga banyaknya persediaan tenaga listrik yang tidak dibutuhkan. Begitu juga sebaliknya jika peramalan yang dihasilkan itu kecil maka dapat menghasilkan ketidaksanggupan dan ketidakandalan sistem dalam upaya pemenuhan kebutuhan. Besar kecilnya biaya operasi yang mengalami kenaikan dan juga mahal biaya pelayanan energi listrik juga termasuk hal yang berhubungan dengan kesalahan pada peramalan.

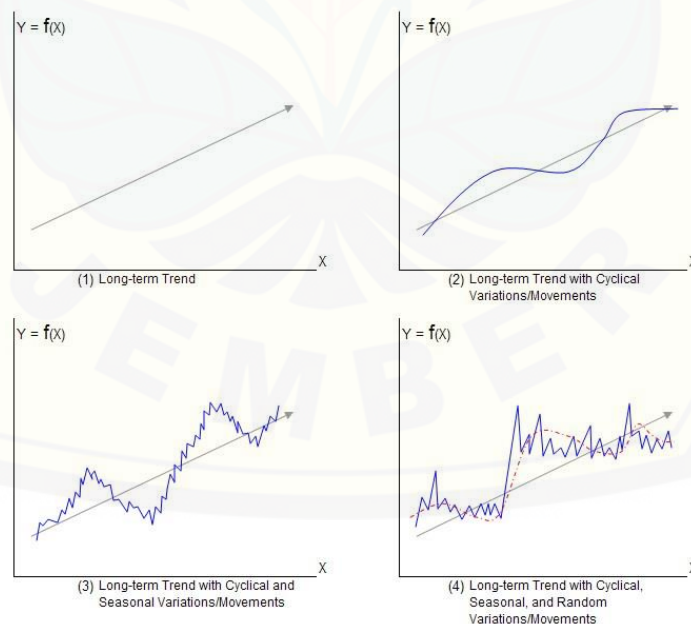
## 2.6 Analisa Data Berkala

Data berkala atau sering kita sebut *time series*, ialah suatu data dari periode tertentu mengenai perkembangan dari suatu kejadian. Analisa data berkala dapat membuat kita mengetahui beberapa perkembangan kejadian dan juga pengaruh serta hubungannya dengan kejadian yang lainnya.

Ketidakpastian aktivitas pemakaian sepanjang waktu adalah hal yang dapat membuat sulit dalam melakukan suatu peramalan. Dari ketidakpastian ini, terdapat empat pola permintaan terhadap energi listrik, yaitu seperti berikut:

1. Kecenderungan (*trend*), ialah pola gerakan aktivitas yang bentuk dari grafiknya cenderung bersifat turun, rata, maupun naik, dan hal tersebut dapat bertahan dalam waktu yang lama.
2. Variasi siklikal (*cyclical variation*) ialah pola gerakan pada grafiknya dimana mempunyai beberapa variasi dalam kurun waktu tertentu pada perubahan nilainya.
3. Variasi musiman (*seasonal variation*) ialah pola gerakan aktivitas yang mengalami naik turun dalam kurun waktu tertentu dan dapat berulang sesuai dengan waktu yang ada.
4. Gerakai variasi yang tidak teratur (*irregular movements*), ialah pola gerakan yang nilainya acak atau tidak tentu tanpa ada penyebabnya.

Pola variasi *trend* permintaan daya listrik yang telah dijelaskan sebelumnya dapat diamati pada gambar 2.6 seperti berikut.



Gambar 2.6 Pola Variasi *Trend* Permintaan Daya Listrik (Imdad, 2014)

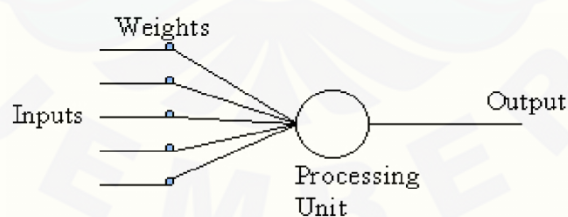
Menurut Putri (2013), kurva kecenderungan *trend* merupakan suatu gerakan kecenderungan naik atau turun dalam jangka panjang yang diperoleh dari rata-rata

perubahan dari waktu ke waktu dan nilainya cukup rata. Pada dasarnya, hasil yang didapatkan dalam suatu data peramalan akan memberikan hasil yang berbeda dengan hasil yang didapatkan dari kenyataan. Hal tersebut terjadi karena tidak semua faktor dan juga variabel yang membuat data berubah tidak dapat diketahui.

## 2.7 Neural Network (NN)

*Neural Network* (NN) ialah suatu jaringan pemroses yang dapat memproses, memodelkan, atau mempolakan suatu kejadian atau data yang cara kerjanya seperti jaringan syaraf pada manusia. *Neural network* dapat membuat suatu struktur baru jaringan untuk melakukan pemecahan suatu masalah dengan menggunakan suatu data atau kejadian masa lalu. Jika diamati sesederhana mungkin, NN merupakan suatu alat untuk memodelkan data yang nonlinier atau tidak beraturan. NN juga dapat digunakan untuk membentuk pemodelan baru antara masukan dan keluarannya, serta untuk menemukan beberapa pola data pada *input*.

Pada dasarnya, penambahan pengetahuan yang dilakukan pada *neural network* dapat dilakukan dengan cara pembelajaran pada sistem jaringan untuk dapat mengenali suatu objek ataupun sesuatu. *Neuron* adalah bagian dasar dari pemrosesan suatu *Neural Network* (Noviana, 2008). Pada gambar berikut adalah gambar bentuk dasar *neuron*.



Gambar 2.7 Bentuk Dasar *Neuron* (Noviana, 2008)

Keterangan untuk gambar 2.7 di atas yaitu sebagai berikut:

1. *Input* ialah suatu masukan yang digunakan saat mengenali suatu pola atau objek ataupun saat melakukan pelatihan.
2. *Weight* ialah beban yang keadaanya selalu berubah setiap kali diberikan *input* sebagai proses pelatihan.

3. *Processing Unit* ialah suatu tempat untuk mengenali suatu aktivitas atau objek yang mengalami perubahan dengan menggunakan pembebanan yang ditentukan.
4. *Output* ialah suatu keluaran dari proses pelatihan atau pembelajaran terhadap suatu objek..

Beberapa keuntungan penggunaan *neural network* dalam suatu sistem adalah:

1. *Neural network* dapat mengenali objek, *non-linier* mampu untuk mengenali suatu objek secara *non-linier*.
2. Pemetaan *input* dapat menghasilkan suatu hasil tanpa tahu proses yang sebenarnya terjadi.
3. Adaptasi terhadap mengenali suatu objek dapat dilakukan dengan *neural network*.
4. Toleransi yang dimiliki *neural network* dapat mentolerir kesalahan data dalam melakukan pelatiha untuk mengenali suatu objek.
5. *Neural network* dapat diterapkan dalam sebuah perangkat keras atau *hardware*.

JST (*Neural Network*) ialah suatu kumpulan integrasi komputasi dengan arsitektur dan juga konsep pengoperasian dan sistemnya seperti pada cara kerja sel syaraf pada otak manusia, yang dapat mengenali suatu pola, melakukan komputasi, dan juga mengendalikan organ-organ pada tubuh yang mempunyai kecepatan berpikir yang lebih cepat daripada sebuah komputer digital. Hal tersebut menjadikan JST sangat cocok untuk menyelesaikan masalah dengan tipe sama seperti otak manusia (Dwisatya, 2015). Sebuah JST ditentukan dengan tiga hal berikut:

1. Pola-pola yang menghubungkan antara *neuron* satu dengan yang lainnya atau yang biasa disebut arsitektur jaringan.
2. Metode untuk menentukan bobot penghubung *neuron* yang disebut juga metode algoritma / *learning* / *training*.
3. Fungsi untuk mengaktivasi jaringan yang digunakan.

Menurut Irma Handayani (2012), Data pelatihan ini dimaksudkan untuk mendapatkan hasil yang baik, namun aspek pembagian data harus ditekankan agar jaringan mendapat data pelatihan yang secukupnya. Jumlah *input* yang terlalu

sedikit jika digunakan untuk pelatihan (*training*) suatu jaringan, maka jaringan tidak dapat stabil dan juga memiliki sebaran data yang kurang cukup baik. Dan sebaliknya, jika jumlah *input* yang terlalu banyak digunakan untuk pelatihan (*training*) suatu jaringan, maka jaringan akan membuat pemusatan atau konvergensi data mengalami perlambatan. Pelatihan dengan frekuensi yang berlebihan juga membuat jaringan cenderung lebih bersifat menghafal data *input* yang membuat proses generalisasi data menurun.

Munculnya *error* pada suatu peramalan beban merupakan hal yang selalu terjadi, dimana menurut Nasution (2006), ukuran akurasi yaitu MAPE (*Mean Absolute Percentage Error*) atau rata-rata persentase kesalahan absolut, yang menyatakan persentase kesalahan hasil peramalan terhadap permintaan aktual selama periode waktu tertentu yang akan memberikan informasi persentase kesalahan terlalu tinggi atau terlalu rendah.

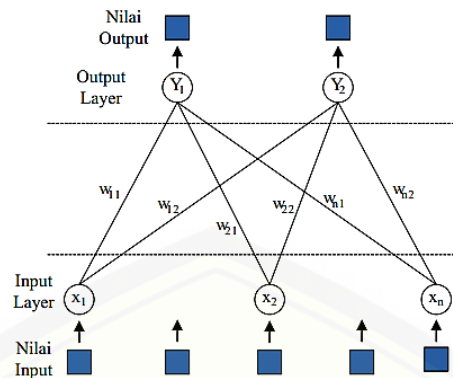
#### 2.7.1 Arsitektur *Neural Network*

Berikut ini merupakan arsitektur suatu jaringan yang sering dipakai dalam menentukan konfigurasi *neural network*.

##### 1. Jaringan Layar Tunggal (*Single Layer Network*)

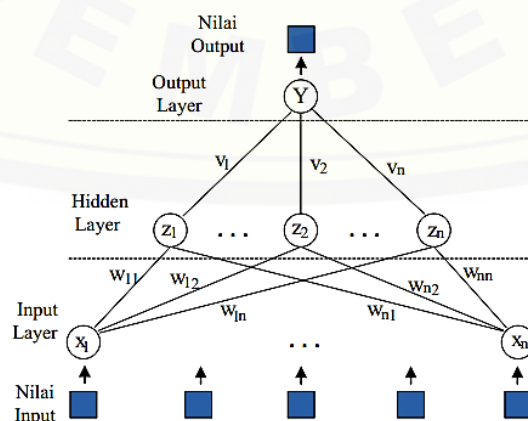
Suatu jaringan yang tersusun atas 1 layar *input* dan juga 1 layar *output* disebut juga dengan jaringan layar tunggal. *Neuron* pada suatu layar *input* mempunyai hubungan dengan *neuron* yang berada pada layar *output*. Pada jaringan ini, penerimaan *input* akan langsung diolah oleh jaringan tanpa melalui perantara layar tersembunyi.

JST layar tunggal mempunyai kekurangan, yaitu keterbatasan jaringan dalam mengenal suatu pola. Kekurangan tersebut dapat ditangani dengan cara menambah satu atau beberapa *hidden layer* yang berada diantara *input layer* dan *output layer* (Pratama, 2016). Algoritma jaringan syaraf tiruan yang menggunakan metode jenis ini yaitu *perceptron* dan *adaline*. Pada Gambar 2.8 dapat diamati arsitektur *single layer network* yang memiliki  $n$  buah *input layer* ( $X_1, X_2, X_3, \dots, X_n$ ) serta 2 buah *output layer* ( $Y_1, Y_2$ ) dengan bobot masing-masing yang menghubungkan antara *input layer* dengan *output layer* ( $W_{11}, W_{12}, W_{21}, W_{22}, W_{n1}, W_{n2}$ ).

Gambar 2.8 Arsitektur *Single Layer Network* (Siregar, 2013)

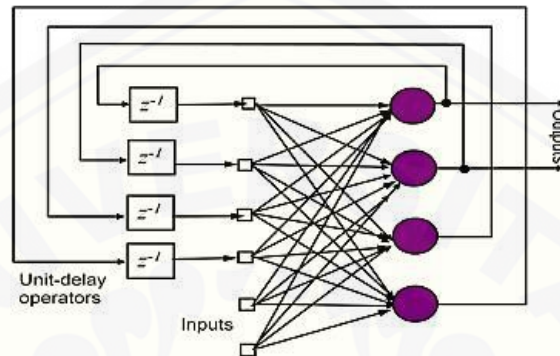
## 2. Jaringan Layer Jamak (*Multi Layer Network*)

Jaringan jenis ini yaitu lapisan layer jamak mempunyai ciri-ciri tertentu ialah memiliki tiga jenis *layer* yaitu *input layer*, *output layer*, dan juga layer tersembunyi atau disebut juga *hidden layer*. Jaringan yang memiliki beberapa lapisan ini dapat membantu menyelesaikan suatu permasalahan yang lebih rumit daripada menggunakan jaringan yang mempunyai layer tunggal. Tetapi, proses untuk melatih jaringan ini membutuhkan waktu yang lebih lama. Contoh pada algoritma *neural network* dengan menggunakan suatu metode jenis ini ialah *recurrent*. Pada gambar 2.9 dapat diamati arsitektur jaringan layer jamak yang mempunyai  $n$  buah layer *input* ( $X_1, X_2, \dots, X_n$ ),  $n$  buah layer tersembunyi ( $Z_1, Z_2, \dots, Z_n$ ) dan sebuah layer *output* ( $Y$ ) dengan bobot yang menghubungkan layer *input* dengan layer tersembunyi ( $W_{11}, W_{12}, W_{1n}, W_{n1}, W_{n2}, W_{nn}$ ) dan bobot yang menghubungkan layer tersembunyi dengan layer *output* ( $V_1, V_2, \dots, V_n$ ).

Gambar 2.9 *Multi Layer Network* (Siregar, 2013)

### 3. Recurrent Network

*Recurrent network* memiliki minimal jumlah paling sedikitnya terdapat satu *feedback loop* didalam *recurrent*. *Feedback loop* mempengaruhi kemampuan belajar dan kinerja jaringan (Udin, 2016).



Gambar 2.10 Struktur *Recurrent Network* (Siregar, 2013)

#### 2.7.2 Fungsi Transfer

*Neural network* memiliki karakter dimana *neural network* bergantung kepada bobot, serta fungsi masukan dan keluaran (fungsi transfer) yang memiliki ciri-ciri berbeda pada setiap unitnya. Fungsi transfer terbagi menjadi tiga jenis seperti berikut.

1. *Linear Units*, dimana aktifitas pada *output* ialah berbanding lurus dengan jumlah bobot data keluaran.
2. *Threshold Units*, keluaran diatur menjadi satu dari beberapa tingkatan yang bergantung dari jumlah masukan ialah lebih rendah atau lebih tinggi dari nilai ambangnya.
3. *Sigmoid Units*, keluaran akan berubah secara berkelanjutan akan tetapi tidak memiliki bentuk yang *linear*. Unit jenis ini memiliki kesamaan pemodelan yang lebih tinggi daripada sel syaraf yang sebenarnya daripada dengan *linear* dan juga *threshold unit*, tetapi ketiga fungsi transfer tersebut masih harus diperkirakan kembali dengan menggunakan perkiraan yang kasar.

Agar *neural network* dapat menegerjakan beberapa pekerjaan yang khusus, pemilihan beberapa unitnya harus menggunakan unit yang tepat agar dapat

dihubungkan untuk mengatur bobotnya. Hubungan tersebut akan menentukan apakah mungkin suatu *unit* mempengaruhi *unit* yang lain (Nasution, 2009).

### 2.7.3 Aplikasi *Neural Network*

Saat ini *neural network* banyak digunakan dalam berbagai sistem. Beberapa aplikasi *neural network* ialah seperti berikut.

#### 1. Pengenalan Pola (*Pattern Recognition*)

*Neural network* dapat digunakan untuk dapat mengenali suatu pola (seperti angka, huruf, tanda tangan, atau suara) yang telah mengalami sedikit perubahan. Hal ini seperti pada otak manusia yang dapat mengenali seseorang yang telah lama tidak dijumpai (seperti bentuk tubuh atau wajah yang telah mengalami sedikit perubahan).

#### 2. *Signal Processing*

*Neural network* (pada model jenis *adaline*) dapat digunakan untuk menekan gangguan suara pada saluran telpon.

#### 3. Peramalan

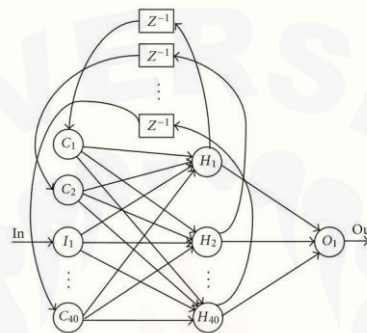
*Neural network* dapat digunakan juga untuk memprediksikan sesuatu yang akan terjadi pada beberapa kurun waktu ke depan dengan berdasarkan suatu pola kejadian yang telah terjadi pada waktu lampau. Peramalan tersebut dapat dilakukan mengingat tingkat kemampuan dari *neural network* dapat menggeneralisasi kejadian dari yang telah terjadi sebelumnya. *Neural network* juga dapat menyelesaikan masalah dalam bidang kontrol dan kedokteran (Siregar, 2013).

### 2.8 *Recurrent Neural Network*

*Recurrent Neural Network* ialah jenis *neural network* yang memiliki kemampuan *feedback* kembali ke *neuron* itu sendiri ataupun ke *neuron* yang lainnya, sehingga aliran informasi dari masukan mempunyai arah jamak (Ariboowo dalam Ammar, 2014). RNN ialah suatu jaringan yang memiliki paling sedikit satu umpan balik berulang. *Feedback loop* dapat mempengaruhi kemampuan belajar dan kinerja suatu jaringan. RNN mempunyai kemampuan penggambaran yang sangat bagus dan dapat mengatasi kelemahan *feedforward* (Soelaiman & Rifa'i dalam Ammar, 2014). *Output* dari RNN tidak bergantung hanya pada *input* pada waktu



itu saja, namun juga bergantung pada kondisi *input neural network* untuk waktu masa lampau. Kondisi seperti ini dimaksudkan agar jaringan dapat menampung beberapa kejadian pada masa lampau yang diikuti pada proses perhitungan. Hal ini penting untuk masalah yang cukup rumit, dan tanggapan keluaran *neural network* berkaitan dengan variasi waktu (*time-varying*), sehingga *neural network* memiliki kepekaan terhadap waktu dengan memori kondisi lampau (Ammar, 2014). Struktur dari *recurrent neural network* dapat diamati pada gambar 2.11 dibawah ini.



Gambar 2.11 Struktur dari *Recurrent Neural Network* (Jaroslaw S., 2011)

*Output* dari *recurrent neural network* dapat dilakukan perhitungan dengan menggunakan persamaan seperti berikut ini:

$$y(k) = x(k + 1) - Ax(k) - V_1^0 \sigma[W_1^0 x(k)] - V_2^0 \phi[W_2^0 x(k)]u(k) \dots \dots \dots (6.1)$$

Keterangan:

$x(k) \in n$  ialah *state vector* pada internal

$A \in \mathbb{R}^{n \times n}$  ialah matrik yang mempunyai nilai tetap

$x(k) \in \mathbb{R}^n$  ialah *state vector*

$u(k) \in \mathbb{R}^m$  ialah *input* dari vektor, dengan nilai  $u(k)$  dan  $x(k)$  diketahui

$W_1^0 \in \mathbb{R}^{n \times n}$ ,  $W_2^0 \in \mathbb{R}^{n \times n}$ , ialah suatu bobot yang terdapat pada lapisan tersembunyi

$V_1^0 \in \mathbb{R}^{n \times 1}$ ,  $V_2^0 \in \mathbb{R}^{n \times 1}$ , ialah suatu bobot yang terdapat pada lapisan keluaran

$x(k+1)$  ialah nilai pada *state vector* yang merupakan hasil perkiraan pada saat iterasi ke  $k+1$

Fungsi aktivasi pada jaringan yang akan digunakan ialah seperti berikut:

$$T m [ \dots ] 1 \sigma = \sigma \sigma , \sigma \in \mathbb{R}^m \text{ dan } \varphi \in \mathbb{R}^{m \times m} \dots\dots\dots(6.2)$$

Nilai  $\sigma$  dan  $\varphi$  dihitung dengan menggunakan fungsi sigmoid (Soelaiman dan Rifa'i dalam Ammar, 2014).

### 2.9 Levenberg Marquardt

*Levenberg marquardt* ialah algoritma yang dikembangkan dari algoritma *error* pada metode *backpropagation*. Pada algoritma jenis ini digunakan untuk menangani beberapa kelemahan algoritma *error* pada metode *backpropagation* dengan menggunakan optimisasi numerik standar ialah dengan penggunaan pendekatan *Jacobian matrix*. Tujuan dari *levenberg marquardt* adalah meminimalkan total *error*. (Ili Susanti, 2014).

Algoritma pada *levenberg marquardt* ialah salah satu cara untuk menangani kelemahan dari kinerja algoritma *backpropagation*, dikarenakan algoritma *levenberg marquardt* dibuat sedemikian rupa agar dapat meminimalisasi jumlah kuadrat *error*. Algoritma jenis ini dapat memperbarui *weight* dengan memperkecil nilai *error* dari selisih *new weight* dan *old weight*. Tetapi, pada saat yang bersamaan, algoritma ini akan mempertahankan *step size* agar tidak menjadi besar. (Kurniawan, 2012).

Persamaan fungsi agar dapat menghitung *new weight* pada algoritma *levenberg marquardt* seperti di bawah ini:

$$w_{\text{new}} = w_{\text{old}} - (Z^T Z)^{-1} Z^T \epsilon (w_{\text{old}}) \dots\dots\dots(6.3)$$

Keterangan:  $w_{\text{new}}$  ialah bobot yang baru  
 $w_{\text{old}}$  ialah bobot yang lama  
 $\epsilon$  ialah vektor dari nilai *error*  
 $Z$  ialah matriks dari hasil selisih *error* terhadap bobot.

Algoritma *levenberg marquardt* mempunyai jumlah nilai iterasi yang lebih bagus jika dibandingkan dengan algoritma pada optimasi *backpropagation* lainnya. Tetapi, algoritma ini membutuhkan memori yang lebih banyak dibandingkan algoritme sejenisnya (Kurniawan, 2012).

Beberapa struktur algoritma *levenberg marquardt* (Oktaorora, 2011):

1. Menginisialisasi bias dan juga bobot dengan menggunakan bilangan *random*, *epoch* yang maksimum, dan juga nilai minimal pada *goal* (*performance* dihitung menggunakan MSE).
2. Menentukan beberapa parameter, seperti di bawah ini:
  - a. Parameter pada *levenberg marquardt* dengan nilai yang harus lebih besar dari 0.
  - b. Parameter dari faktor nilai Beta ( $\beta$ ) pada algoritma ini digunakan menjadi sebuah parameter yang akan dibagi atau dikalikan dengan menggunakan parameter *levenberg marquardt*.
3. Menghitung maju (*feedforward*) pada layar tersembunyi dan layar keluaran seperti pada langkah-langkah algoritma *error* pada metode *backpropagation*.
4. Menghitung nilai daripada MSE.
5. Menghitung nilai *error* dan juga nilai total *error* pada jaringan dengan rumus menghitung *error*:

$$e_r = t_r - y_r \dots\dots\dots(6.4)$$

Keterangan:

r ialah masukan ke-r

Rumus berikut digunakan untuk menghitung nilai dari total *error*:

$$e = [e_1 \ e_2 \ e_3 \ \dots \ e_n]^T \dots\dots\dots(6.5)$$

Keterangan:

e ialah nilai vektor pada kesalahan yang mempunyai ukuran  $N \times 1$  dan terdiri atas  $e_n$   $r = 1, 2, 3, 4, \dots, n$

6. Menghitung matriks Jacobian  $J(x)$   
 Dengan  $x$  ialah matriks yang mempunyai nilai bias dan bobot yang terdapat pada jaringan.

$$X = [ v_{11}, v_{12}, \dots, v_{ij}; v_{01}, v_{02}, \dots, v_{0j}; w_{11}, w_{12}, \dots, w_{jk}; w_{01}, w_{02}, \dots, w_{0k} ] \dots(6.6)$$

Matriks Jacobian mempunyai rumus turunan pertama dari nilai *error* suatu jaringan terhadap nilai dari bias dan bobotnya. Rumus yang digunakan untuk menghitung nilai matriks jacobian ialah seperti berikut:

$$J = [ \frac{\partial e_r}{\partial w} ] \dots\dots\dots(6.7)$$

7. Setelah nilai dari  $J(x)$  didapat, selanjutnya dapat melakukan perhitungan nilai bias dan perubahan koreksi bobot dengan menggunakan rumus di bawah ini:

$$\Delta x = [J(x)^T J(x) + \mu I]^{-1} * Gradient \dots\dots\dots(6.8)$$

$$Gradient: J(x)^T e$$

8. Setelah nilai  $\Delta x$  didapat, selanjutnya ialah melakukan koreksi nilai bobot dengan menggunakan rumus seperti yang ada pada algoritma nilai *error* pada metode *backpropagation*.
9. Menghitung maju (*feedforward*) menggunakan nilai bias dan bobot yang baru.
10. Menghitung nilai MSE jaringan menggunakan bias dan juga bobot yang baru. Kemudian tes pada kondisi saat berhenti. Jika nilai iterasi terus berjalan, maka akan menghasilkan dua kondisi yang memungkinkan seperti berikut:
- Jika nilai MSE naik maka  $\mu \times \beta$
  - Jika MSE turun  $\frac{\mu}{\beta}$
11. Terakhir dengan melakukan pengulangan kembali dari langkah ke 5 sampai dengan langkah ke 8.

## BAB 3. METODE PENELITIAN

### 3.1 Prosedur Penelitian

Pada metodologi penelitian ini, langkah-langkah yang dilakukan dalam pelaksanaan penelitian peramalan beban listrik jangka panjang pada PT. PLN (Persero) APJ Jember dengan menggunakan metode *recurrent neural network* dengan optimasi *levenberg marquardt* yaitu sebagai berikut:

#### 1. Studi Literatur

Langkah pertama dalam melaksanakan penelitian kali ini yaitu melakukan penelusuran studi literatur dengan mencari beberapa sumber referensi seperti jurnal, internet, *paper*, buku, atau dari skripsi yang sudah ada sebelumnya. Berbagai sumber referensi tadi diharapkan dapat membantu untuk perencanaan dan pelaksanaan penelitian ini agar nanti didapatkan hasil seperti yang diharapkan.

#### 2. Pengambilan Data

Data *input* atau masukan yang digunakan pada penelitian kali ini ialah nilai data beban sekunder distribusi energi listrik di wilayah Jember, yaitu data tersebut diperoleh langsung dari PT. PLN (Persero) APJ Jember. Data yang akan digunakan untuk peramalan beban ialah seperti di bawah ini.

- a. Data beban listrik dari tahun 2010 hingga tahun 2017.
- b. Data beban listrik tersebut merupakan data beban historis yang akan digunakan pada penelitian ini.
- c. Data akan terbagi menjadi 2, antara lain data pelatihan (*training*) yang terdiri atas *input* dan target dan data pengujian (*testing*).

#### 3. Pembuatan Sistem Peramalan Metode *Recurrent Neural Network* dengan Optimasi *Levenberg Marquardt*

Pada tahap ketiga ini, yaitu membuat sistem peramalan beban yang akan menggunakan metode *recurrent neural network* dengan optimasi *levenberg marquardt*. Aplikasi yang akan digunakan untuk menjalankan simulasi pada penelitian ini ialah aplikasi *software* Matlab.

#### 4. *Training Neural Network*

Setelah perencanaan konfigurasi peramalan menggunakan *neural network* selesai, yaitu selanjutnya dilakukan *training* atau pelatihan terhadap data beban historis menggunakan metode *recurrent neural network* dengan optimasi *levenberg marquardt* dan akan muncul grafik serta data sebagai hasil peramalan tersebut. Tujuan dari pelatihan (*training*) tersebut yaitu untuk melakukan proses pembelajaran (*learning*) dengan *recurrent neural network* dengan optimasi *levenberg marquardt* untuk menentukan nilai bobot baru. Data *input* dan data target dilatih untuk mendapatkan *output* suatu nilai bobot baru. Parameter yang harus diperhatikan dalam inisialisasi pada jaringan yaitu:

- a. Maksimum *error*
- b. Toleransi *error*
- c. Menentukan fungsi aktivasi
- d. Menentukan iterasi (*epoch*)
- e. Menentukan *hidden layer* dan *neuron*
- f. Menentukan fungsi pelatihan jaringan
- g. Menentukan *learning rate*

Semua tahapan pelatihan (*training*) tersebut akan dilakukan berulang-ulang untuk mendapatkan bobot terbaik dengan *error* terkecil. Bobot terbaik tersebut akan digunakan untuk melakukan pengujian (*testing*) data.

#### 5. *Testing*

Pada tahap kelima yaitu proses pengujian (*testing*) data adalah proses melakukan tes pembelajaran dengan menggunakan bobot terlatih untuk melakukan peramalan. Dalam proses pengujian, data yang diuji adalah data beban.

#### 6. Analisis Data dan Pembuatan Laporan

Pada tahap keenam ini yaitu melakukan analisis terhadap data yang diperoleh dari data yang sebelumnya. Hasil data yang diperoleh yaitu berupa data peramalan beban listrik selama 6 tahun ke depan.

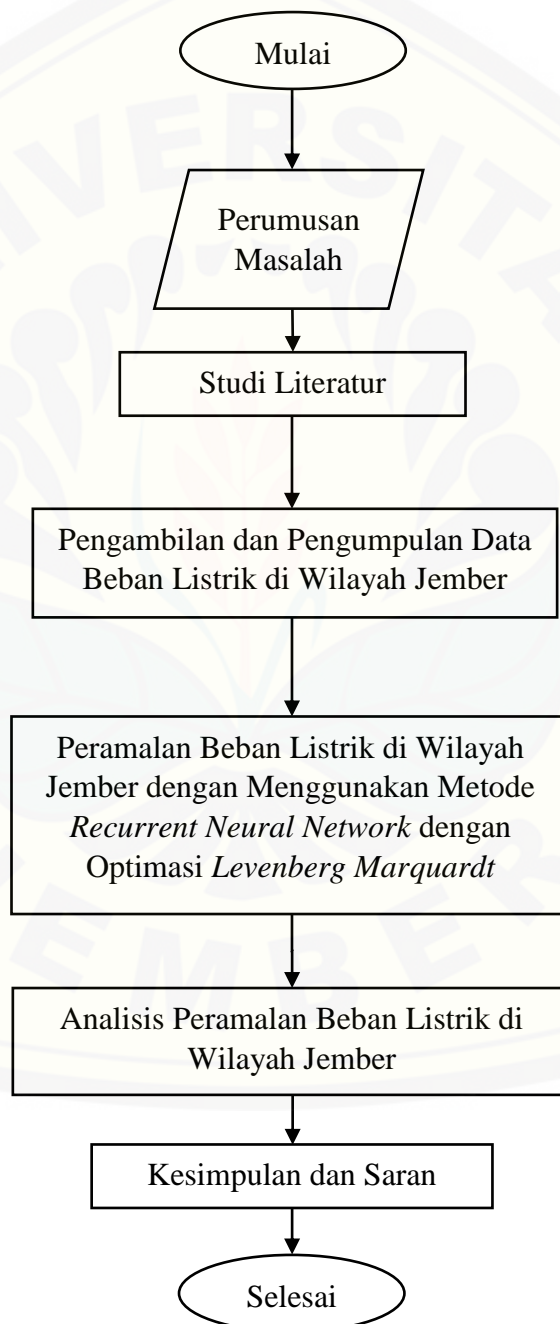
#### 7. Kesimpulan dan Saran

Pada tahap akhir ini yaitu membuat kesimpulan data dari pengujian (*testing*) dan analisis data yang telah diperoleh sebelumnya. Dan juga pemberian saran yang

dimaksudkan untuk pertimbangan atas pengembangan selanjutnya terhadap penelitian yang dilaksanakan sekarang.

### 3.2 Diagram Alir Penelitian

Diagram alir keseluruhan pada penelitian kali ini yaitu ditampilkan pada gambar dibawah ini.

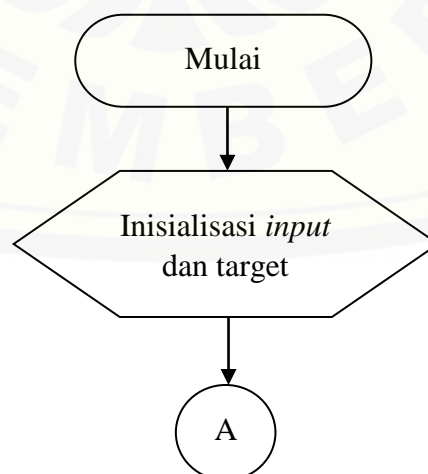


Gambar 3.1 Diagram Alir Penelitian

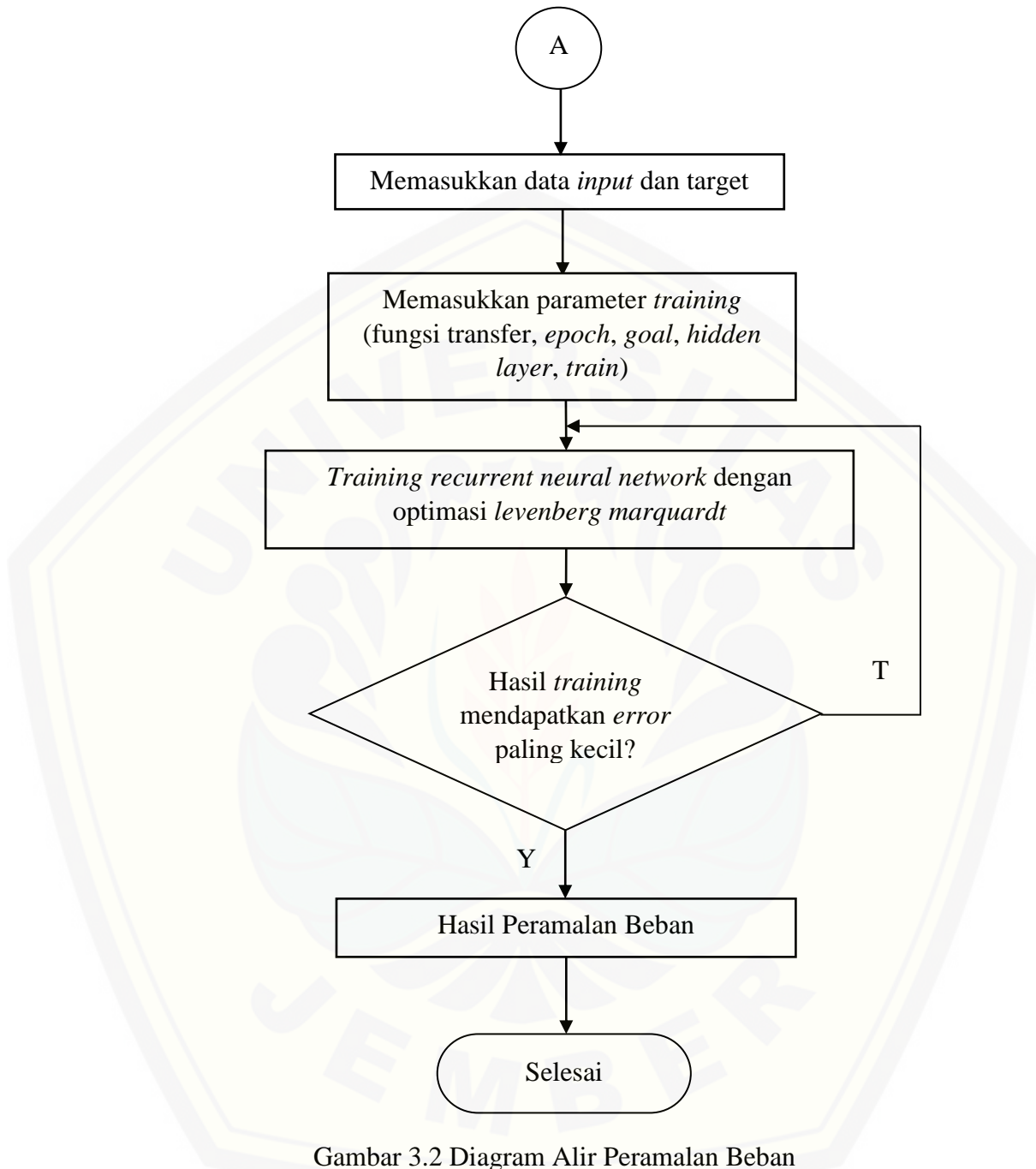
Dari diagram alir penelitian pada gambar 3.1 di atas dapat diamati bahwa proses diagram alir penelitian dimulai yaitu dari perumusan masalah yaitu bagaimana perancangan dan tingkat akurasi metode *recurrent neural network* dengan optimasi *levenberg marquardt* dalam hal peramalan beban listrik di wilayah Jember, dengan menggunakan data beban historis yang didapatkan dari PT. PLN (Persero) APJ Jember. Setelah itu mencari literatur terkait dengan permasalahan tersebut dan mempelajarinya. Kemudian melakukan pengambilan dan pengumpulan data beban historis di wilayah Jember pada PT. PLN (Persero) APJ Jember. Setelah itu dilakukan peramalan beban dengan metode *recurrent neural network* dengan optimasi *levenberg marquardt* dengan menggunakan *software* Matlab. Setelah mendapatkan hasil peramalan, selanjutnya yaitu melakukan analisis perhitungan dan peramalan beban listrik, dan kemudian dilanjutkan dengan membuat kesimpulan dan saran untuk penelitian selanjutnya.

### 3.3 Diagram Alir Peramalan Beban

Untuk mengetahui jumlah kebutuhan energi listrik di wilayah Jember dalam kurun waktu 6 tahun ke depan, dilakukan suatu metode peramalan beban, metode peramalan yang akan digunakan pada penelitian kali ini yaitu dengan menggunakan metode *recurrent neural network* dengan optimasi *levenberg marquardt*. Di bawah ini disajikan gambar diagram alir dari peramalan beban dengan menggunakan *software* Matlab.







Gambar 3.2 Diagram Alir Peramalan Beban

Proses diagram alir pada peramalan beban seperti pada gambar 3.2 di atas dimulai dari inialisasi data dan target dan memasukkannya dalam serangkaian program. Setelah itu memasukkan dan mengatur parameter *training* seperti fungsi transfer, *epoch*, *goal*, *hidden layer*, dan *train* yang disesuaikan dengan kondisi data *input* dan target. Pada penelitian ini menggunakan *training levenberg marquardt* dan arsitektur *recurrent neural network*. Jenis *training levenberg marquardt* ini

dipilih karena sistem *training* ini ialah algoritma baru yang dikembangkan dari *error* dari metode *backpropagation*. Tujuan algoritma tersebut dibuat yaitu untuk menangani beberapa kelemahan algoritma *error* pada metode *backpropagation* menggunakan teknik optimisasi dari numerik standar ialah dengan pendekatan *Jacobian matrix*. Tujuan dari *Levenberg Marquardt* adalah meminimalkan total *error* (Oktaorora, 2011).

Langkah yang selanjutnya yaitu melatih jaringan atau melakukan *training* untuk mendapatkan hasil *training* yang diinginkan, yaitu mendapatkan nilai *error* yang terkecil. Jika hasil *training* belum menghasilkan nilai *error* terkecil, maka program tersebut akan dilakukan *training* lagi sampai hasil *training*nya menghasilkan nilai *error* terkecil kembali. Dari tahapan-tahapan tersebut sampai dengan pengujian (*testing*) data, akan dihasilkan data peramalan beban selama 6 tahun ke depan.

### 3.4 Peramalan Beban Menggunakan *Software Matlab*

Matlab ialah kependekan dari *mathematics laboratory* atau *matrix laboratory*. Pada dunia ilmu komputer, Matlab juga diartikan sebagai bahasa program yang digunakan untuk menyelesaikan suatu operasi aljabar matriks atau operasi matematika. Matlab ialah sistem interaktif yang menggunakan data dasar matriks. Selain itu di dalam Matlab tersedia perangkat lunak (*software*) *simulink*. Manfaat penggunaan *simulink* tersebut yaitu dapat digunakan untuk analisis maupun permodelan suatu sistem. Pada penelitian ini menggunakan *simulink* yang ada pada *software* Matlab.

### 3.5 Arsitektur *Training* dan *Testing Recurrent Neural Network*

Arsitektur yang digunakan pada *recurrent neural network* adalah jaringan berulang atau *looping*. Parameter *input* berisi satu macam data yaitu beban listrik di wilayah Jember dan *output* yaitu data peramalan beban di wilayah Jember selama 6 tahun ke depan. Fungsi transfer yang digunakan dalam peramalan beban ini yaitu *recurrent* dengan menggunakan coba – coba dari berbagai fungsi transfer, dan hasil terbaik yaitu ‘tansig’ untuk *layer input* dan ‘purelin’ untuk *layer output*. Di bawah

ini merupakan penjelasan tahapan proses dalam *training Recurrent Neural Network*.

### 3.5.1 Preprocessing

Data *input* dan target akan mengalami normalisasi yang dikenal juga dengan penskalaan data. Normalisasi ini mempunyai tujuan yaitu untuk mentransformasi data agar kestabilan taburan data dapat tercapai. Menurut Siang (2009), normalisasi berguna untuk menyesuaikan nilai data dengan *range* fungsi aktivasi yang digunakan dalam jaringan, selain itu normalisasi dapat digunakan untuk meningkatkan keakurasian dan hasil *output* serta memfasilitasi proses *learning* dari jaringan syaraf.

Fungsi yang digunakan untuk menormalisasi data yaitu:

```
[pn, meanp, stdp, tn, meant, stdt] = prestd(P, T)
```

Perintah `prestd` dapat menormalisasikan nilai *input* (P) dan target (T) hingga nilai keduanya mempunyai nilai *mean* sebesar 0 dan nilai standar deviasi sebesar 1. Dan fungsi pada `pn` dan `tn` ialah untuk mengubah skala *input* dan target menjadi skala minimum -1 dan maksimum 1. Hal tersebut dapat memberikan kemudahan pada sistem *neural network* agar dapat bekerja dan mencapai *error* yang kecil.

### 3.5.2 Model Jaringan Syaraf

Model jaringan *neural network* yang diterapkan yaitu *multi layer perceptron* dengan 2 *hidden layer*. *Multi layer perceptron* adalah jaringan yang memiliki banyak lapis (*layer*). Fungsi aktivasi yang digunakan yaitu *tansig* dan *purelin*. *tansig* merupakan fungsi aktivasi sigmoid biner. Fungsi aktivasi jenis ini digunakan pada jaringan syaraf yang membutuhkan nilai keluaran yang memiliki nilai interval 0 hingga 1. Fungsi dari sigmoid biner digunakan agar *neuron* dapat aktif pada lapisan *inputnya* agar dapat mengirim informasi melalui bobot-bobotnya ke *neuron* pada *hidden layer*. Sedangkan *purelin* ialah fungsi aktivasi linear yang sering digunakan jika kita menginginkan nilai *output* jaringan yang berupa bilangan *real* sembarang (bukan hanya pada range 0-1 atau pada range -1-1 (Siang, 2009). Fungsi parameter yang digunakan pada sistem peramalan beban ini yaitu sebagai berikut:

```
net=newelm(minmax(pn), [11 12], {'tansig', 'purelin'});
```

```

net.trainFcn = 'trainlm';
net.trainParam.show = 50;
net.trainParam.lr = 0.1;
net.trainParam.epochs = epochs;
net.trainParam.goal = errormax;
net.trainParam.max_fail = 500;
net.trainParam.min_grad = 1e-10;
net.trainParam.mc = 0.5;
net.performFcn = 'mse';

```

*Hidden layer* pada penelitian ini didapat dari hasil coba-coba (*train and error*) menggunakan berbagai jumlah *hidden layer* dan jumlah *neuron* dan kemudian dilakukan *training* data dengan menggunakan *levenberg marquardt*. *Show* yaitu jumlah *epoch* yang akan ditunjukkan kemajuannya, disini berjumlah sebesar 50. *Learning rate* dengan nilai 0,1 (*default*), dan momentum konstanta secara *default* sebesar 0,85. *Epoch* ialah jumlah iterasi maksimal yang dilakukan pengerjaannya selama pada proses pelatihan, pada pelatihan ini berjumlah 1000 *epoch*. *Goal* adalah target nilai fungsi kinerja yang ingin dicapai. *Maximum fail* atau dikenal juga maksimum kegagalan untuk jumlah kegagalan maksimum pada pelatihan, dimana iterasi akan berhenti jika jumlah dari kegagalan lebih dari jumlah maksimum kegagalan. Pada jaringan ini maksimal kegagalan yang digunakan yaitu sebesar 500. Gradien minimum ialah akar dari nilai jumlah kuadrat keseluruhan gradien (bobot *bias*, bobot *input*, bobot lapisan) minimum yang ditoleransi. Dan juga nilai dari iterasi akan berhenti jika nilai dari akar jumlah keseluruhan kuadrat setiap gradien kurang dari nilai gradien minimum. Pada jaringan ini, nilai gradien yang digunakan ialah  $10^{-10}$ . Fungsi kinerja yang dipakai yaitu *Mean Squared Error* (MSE). Nilai *error* yang ingin didapat adalah 0.

### 3.5.3 Training dan Testing

Fungsi yang akan digunakan untuk melakukan proses pelatihan (*training*) ialah seperti dibawah ini:

```
[netRec, tr] = train(net, pn, tn)
```

Model jaringan syaraf yang nantinya akan terbentuk ialah *netRec*, sedangkan *tr* ialah *record* dari proses pelatihan, *pn* dan *tn* ialah matriks *input* dan target suatu jaringan syaraf. Sedangkan fungsi untuk melakukan pengujian ialah seperti di bawah ini:

```
an = trastd(Ujil,meanp, stdp)
```

Fungsi tersebut berfungsi sebagai penormalisasian data agar dapat sesuai dengan data *input* pelatihan. Sedangkan *trastd* akan melakukan *preprocessing* pada data *input* (P) dengan rata-rata dari data *input* (*meanp*) dan standar deviasi data *input* (*stdp*), dan *an* adalah hasil normalisasi data pengujian. Kemudian melakukan pengujian data hasil normalisasi tersebut dengan menggunakan fungsi seperti berikut:

```
cn = sim(net, an)
```

Fungsi dari *cn* yaitu menghasilkan hasil pengujian data normalisasi terhadap jaringan syaraf yang telah terbentuk. Setelah hasil pengujian tersebut didapat maka akan dilakukan denormalisasi pada data tersebut, yaitu mengembalikan data hasil pengujian menjadi data aslinya, yaitu dengan fungsi seperti berikut:

```
c1 = poststd(cn,meant, stdt)
```

Fungsi dari *c1* ialah nilai dalam bentuk denormalisasi dari data pengujian sebelumnya, *cn* yaitu *output* jaringan syaraf pelatihan, *meant* yaitu nilai *mean* dari proses normalisasi, dan *stdt* yaitu nilai standar deviasi dari proses normalisasi.

## BAB 5. PENUTUP

### 5.1 Kesimpulan

Berdasarkan dari hasil dan analisis data pada penjelasan sebelumnya, kesimpulan yang diperoleh ialah seperti berikut:

1. Konfigurasi untuk membangun jaringan pelatihan *recurrent neural network* terkecil yang digunakan sebagai jaringan peramalan beban menggunakan fungsi pelatihan *trainlm*, fungsi sebagai aktivasi *neuron* menggunakan *tansig* untuk layar *input* dan *purelin* untuk layar *output*, serta berhenti saat *hidden layer* ke 1, neuron = 12 serta *error* sebesar  $3,8361E-31$ .
2. Peramalan beban jangka panjang dengan menggunakan metode *recurrent neural network* dengan optimasi *levenberg marquardt* pada saat dilakukan pengujian data beban sejak bulan Januari 2016 sampai dengan bulan Desember 2017 memiliki nilai MAPE (*Mean Absolute Percentage Error*) sebesar 4,455%. Dengan *error* terkecil yaitu 0,097% yang didapat pada bulan April 2016 dengan selisih data beban sebesar 0,0637 MW, dan *error* terbesar yaitu 21,619% yang didapat pada bulan November 2016 dengan selisih data beban sebesar 14,3149 MW. Seperti grafik yang ditunjukkan pada gambar 4.8 yang menunjukkan nilai beban sekunder pada wilayah Jember yang bersifat fluktuatif pada bulan-bulan tertentu sejak tahun 2016 hingga tahun 2017, dimana nilai beban puncak tertinggi yang terjadi pada bulan Desember tahun 2016 ialah sebesar 77,843 MW. Sedangkan nilai beban puncak terendah terjadi pada bulan Juli tahun 2017 yaitu sebesar 58,055 MW.
3. Pengujian beban dengan menggunakan metode *recurrent neural network* dengan optimasi *levenberg marquardt* dan memakai 5 variabel *input* berupa data beban, jumlah penduduk, pertumbuhan ekonomi, pendapatan perkapita, dan pertumbuhan industri memiliki nilai *error* dan MAPE yang lebih kecil daripada menggunakan metode *recurrent neural network* dengan optimasi *levenberg marquardt* dan memakai 1 variabel *input* hanya berupa data beban sekunder. Nilai MAPE pada peramalan menggunakan metode *recurrent neural*

*network* dengan optimasi *levenberg marquardt* dan memakai 5 variabel *input* ialah sebesar 0,116%, dan nilai MAPE pada peramalan menggunakan metode *recurrent neural network* dengan optimasi *levenberg marquardt* dan memakai 1 variabel *input* ialah sebesar 4,455%.

4. Hasil peramalan beban yang didapat dari tahun 2018 sampai dengan tahun 2023 memiliki nilai data beban yang fluktuatif. Nilai data beban tertinggi pada peramalan beban tahun 2018 sampai dengan tahun 2023 dengan metode *recurrent neural network* yaitu sebesar 77,127 MW yang terjadi pada bulan Desember tahun 2020, dan nilai data beban terendahnya yaitu sebesar 50,598 MW yang terjadi pada bulan Oktober tahun 2021.

## 5.2 Saran

Saran yang dianjurkan untuk pengembangan penelitian ini lebih lanjut ke depan dengan harapan mampu memperbaiki kekurangan dan mendapatkan hasil yang lebih maksimal ialah seperti berikut:

1. Penelitian yang dilakukan dalam penelitian kali ini masih dalam tahap *offline* sehingga perlu pengembangan penelitian secara *online*.
2. Sebagai referensi pembanding yang dapat digunakan pada metode lain dalam peramalan beban.

## DAFTAR PUSTAKA

- Adianshar, A. 2014. Penerapan *Recurrent Neural Network* dalam Identifikasi Tulisan Tangan Huruf Jepang Jenis Katakana. *Skripsi*. Medan: Program Studi Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
- Badan Pusat Statistik. 2010. *Kabupaten Jember dalam Angka 2010*. Jember: BPS Kabupaten Jember.
- Badan Pusat Statistik. 2011. *Kabupaten Jember dalam Angka 2011*. Jember: BPS Kabupaten Jember.
- Badan Pusat Statistik. 2012. *Kabupaten Jember dalam Angka 2012*. Jember: BPS Kabupaten Jember.
- Badan Pusat Statistik. 2013. *Kabupaten Jember dalam Angka 2013*. Jember: BPS Kabupaten Jember.
- Badan Pusat Statistik. 2014. *Kabupaten Jember dalam Angka 2014*. Jember: BPS Kabupaten Jember.
- Badan Pusat Statistik. 2015. *Kabupaten Jember dalam Angka 2015*. Jember: BPS Kabupaten Jember.
- Badan Pusat Statistik. 2016. *Kabupaten Jember dalam Angka 2016*. Jember: BPS Kabupaten Jember.
- Badan Pusat Statistik. 2017. *Kabupaten Jember dalam Angka 2017*. Jember: BPS Kabupaten Jember.
- Badan Pusat Statistik. 2015. *Proyeksi Penduduk Kabupaten/Kota Provinsi Jawa Timur 2010-2020*. Jakarta: Badan Pusat Statistik.
- Defit, S. 2013. Perkiraan Beban Listrik Jangka Pendek dengan Metode *Adaptive Neuro Fuzzy Inference System*. *SAINTIKOM*. 12(3): 165-176.
- Dwisatya, R. 2015. Prediksi Beban Listrik Jangka Pendek Berbasis Algoritma Feed Forward Backpropagation dengan Mempertimbangkan Variasi Tipe Hari. *E-Proceeding of Engineering*. 2(3): 7315-7322.
- Handayani, I. 2012. Peramalan Beban Listrik Jangka Pendek Menggunakan Metode Jaringan Syaraf Tiruan. *Setrum*. 1(1): 35-40.
- Jaroslawa, S., K. Pancerz, dan J. Warchol. 2011. Recurrent Neural Networks in Computer-Study. *Computational Intelligence and Neuroscience*. 2011(289398): 1-8.



- Kharis, M. N. 2008. Sistem Tenaga Listrik. <https://eemnkharis27.wordpress.com/2013/08/02/sistem-tenaga-listrik/>. [Diakses pada 3 Juni 2017].
- Kurniawan, A. 2012. Peramalan Awal Musim Hujan Menggunakan Jaringan Syaraf Tiruan Backpropagation Levenberg-Marquardt. *Seminar Nasional Aplikasi Teknologi Informasi 2012 (SNATI 2012)*.
- Lewis, C. D. 1982. *Industrial and Business Forecasting Methods: A Radical Guide to Exponential Smoothing and Curve Fitting*. London: Butterworths.
- Marlia, S. 2016. Peramalan Beban Puncak Harian pada PT. PLN (Persero) APB Jateng dan DIY Menggunakan ANFIS (*Adaptive Neuro-Fuzzy Inference System*). *TRANSIENT*. 5(3): 255-261.
- Marsudi, D. 2006. *Operasi Sistem Tenaga Listrik*. 2. Yogyakarta: Graha Ilmu.
- Minaye, E., dan M. Matewose. 2016. *Long Term Load Forecasting of Jimma Town for Sustainable Energy Supply*. 5(2): 1500-1504.
- Nasution, A. H. 2005. *MANAJEMEN INDUSTRI*. 1. Yogyakarta: Andi.
- Nasution, J. F. 2009. *Implementasi Jaringan Syaraf Tiruan untuk Prediksi Keputusan Medis pada Penyakit Demam Berdarah Dengue*. Skripsi. Medan: Program Studi Sarjana Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sumatera Utara.
- Noviana, A. 2008. Analisa Pengaruh Parameter-Parameter *Neural Network* pada Kasus Pemodelan. Jurusan Teknik Elektro, Fakultas Teknik dan Ilmu Komputer, Universitas Komputer Indonesia.
- Oktaorora, A. 2011. Penentuan Jenis Produk Kosmetik Pilihan Berdasarkan Faktor Usia dan Warna Kulit Menggunakan Metode Jaringan Syaraf Tiruan. *Makalah Seminar Tugas Akhir*. 1-8.
- Pratama, R. A. 2016. Peramalan Beban Listrik Jangka Panjang Provinsi D.I. Yogyakarta Menggunakan *Neural Network Backpropagation*. *Jurnal Teknik Elektro*. 5(3): 37-47.
- Putri, A. A. 2013. Prakiraan Kebutuhan Beban dan Energi Listrik Kabupaten Kendal. *Makalah Kuliah Umum*. Semarang: Seminar Kerja Praktek. 27 Desember.
- SatuEnergi. 2015. Dasar Listrik. <http://www.satuenergi.com/2015/12/bagaimana-listrik-dihantarkan.html>. [Diakses pada 7 Juni 2018].
- Setiabudi, D. 2015. Sistem Informasi Peramalan Beban Listrik Jangka Panjang di Kabupaten Jember Menggunakan *JST Backpropagation*. *SMARTICS Journal*. 1(1): 1-5.

- Siang, J. J. 2005. *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*. Yogyakarta: Andi.
- Sidemen, M. 2014. Peramalan Beban Listrik Harian dengan Metode *Adaptive Neuro Fuzzy Inference System*. *Teknologi Elektro*. 13(1): 12-15.
- Siregar, T. R. 2013. *Implementasi Jaringan Syaraf Metode Bidirectional Associative Memory untuk Pengenalan Pola Wajah*. Skripsi. Medan: Program Studi S1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
- Soelaiman, R., dan Rifa'i, R. 2015. Identifikasi Sistem Nonlinier dengan Menggunakan *Recurrent Neural Network* dan Algoritma *Dead-Zone Kalman Filter*.
- Suhartono. 2009. Peramalan Konsumsi Listrik Jangka Pendek dengan ARIMA Musiman Ganda dan Elman-Recurrent Neural Network. 7(4): 185-192.
- Susanti, I. 2014. Sistem Peramalan Kenaikan Permukaan Air dengan *Artificial Neural Networks Backpropagation*. Skripsi. Bengkulu: Program Studi Informatika Fakultas Teknik Universitas Bengkulu.
- Susanti, M. Handoko, S. dan Winardi B. 2016. Peramalan Beban Puncak Harian pada PT. PLN (Persero) APB Jateng dan DIY Menggunakan *ANFIS (Adaptive Neuro-Fuzzy Inference System)*. *Transient*. 5(3): 255-261.
- Suswanto, D. 2010. *Sistem Distribusi Tenaga Listrik*. Padang: Universitas Negeri Padang.
- Triwulan, Y., N. Hariyanto, dan S. Anwari. 2013. Peramalan Beban Puncak Listrik Jangka Pendek Menggunakan Metode Jaringan Syaraf Tiruan. *Reka Elkomika*. 1(4): 339-350.
- Udin, M. 2017. Peramalan Kapasitas Baterai *Lead Acid* pada Mobil Listrik Berbasis *Levenberg Marquardt Neural Network*. *BERKALA SAINSTEK 2017*. 2: 112-117.
- Ullah, M. I. 2014. Component of Time Series Data. <http://itfeature.com/time-series-analysis-and-forecasting/component-of-time-series-data>. [Diakses pada 23 Juli 2018].
- Zulfa dan Suhartono. 2015. Peramalan Beban Listrik di Jawa Timur Menggunakan Metode ARIMA dan *Adaptive Neuro Fuzzy Inference System (ANFIS)*. *Jurnal Sains dan Seni*. 4(1): D-91-D-96.

## LAMPIRAN

**A. Listing Program Recurrent Neural Network dengan Optimasi Levenberg****Marquardt**

```

clear;
clc;
load JumlahPenduduk.mat;
load JumlahIndustri.mat;
load PendapatanPerkapita.mat;
load PertumbuhanEkonomi.mat;
load DataPelatihan2.mat;

%% Preprocessing
P=Data2(:,1:42)';
P2=JumlahPenduduk(:,1:42)';
P3=JumlahIndustri(:,1:42)';
P4=Perkapita(:,1:42)';
P5=Ekonomi(:,1:42)';
T=Data2(:,43:54)';
T2=JumlahPenduduk(:,43:54)';
T3=JumlahIndustri(:,43:54)';
T4=Perkapita(:,43:54)';
T5=Ekonomi(:,43:54)';

[pn,meanp,stdp,tn,meant,stdt] = prestd(P2,T2);
[pn,meanp,stdp,tn,meant,stdt] = prestd(P3,T3);
[pn,meanp,stdp,tn,meant,stdt] = prestd(P4,T4);
[pn,meanp,stdp,tn,meant,stdt] = prestd(P5,T5);
[pn,meanp,stdp,tn,meant,stdt] = prestd(P,T);

%% Membuat Parameter
hiddenlayer1max = 13;
hiddenlayer2max = 11;
error = 1;
errormax = 0;
modmse = 2e2;
epochs = 1000;

% Pelatihan hidden layer 1
for l1=1:hiddenlayer1max;
    if error > errormax ;
        net=newelm(minmax(pn), [l1 l2], {'tansig', 'purelin'});

        %parameter pelatihan
        net.trainfcn = 'trainlm';
        net.trainParam.show = 50;
        net.trainParam.lr = 0.1;
        net.trainParam.epochs = epochs;
        net.trainParam.goal = errormax;
    end
end

```

```
net.trainParam.max_fail = 500;
net.trainParam.min_grad = 1e-10;
net.trainParam.mc = 0.5;
net.performFcn = 'mse';

%Menampilkan Jumlah Neuron
disp([' Hidden Layer I, Jumlah Neuron = ' num2str(l1)]);

%Train
[netRec,tr]= train(net, pn, tn);

%Simulasi Hasil
Rec= sim(netRec,pn);

%Hitung mse
error=mse(Rec-tn);

%error(l1)=error; %cek
%Ambil Network yg terbaik
if modmse > error;
    layerRec=l1;
    modmse=error;
    hl=1;
    save training.mat;
end
else
    layerl=l1;
    break;
end
end

% Pelatihan hidden layer 2
for l2=1:hiddenlayer2max;
    if error > errormax ;
        net=newelm(minmax(pn),[l1 l2
12],{'tansig','tansig','purelin'});

%parameter pelatihan
net.trainFcn = 'trainlm';
net.trainParam.show = 50;
net.trainParam.lr = 0.1;
net.trainParam.epochs = epochs;
net.trainParam.goal = errormax;
net.trainParam.max_fail = 500;
net.trainParam.min_grad = 1e-10;
net.trainParam.mc = 0.5;
net.performFcn = 'mse';

%Menampilkan Jumlah Neuron
```



```

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

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State,varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before GUIskripsi is made visible.
function GUIskripsi_OpeningFcn(hObject, eventdata, handles,
varargin)

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

% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command
line.
function varargout = GUIskripsi_OutputFcn(hObject, eventdata,
handles)

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

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
A={64.243 59.772 63.852 65.943 64.09 62.9 61.319 64.991 76.959
65.076 66.215 77.843};
B={66.011 62.305 63.325 61.03 65.841 61.999 58.055 63.665
65.178 60.112 65.025 70.975};
Datakul=[A' B'];
set(handles.uitable1,'Data',Datakul);

msgbox('Silahkan klik tombol Pengujian untuk melihat data
pengujian');
set(handles.pushbutton2,'enable','on');
set(handles.uitable2,'enable','on');

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)

```

```
K={63.456 60.243 63.552 66.007 71.103 54.933 61.492 63.982
71.881 57.767 51.9 68.955};
L={62.573 54.544 62.604 60.912 70.851 55.206 65.672 56.694
64.639 51.014 51.683 74.096};
Dataku2=[K' L'];
set(handles.uitable2, 'Data', Dataku2);

msgbox('Silahkan klik tombol Error Persen untuk melihat error
persen data');
set(handles.pushbutton3, 'enable', 'on')
set(handles.uitable3, 'enable', 'on')

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
E={1.225 -0.788 0.47 -0.097 -10.942 12.665 -0.283 1.553 6.599
11.231 21.619 11.418};
F={5.208 12.457 1.139 0.193 -7.61 10.956 -13.121 10.949 0.827
15.136 20.518 -4.397};
Dataku3=[E' F'];
Dataku3=[E' F'];
set(handles.uitable3, 'Data', Dataku3);

msgbox('Silahkan klik tombol MAPE untuk melihat MAPE dari
data');
set(handles.pushbutton4, 'enable', 'on')
set(handles.edit4, 'enable', 'on')

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
mape={4.455};
set(handles.edit4, 'string', mape);

msgbox('Silahkan klik tombol Grafik untuk melihat perbandingan
beban sekunder dengan data pengujian beban RNN untuk tahun 2016
dan 2017');
set(handles.pushbutton5, 'enable', 'on')

% --- Executes during object creation, after setting all
properties.
function edit4_CreateFcn(hObject, eventdata, handles)

% 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 on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
myData=load('DataPLOTGUIPerb', '-mat');
plot(myData.DataPLOTGUIPerb);
title('Grafik Pengujian Beban Menggunakan Metode RNN');
xlabel('Tahun 2016-2017, Bulan ke-');
ylabel('Beban (MW)');
set(handles.text4,'enable','on')

% -----
function PeramalanBeban_Callback(hObject, eventdata, handles)
open('GUIskripsitab2')

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
close

%%GUIskripsitab2
function varargout = GUIskripsitab2(varargin)

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @GUIskripsitab2_OpeningFcn, ...
                  'gui_OutputFcn',    @GUIskripsitab2_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

% --- Executes just before GUIskripsitab2 is made visible.
function GUIskripsitab2_OpeningFcn(hObject, eventdata, handles,
varargin)

% Choose default command line output for GUIskripsitab2
handles.output = hObject;

```



```
% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command
line.
function varargout = GUIskripsitab2_OutputFcn(hObject,
eventdata, handles)

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

% --- Executes on button press in pushbutton100.
function pushbutton100_Callback(hObject, eventdata, handles)
H={64.938 59.92 69.548 59.476 62.325 62.822 60.854 58.697
63.659 55.221 55.267 69.725};
I={68.995 66.619 71.53 66.206 65.146 59.95 61.107 60.069 70.804
60.452 53.988 73.403};
J={69.452 63.996 70.09 64.537 65.518 58.095 63.147 57.601
68.806 56.472 52.249 77.127};
K={66.305 61.136 67.268 63.384 67.422 59.382 63.584 57.699
68.476 50.598 52.444 73.914};
L={65.529 65.231 68.49 66.232 67.389 61.973 61.53 59.89 71.628
50.765 53.031 71.149};
M={68.846 67.014 70.511 66.448 65.032 59.83 61.631 58.509
69.458 53.939 54.035 75.012};
Dataku4=[H' I' J' K' L' M'];
set(handles.uitable100,'Data',Dataku4)

msgbox('Silahkan klik tombol grafik untuk melihat grafik
peramalan beban');

set(handles.pushbutton101,'enable','on')

% --- Executes on button press in pushbutton101.
function pushbutton101_Callback(hObject, eventdata, handles)
myData1=load('DataPLOTGUItab2', '-mat');
plot(myData1.DataPLOTGUItab2);
title('Grafik Perbandingan Peramalan Beban Metode RNN dan
Metode BNN');
xlabel('Tahun 2018-2023, Bulan ke-');
ylabel('Beban (MW)');
set(handles.text4,'enable','on')

% -----
function PengujianRNN_Callback(hObject, eventdata, handles)
open('GUIskripsi')
```

```
% --- Executes when entered data in editable cell(s) in
uitable100.
function uitable100_CellEditCallback(hObject, eventdata,
handles)

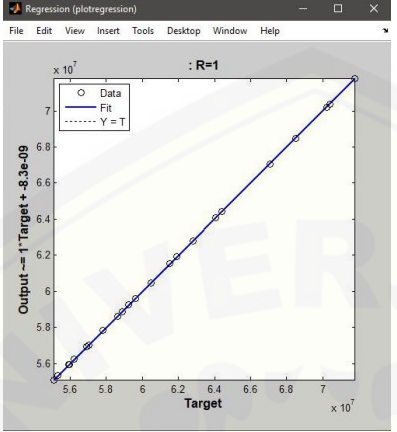
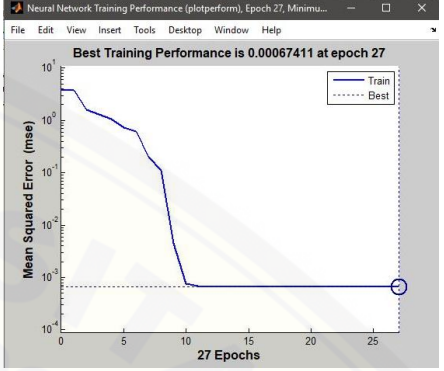
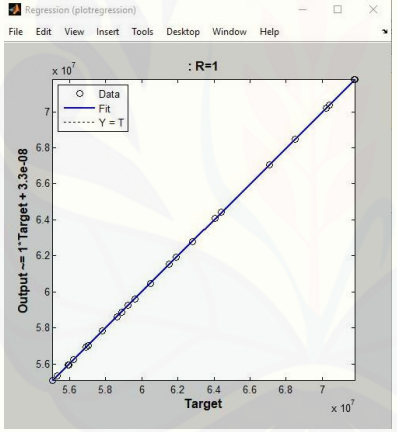
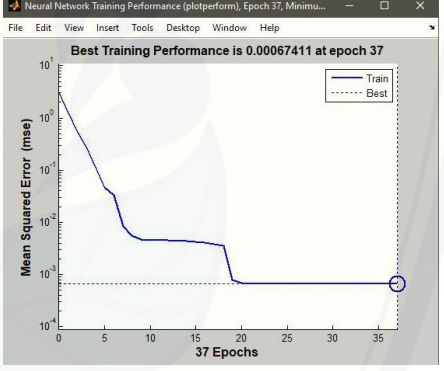
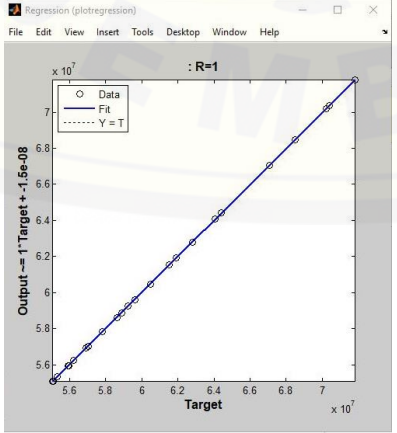
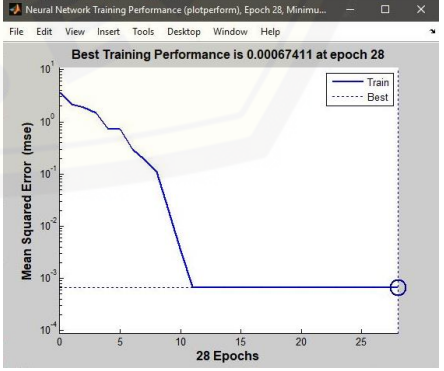
% --- Executes during object creation, after setting all
properties.
function text4_CreateFcn(hObject, eventdata, handles)
```

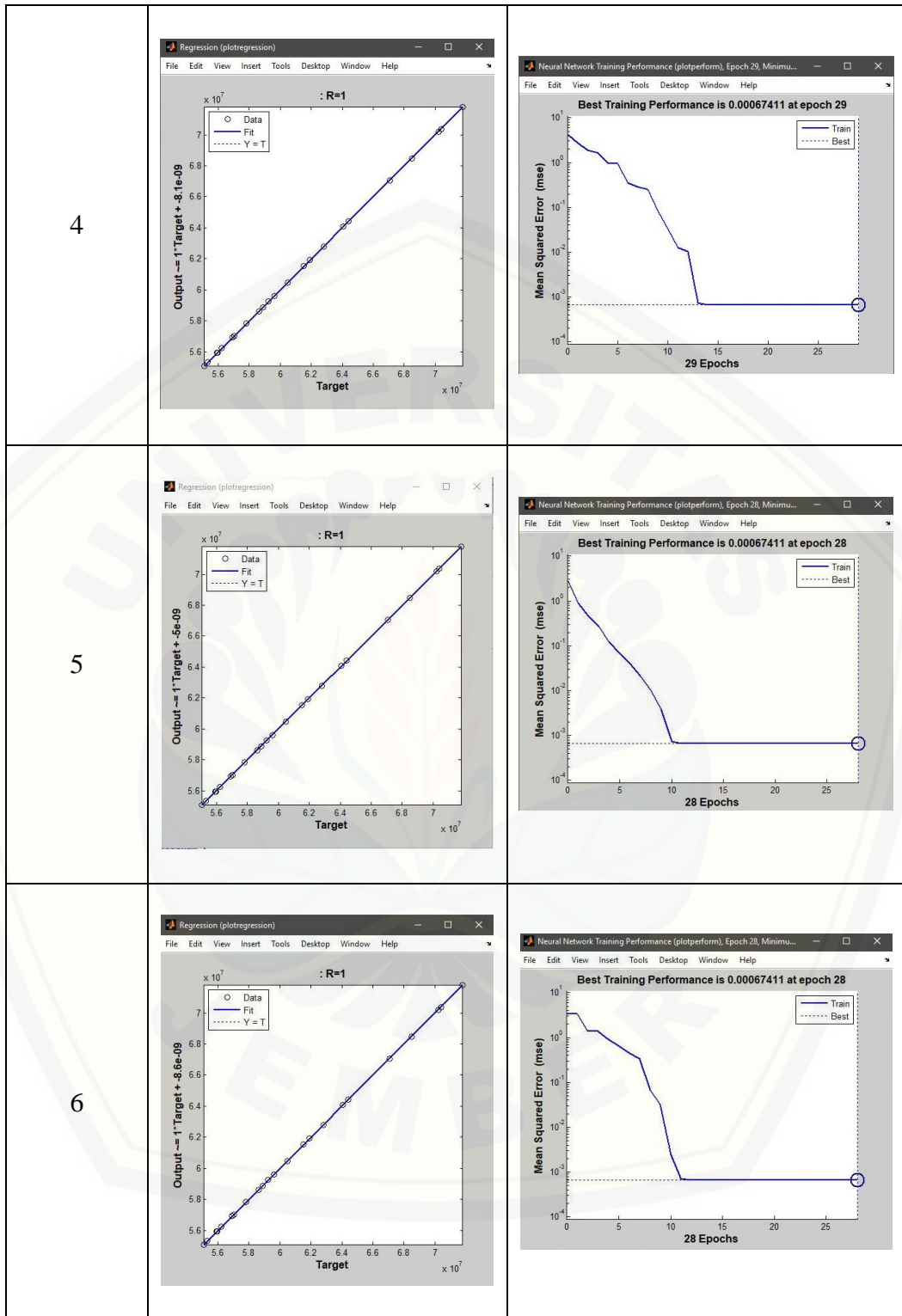


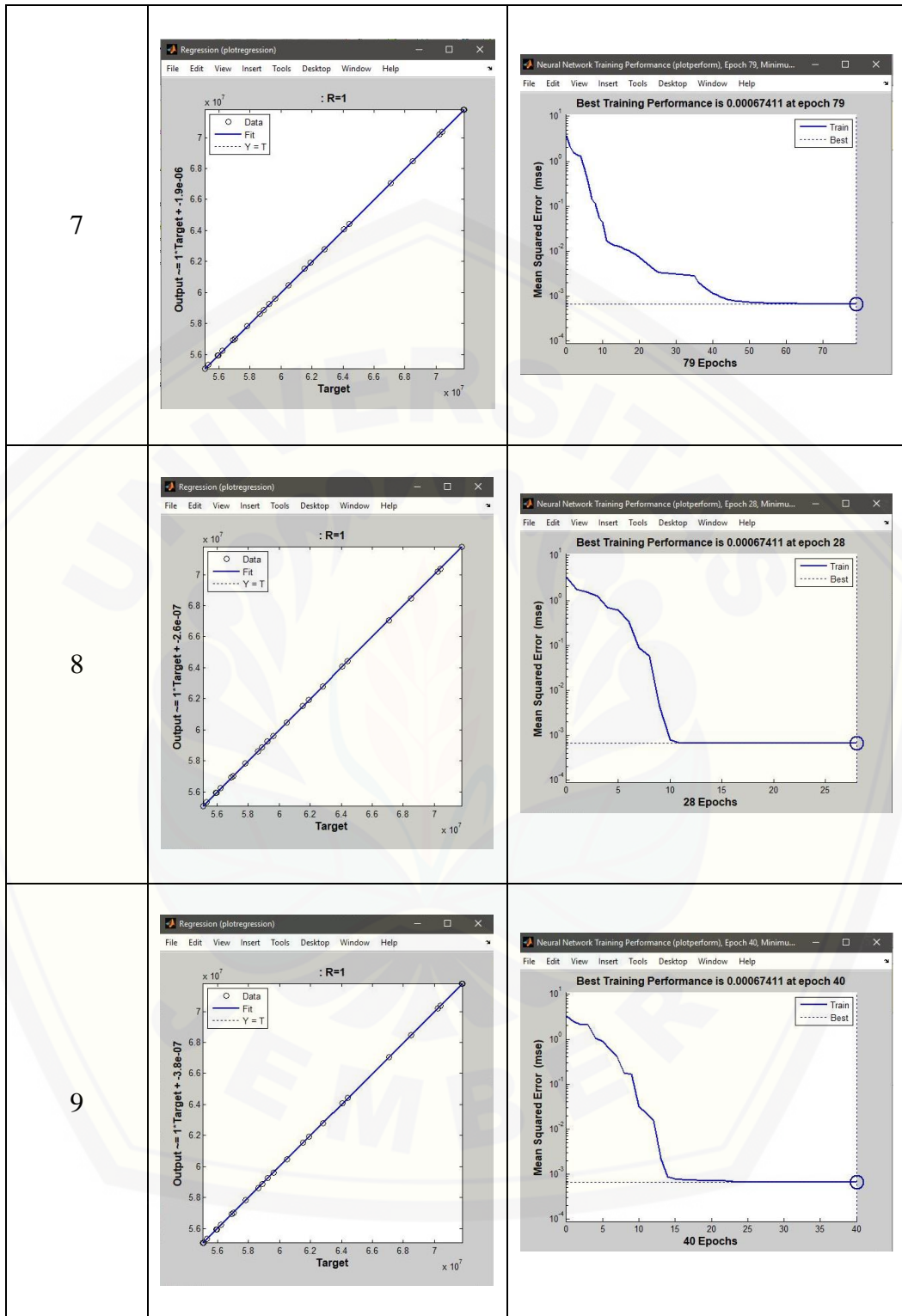
**C. Tabel Data Beban Puncak Wilayah Jember**

No	Bulan	Beban (MW)							
		2010	2011	2012	2013	2014	2015	2016	2017
1	Januari	42.024	50.303	46.784	51.799	58.888	64.073	64.243	66.011
2	Februari	40.477	41.429	48.178	54.842	55.930	56.236	59.772	62.305
3	Maret	42.007	50.303	59.517	56.304	55.947	64.413	63.852	63.325
4	April	42.058	44.982	46.291	57.562	55.318	60.469	65.943	61.030
5	Mei	44.336	44.387	58.718	52.411	67.065	70.210	64.090	65.841
6	Juni	42.806	49.776	47.192	50.796	59.245	56.916	62.900	61.999
7	Juli	41.310	56.780	48.552	66.997	57.834	57.035	61.319	58.055
8	Agustus	40.817	45.288	54.179	52.751	55.080	61.523	64.991	63.665
9	September	48.416	60.384	51.612	56.049	58.633	70.380	76.959	65.178
10	Oktober	48.909	48.365	57.001	61.098	59.619	71.791	65.076	60.112
11	November	41.922	50.388	51.612	52.819	62.798	61.914	66.215	65.025
12	Desember	41.293	50.847	57.919	62.237	62.798	68.476	77.843	70.975

**D. Gambar Hasil Percobaan *Training Recurrent Neural Network***

<i>Training</i>	<i>Regression</i>	<i>Performance</i>
1		
2		
3		





10

