



**STUDI PERAMALAN BEBAN PUNCAK JANGKA PANJANG
MENGUNAKAN JARINGAN SARAF TIRUAN LEVENBERG
MARQUARDT DITINJAU DARI KAPASITAS TRANSFORMATOR GI
JEMBER PADA PT. PLN (PERSERO) APJ JEMBER**

SKRIPSI

Oleh
Septa Rahardimas
NIM 151910201039

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2019**



**STUDI PERAMALAN BEBAN PUNCAK JANGKA PANJANG
MENGUNAKAN JARINGAN SARAF TIRUAN LEVENBERG
MARQUARDT DITINJAU DARI KAPASITAS TRANSFORMATOR GI
JEMBER PADA PT. PLN (PERSERO) APJ JEMBER**

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
Septa Rahardimas
NIM 151910201039

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2019**

PERSEMBAHAN

Alhamdulillah, segala puji bagi Allah SWT atas segala karunia dan rahmat-Nya kepada penulis sehingga penulis dapat menyelesaikan skripsi yang berjudul “Studi Peramalan Beban puncak Jangka Panjang Menggunakan jaringan saraf tiruan LEVENBERG MARQUARDT Ditinjau dari Kapasitas Transformator GI Jember pada PT. PLN (Persero) APJ Jember” dan skripsi ini merupakan gerbang awal dalam mencapai kesuksesan yang lebih baik lagi. Maka dari itu, saya ingin mempersembahkan karya ini kepada :

1. Ibu Sri Rahayu dan Ayah Junif Junaidi sebagai orang tua yang telah mendidik sampai dewasa.
2. Semua Dosen Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember yang telah memberikan ilmu dan membimbing kami selama dibangku perkuliahan. Terutama Dosen Pembimbing Bapak Suprihadi Prasetyono, ST., MT dan Prof. Dr. Ir. Bambang Sujanarko, M.M. yang telah membimbing dan meluangkan waktunya untuk membantu hingga terselesaikannya skripsi ini.
3. Saudaraku seperjuangan Teknik Elektro angkatan 2015 “DISTORSI”.
4. Almamater Fakultas Teknik Universitas Jember.
5. Muhammad Haidir Z, S.T., Martin Sanjaya, S.T., dan Ria yulia yang telah membantu dan menyemangati sampai selesai.
6. 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

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

(Q.S. At-Taubah : 40)

“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)

“Taklukkan hati allah maka kamu akan merasakan kemudahan karena allah akan selalu ada di sampingmu ”

(Septa Rahardimas)

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Septa Rahardimas

NIM : 151910201039

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul : “Studi Peramalan Beban puncak Jangka Panjang Menggunakan Jaringan Saraf Tiruan *Levenberg Marquardt* Ditinjau dari Kapasitas Transformator GI Jember pada PT. PLN (Persero) APJ Jember” 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, 21 Mei 2019

Yang menyatakan,

Septa Rahardimas

NIM 151910201039

SKRIPSI

**STUDI PERAMALAN BEBAN PUNCAK JANGKA PANJANG
MENGUNAKAN JARINGAN SARAF TIRUAN LEVENBERG
MARQUARDT DITINJAU DARI KAPASITAS TRANSFORMATOR GI
JEMBER PADA PT. PLN (PERSERO) APJ JEMBER**

Oleh
Septa Rahardimas
NIM 151910201039

Pembimbing :

Dosen Pembimbing Utama : Suprihadi Prasetyono, ST., MT
NIP : 197004041996011001

Dosen Pembimbing Anggota : Prof. Dr. Ir. Bambang Sujanarko, M.M.
NIP 196312011994021002

PENGESAHAN

Skripsi berjudul “Studi Peramalan Beban puncak Jangka Panjang Menggunakan Jaringan Saraf Tiruan *Levenberg Marquardt* Ditinjau dari Kapasitas Transformator GI Jember pada PT. PLN (Persero) APJ Jember” Karya Septa Rahardimas telah diuji dan disahkan pada :

Hari : Selasa

Tanggal : 21 Mei 2019

Tempat : Fakultas Teknik Universitas Jember

Tim Penguji,

Ketua,

Anggota I,

Suprihadi Prasetyono, ST., MT
NIP : 197004041996011001

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

Anggota II,

Anggota III,

Ir. Widyono Hadi, MT.
NIP 196104141989021001

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

Mengesahkan,
Dekan Fakultas Teknik

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

RINGKASAN

Studi Peramalan Beban puncak Jangka Panjang Menggunakan Jaringan Saraf Tiruan *Levenberg Marquardt* Ditinjau dari Kapasitas Transformator GI Jember pada PT. PLN (Persero) APJ Jember; Septa Rahardimas, 151910201039; 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 puncak yang digunakan juga akan semakin besar. Untuk mengantisipasi hal tersebut, dibutuhkan metode agar dapat memprediksikan jumlah beban puncak 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 puncak untuk memprediksi jumlah beban yang dibutuhkan di dalam beberapa waktu ke depan.

Penelitian untuk meramalkan beban puncak ini menggunakan simulasi dengan *software* Matlab. Data yang digunakan pada penelitian kali ini merupakan data beban puncak listrik setiap bulan selama 7 tahun yang terbagi menjadi 3 data, yaitu data *input*, data target dan data uji . Metode yang diterapkan pada penelitian kali ini yaitu metode *levenberg marquardt*. Data yang dipakai untuk pelatihan penelitian ini yaitu data pada tahun 2011 sampai dengan pada tahun 2017, dimana dari data-data tersebut akan dibagi menjadi data *input*, data target dan data uji pelatihan *recurrent neural network*. Data tersebut disusun menjadi 36 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 puncak 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 16,11166595%. 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 11% hingga 20% di kategorikan ramalan bagus dan hasil MAPE yang diperoleh yaitu 16,11166595% dapat diartikan. Dari data rekap peramalan beban mulai tahun 2018 sampai dengan 2023 berdasarkan hasil perhitungan total beban puncaknya disetiap tahun. Menurut RUPTL 2016-2025 untuk penambahan kapasitas trafo di suatu GI ditentukan pada saat pembebanan trafo mencapai 60%-70% dari kapasitas maksimal transformator yang terpasang saat ini yaitu 5773,67Amp. Sehingga 70% dari kapasitas transformator 5773,67Amp diperoleh 4041,171Amp. Bahwa untuk perkiraan beban puncak diperkirakan baru melewati kapasitas transformator pada Gardu Induk Jember pada tahun 2023 beban puncak yaitu pada bulan Maret sebesar 5031,254Amp, Juni sebesar 4657,63Amp, dan September sebesar 4162,68Amp yaitu melewati maksimal 4041,171Amp dari 70% kapasitas trafo yaitu 5773,67Amp.

PRAKATA

Puji syukur kehadiran Allah SWT, atas segala limpahan karunia dan rahmat-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Studi Peramalan Beban puncak Jangka Panjang menggunakan jaringan saraf tiruan Levenberg Marquardt Ditinjau dari Kapasitas Transformator GI Jember pada PT. PLN (Persero) APJ Jember”. 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. Ibu Dr. Ir. Entin Hidayah M.U.M selaku Dekan Fakultas Teknik Universitas Jember.
2. Bapak Suprihadi Prasetyono, ST., MT selaku Dosen Pembimbing Utama dan Bapak Prof. Dr. Ir. Bambang Sujanarko, M.M. selaku Dosen Pembimbing Anggota yang telah membantu dan meluangkan waktunya, serta memberikan bimbingan dalam penyusunan penelitian ini.
3. Bapak Prof. Dr. Ir. Bambang Sujanarko, M.M., selaku dosen penguji utama dan Bapak Dr. Bambang Sri Kaloko, S.T., M.T., selaku dosen penguji anggota yang telah membantu memberikan kritik dan saran yang dapat membangun sehingga sangat membantu terhadap penyempurnaan penyusunan skripsi ini.
4. Bapak R. B. Moch. Gozali, ST., MT. selaku dosen pembimbing akademik yang telah membimbing dan menanamkan rasa disiplin dan tanggung jawab dengan apa yang dilakukan selama penulis menjadi mahasiswa.
5. Para dosen beserta seluruh staf karyawan Fakultas Teknik Universitas Jember, terima kasih atas segala bantuan dan dukungannya selama ini.
6. Ibu Sri Rahayu dan Ayah Junif Junaidi tercinta, yang telah mendidik, memberi kasih sayang dan mendoakan.
7. Muhammad Haidlir Z, S.T., Martin Sanjaya, S.T., dan Ria Yulia yang selalu ada untuk membantu dan memberi semangat untuk penulis setiap waktu dan tanpa mengenal lelah.
8. Saudaraku elektro angkatan 2015 “DISTORSI” terima kasih kalian adalah inspirasi dan penyemangatku.

9. Keluarga Kosan 57 dikala senang dan susah Mas Yoga, Mas Riski, Mas Ilham, Mas Dwi, Tama, Yoga, Novi, Arip, Dio, Hilmi, Yoga Buki, Farhan, Riky, dan Sahrul
10. Keluarga UKM PAMADIKSI Mas Rendi, Mas Martin, Mas Julio, MbK Astri, MbK May, Amin, Martina, Sulis, Hendra, Evi, Alfi, Chusnul, Savira, Agung, Yahya, Mila, Firda dan yang lainnya.
11. Keluarga KKN Muneng 278 di kala senang dan susah Nilam, Shita, Ghesi, Danang, Hayati, Intan, Safira, Silvia, dan Desita
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, 21 Mei 2019

Penulis

DAFTAR ISI

HALAMAN SKRIPSI.....	Error! Bookmark not defined.
HALAMAN PERSEMBAHAN	ii
HALAMAN MOTTO	iii
HALAMAN PERNYATAAN.....	iv
HALAMAN PEMBIMBING	v
HALAMAN PENGESAHAN.....	vi
HALAMAN RINGKASAN	vii
HALAMAN PRAKATA	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xiv
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah	4
1.4 Tujuan.....	4
1.5 Manfaat.....	5
BAB 2. TINJAUAN PUSTAKA.....	6
2.1 Sistem Tenaga Listrik.....	6
2.1.1 Sistem Transmisi.....	6
2.1.2 Sistem Distribusi.....	7
2.2 Sistem Pendistribusian Tenaga Listrik.....	8
2.2.1 Sistem Pendistribusian Langsung	8
2.2.2 Sistem Pendistribusian Tidak Langsung.....	8
2.3 Karakteristik Beban Tenaga Listrik.....	9
2.4 Waktu Beban Puncak (WBP)	9
2.5 Perkiraan Bebabn Tenaga Listrik.....	10
2.5.1 Jenis dan Akibat Kesalahan Peramalan	11

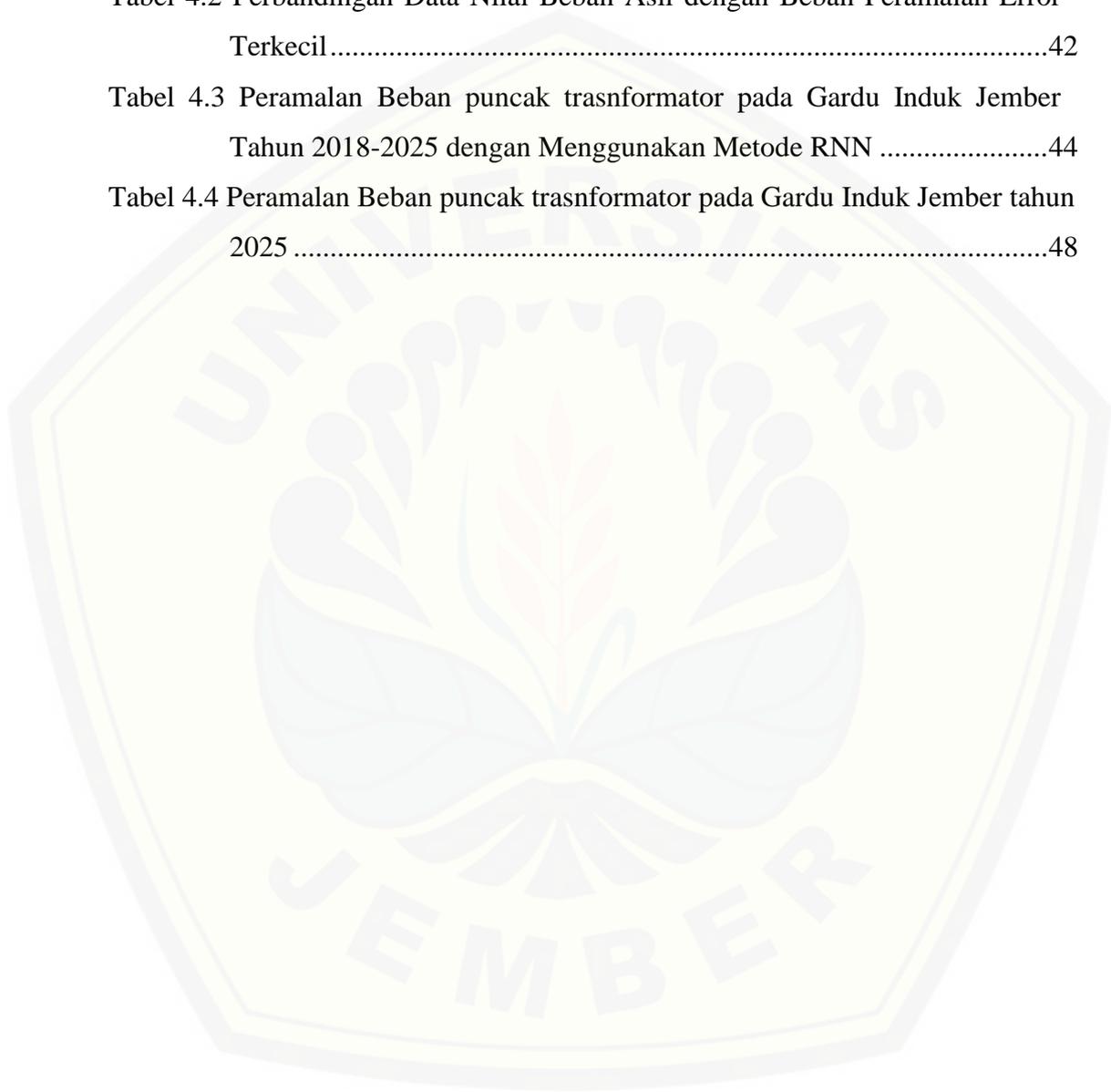
2.5.2 Jenis Kesalahan Peramalan	11
2.5.3 Akibat Kesalahan Peramalan	11
2.6 Analisa Data Berkala	12
2.7 <i>Neural Network</i> (NN)	13
2.7.1 Arsitektur <i>Neural Network</i>	16
2.7.2 Fungsi Transfer	18
2.8 <i>Recurrent Neural Network</i>	19
2.9 <i>Levenberg Marquardt</i>	20
BAB 3. METODE PENELITIAN.....	23
3.1 Prosedur Penelitian.....	23
3.2 Diagram Alir Penelitian	26
3.3 Diagram Alir Peramalan Beban	27
3.4 Peramalan Beban Menggunakan <i>Software</i> Matlab	29
3.5 Arsitektur <i>Training</i> dan <i>Testing Recurrent Neural Network</i>	30
3.5.1 <i>Preprocessing</i>	30
3.5.2 Model Jaringan Syaraf	30
BAB 4. HASIL DAN PEMBAHASAN.....	34
4.1 Data Beban Gardu Induk Jember	34
4.2 Data Pelatihan <i>Recurrent Neural Network</i>	35
4.3 Hasil dan Analisis Peramalan Beban Menggunakan Metode <i>Recurrent Neural Network</i> dengan Optimasi <i>Levenberg</i> <i>Marquardt</i>	37
4.4 Hasil dan Analisa Pengujian Data Peramalan Beban pada <i>Recurrent Neural Network</i>	40
4.5 Analisa Hasil Kapasitas Beban puncak Trasformator Pada Gardu Induk Jember	46
4.6 Hasil dan Pengujian Data Peramalan Beban pada <i>Recurrent</i> <i>Neural Network</i> pada <i>Graphical User Interface</i> (GUI).....	48
BAB 5. PENUTUP.....	53
5.1 Kesimpulan.....	53
5.2 Saran	53
DAFTAR PUSTAKA	54
LAMPIRAN.....	57

DAFTAR GAMBAR

Gambar 2.1 Sistem Tenaga Listrik.....	6
Gambar 2.2 Konfigurasi Transmisi Tenaga Listrik	7
Gambar 2.3 Pola Variasi <i>Trend</i> Permintaan Daya Listrik	13
Gambar 2.4 Bentuk Dasar <i>Neuron</i>	14
Gambar 2.5 Arsitektur <i>Single Layer Network</i>	17
Gambar 2.6 <i>Multi Layer Network</i>	18
Gambar 2.7 Struktur dari <i>Recurrent Neural Network</i>	20
Gambar 3.1 Diagram Alir Penelitian	26
Gambar 3.2 Diagram Alir Peramalan Beban	28
Gambar 4.1 Grafik Perubahan Data Beban puncak Trasformator di GI Jember dari Tahun 2010 – 2017 (Data Sekunder PT. PLN (Persero) APJ Jember.....	34
Gambar 4.2 nilai <i>error</i> yang terkecil pada pelatihan 12	38
Gambar 4.3 Grafik <i>Performance</i> Pelatihan 12 <i>Recurrent Neural Network</i>	39
Gambar 4.4 Grafik <i>Regression</i> Pelatihan 12 <i>Recurrent Neural Network</i>	39
Gambar 4.5 Grafik <i>Training State</i> Pelatihan 12 <i>Recurrent Neural Network</i>	40
Gambar 4.6 Tampilan GUI Data untuk Pengujian Tahun 2016 - 2017	49
Gambar 4.7 tampilan GUI Grafik Untuk Pengujian Tahun 2016 – 2017	50
Gambar 4.8 Tampilan GUI Data untuk Peramalan Beban Tahun 2018 - 2025	51
Gambar 4.9 Tampilan GUI Grafik untuk Peramalan Beban Tahun 2018 – 2025..	51

DAFTAR TABEL

Tabel 4.1 Hasil <i>Error</i> dari Pelatihan <i>Recurrent Neural Network</i>	37
Tabel 4.2 Perbandingan Data Nilai Beban Asli dengan Beban Peramalan Error Terkecil.....	42
Tabel 4.3 Peramalan Beban puncak transformator pada Gardu Induk Jember Tahun 2018-2025 dengan Menggunakan Metode RNN	44
Tabel 4.4 Peramalan Beban puncak transformator pada Gardu Induk Jember tahun 2025	48



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. Meningkatnya aktivitas kehidupan manusia secara langsung akan mengakibatkan tingginya permintaan energi listrik yang mengakibatkan penambahan beban pada jaringan listrik. Seiring berjalannya waktu, beban puncak 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).

Untuk mengetahui jumlah beban puncak listrik jangka panjang di Indonesia, perusahaan listrik negara melakukan prakiraan beban pada jangka waktu diatas satu tahun. Hal ini di gunakan untuk menentukan kapasitas penambahan sistem pembangkit, transmisi, dan distribusi. Apabila prakiraan yang diperoleh belum sesuai dengan yang diharapkan, maka hal ini dapat dioptimalkan. Apabila kapasitas pembangkit dirasa tidak cukup, maka pemerintah akan merancang strategi agar kebutuhan listrik dimasa mendatang bisa terpenuhi seperti persiapan dana, infrastruktur dan persiapan lainnya (Rahman,ade dan Dadang, 2012)

Daya yang akan dibangkitkan harus selalu sama dengan daya yang dikonsumsi oleh para pemakai tenaga listrik yang secara teknis umumnya dikatakan sebagai beban sistem (Marsudi, 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.

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 puncak 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 puncak (Defit, 2013).

Dari uraian yang telah dijelaskan, penelitian ini akan dilaksanakan dengan menggunakan metode *recurrent neural network levenberg marquardt*. Dimana penelitian peramalan beban serupa dengan menggunakan metode tersebut pernah dilakukan oleh Mambak Udin. Metode yang digunakan adalah metode *recurrent neural network levenberg marquardt* tentang meramalkan kapasitas baterai *lead acid* pada mobil listrik yang menggunakan jaringan saraf tiruan *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*.

Gardu induk adalah bagian dari suatu sistem tenaga yang dipusatkan pada suatu tempat berisi saluran transmisi dan distribusi. Gardu induk berfungsi untuk mengatur aliran daya listrik dari saluran transmisi yang satu ke saluran transmisi yang lain, mendistribusikan ke konsumen, sebagai tempat untuk menurunkan tegangan transmisi menjadi tegangan distribusi. Dalam penyaluran daya Gardu Induk memiliki alat/komponen listrik yang berupa transformator daya yang berfungsi untuk mentransformasikan daya listrik, dengan merubah besarnya tegangan sedangkan frekuensinya tetap. Dalam operasi penyaluran tenaga listrik

transformator dapat dikatakan jantung dari sistem transmisi dan distribusi. Dalam kondisi ini suatu transformator diharapkan dapat beroperasi secara maksimal. Mengingat kerja keras dari suatu transformator seperti itu, maka pemeliharaan juga dituntut sebaik mungkin.

Berdasarkan tegangan daya yang dihasilkan dapat dibedakan menjadi transformator 500/150 kV dan 150/70 kV biasa disebut *Interbus Transformer* (IBT). Transformator 150/20 kV dan 70/20 kV disebut juga transformator distribusi. Transformator distribusi digunakan untuk menurunkan tegangan listrik dari jaringan transmisi tegangan tinggi menjadi tegangan terpakai pada jaringan distribusi tegangan rendah (*step down transformer*).

Pada dasarnya transformator tiga fasa ini terdiri dari tiga buah transformator satu fasa dengan tiga buah teras besi yang dipasang pada satu kerangk. Dari tiga teras besi ini ditempatkan masing-masing sepasang kumparan primer dan kumparan sekunder. Dengan demikian seluruhnya akan terdapat tiga buah kumparan primer dan tiga buah kumparan sekunder. Dari ketiga kumparan primer maupun kumparan sekunder dapat dihubungkan bintang (*star connection*) Y dan dihubungkan segitiga (*delta connection*) Δ . Dalam transformator satu fasa besarnya ggl induksi tersebut sama besarnya dan berlawanan arah dengan tegangannya, maka untuk transformator tiga fasa besarnya tegangan tergantung pada hubungan antara kumparan primer dan kumparan sekunder (Za'im, 2013).

Berdasarkan data perkembangan beban di Gardu Induk pada PT. PLN (Persero) Cabang jember dengan menggunakan transformator daya 3 fasa 20 kV yang ada di Gardu Induk jember berjumlah 4 transformator, yang meliputi transformator 20 kV 60 MVA sebanyak 3 buah trafo dan transformator 20 kV 20 MVA sebanyak 1 buah trafo. Dapat dilihat di tahu 2017 mengalami pembangunan di bidang perekonomian dan bisnis antara lain pembangunan transmar jember di Jalan Gajahmada, pembangunan apartement di jalan perempatan tidar dan kedepanya pengembangan industri kecil, menengah dan besar di Puger, Tanggul dan Arjasa.

1.2 Rumusan Masalah

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

1. Bagaimana kondisi pertumbuhan beban listrik transformator di GI Jember?
2. Bagaimana perbedaan tingkat akurasi pengujian beban antara beban asli Transformator GI Jember dengan hasil data peramalan beban menggunakan Matlab ?
3. Kapan Transformator yang ada di GI Jember mencapai batas maksimal penyaluran Energi Listrik ?

1.3 Batasan Masalah

Adapun beberapa batasan masalah yang digunakan pada penelitian ini yaitu sebagai:

1. Peramalan beban dilakukan sebagai peramalan beban puncak bulanan, dengan data input menggunakan data beban historis selama 7 tahun yaitu dari tahun 2011 sampai dengan tahun 2017, dengan membaginya menjadi dua data, yaitu data pelatihan (*training*) dan data pengujian (*testing*).
2. Tidak membahas masalah gangguan dari dalam dan luar.
3. Pasokan energi listrik dari Gardu induk Jember dianggap tetap atau tidak ada penambahan.
4. Simulasi peramalan beban yang digunakan menggunakan *software Matlab*.

1.4 Tujuan

Tujuan yang akan dicapai dalam melaksanakan penelitian ini adalah sebagai berikut:

1. Untuk mengetahui kondisi pertumbuhan beban di Gardu Induk Jember
2. Untuk mengetahui tingkat akurasi kemampuan peramalan beban menggunakan *software matlab*.
3. Mengetahui sampai kapan Transformator pada Gardu Induk Jember dapat mentransmisikan daya dan arus maksimal.

1.5 Manfaat

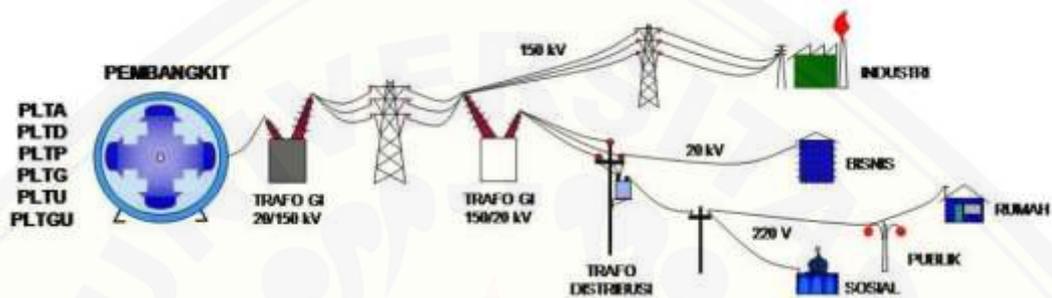
Dengan dilakukannya penelitian ini, diharapkan dapat memberikan manfaat yaitu mengetahui hasil analisis kapasitas Transformator terhadap pertumbuhan beban pada GI Jember dapat diperoleh perkiraan sampai kapan transformator dapat beroperasi dengan maksimal. Dan diharapkan dapat membantu pihak pemerintah dalam merencanakan pembangunan jalur transmisi baru maupun pembangkit listrik baru.



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.



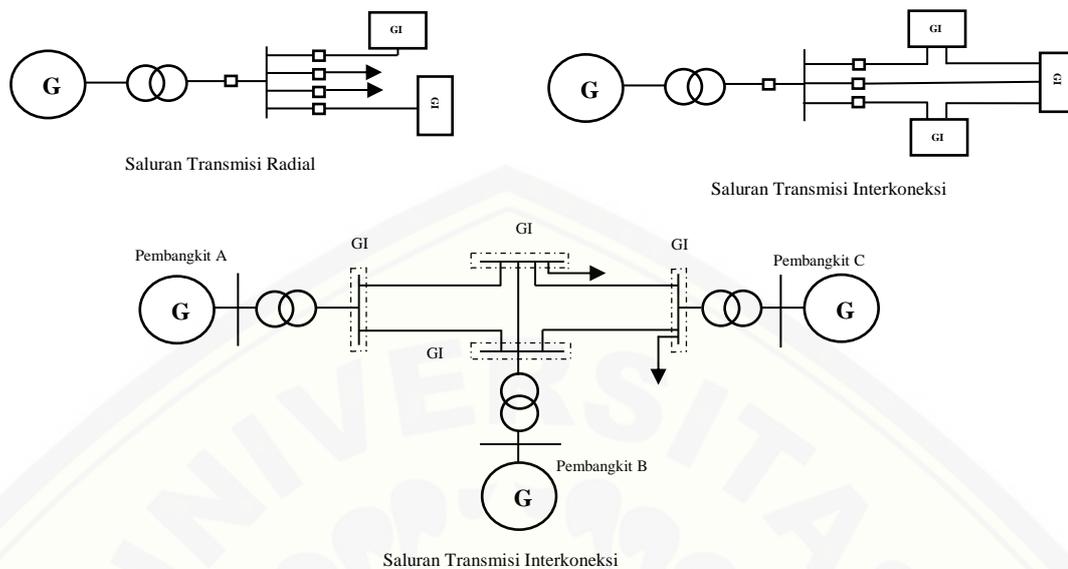
Gambar 2.1 Sistem Tenaga Listrik

(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 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 2.2 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 2.2 Konfigurasi Transmisi Tenaga Listrik

(Kharis, diakses pada 7 Juni 2018)

Gardu induk sendiri ialah bagian (sub sistem) dari sistem transmisi, yang berfungsi untuk (Kharis, 2008):

- 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.

2.1.2 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.

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 (Suswanto, 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 puncak, 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 Karakteristik Beban Tenaga Listrik

Secara umum beban tenaga listrik yang di supply oleh pembangkit listrik dapat dikategorikan kedalam beberapa sektor, yaitu sektor perumahan, sektor industri, sektor komersial dan sektor usaha. Setiap sektor memiliki karakteristik pembebanan yang berbeda-beda, hal ini disebabkan oleh pola konsumsi energi dari masing-masing sektor tersebut. Pada sektor perumahan, pembebanan ditujukan oleh adanya fluktuasi konsumsi energi listrik yang cukup besar. Penyebab dari fluktuasi ini adalah konsumsi energi listrik yang cukup besar pada malam hari. Untuk sektor industri memiliki fluktuasi yang kecil atau hampir sama tiap harinya, sehingga perbandingan beban puncak terhadap beban rata-rata hampir mendekati satu. Beban pada sektor komersial dan usaha mempunyai karakteristik yang hampir sama, hanya pada sektor komersial akan mempunyai beban puncak yang lebih tinggi pada malam hari.

2.4 Waktu Beban Puncak (WBP)

Waktu beban puncak adalah waktu tertentu tingkat pemakaian daya listrik konsumen yang telah mencapai tingkat puncak kapasitas pembebanan. PLN telah menetapkan waktu beban puncak ini mulai pukul 17:00 WIB sampai dengan 22:00 WIB. Namun pada kenyataan di setiap daerah memiliki waktu beban puncak yang berbeda. Ada sebagian daerah terjadinya waktu beban puncak dimulai pada jam 18:00 sampai dengan 00:00. Waktu beban puncak terjadi disebabkan beban puncak yang melebihi daya pasok dari pembangkit listrik.

Pada saat waktu beban puncak, konsumsi listrik meningkat drastis dibandingkan luar waktu beban puncak. Penggunaan listrik secara berlebihan saat waktu beban puncak akan menyebabkan tarif listrik yang dibayar akan lebih mahal dibandingkan luar waktu beban puncak. Selain itu pemadaman listrik secara bergilir akan terjadi jika kebutuhan daya pada konsumen lebih besar dibandingkan dengan pasokan daya dari pembangkit. (Mulyadi, 2017)

2.5 Perkiraan Beban Tenaga Listrik

Secara umum beban puncak yang di *supply* merupakan suatu hal yang perlu diperhatikan dalam beberapa sektor perencanaan sistem tenaga listrik dan besarnya beban merupakan kriteria perencanaan sangat penting. Untuk itu diperlukan suatu cara untuk dapat meramalkan atau memperkirakan suatu nilai beban puncak dimasa mendatang. Peramalan pada beban tenaga listrik (Electric Power Load Forecasting) yang akurat mengakibatkan penghematan substansial dalam biaya operasi dan pemeliharaan, peningkatan kehandalan pasokan listrik dan sistem pengiriman, dan keputusan yang benar untuk embangunan masa depan. Biasanya peramalan pada bidang tenaga listrik ini berupa peramalan beban puncak (MW) dan peramalan kebutuhan energi listrik (*demand forecasting*) (MWh). Menurut 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 puncak 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 merupakan suatu peramalan beban yang memperkirakan besar suatu beban puncak dalam jangka waktu dari satu bulan sampai dengan satu tahun.

c. Peramalan Beban Jangka Pendek

Peramalan beban jangka pendek merupakan suatu peramalan beban yang memperkirakan besar suatu beban puncak dalam jangka waktu beberapa jam, hari, sampai satu 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 puncak ini 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 nya biaya pelayanan energi listrik juga termasuk hal yang berhubungan dengan kesalahan pada peramalan.

2.6 Analisa Data Berkala

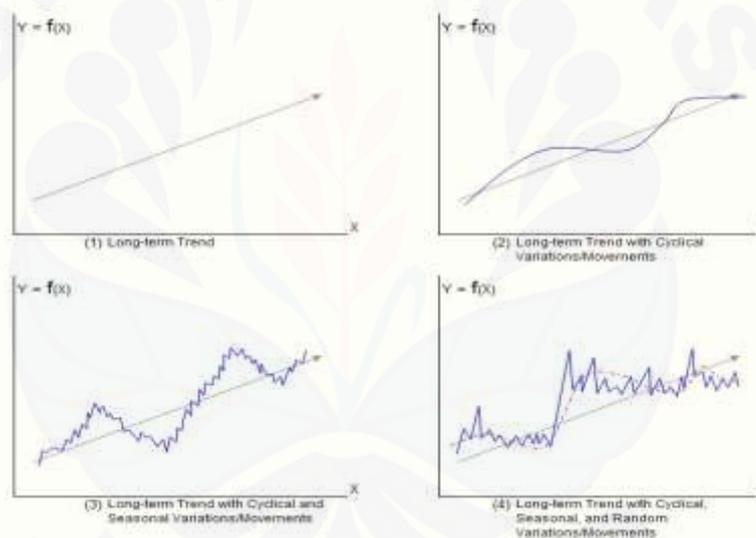
Data berkala atau sering kita sebut *time series*, ialah suatu data dari periode tertentu mengenai perkembangan dari suatu kejadian. Analisa data berkala dapat membuat kita mengetahui beberapa perkembangan kejadian dan juga pengaruh serta hubungannya dengan kejadian yang lainnya.

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.

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. Pola variasi *trend* permintaan daya listrik yang telah dijelaskan sebelumnya dapat diamati pada gambar 2.3 seperti berikut.



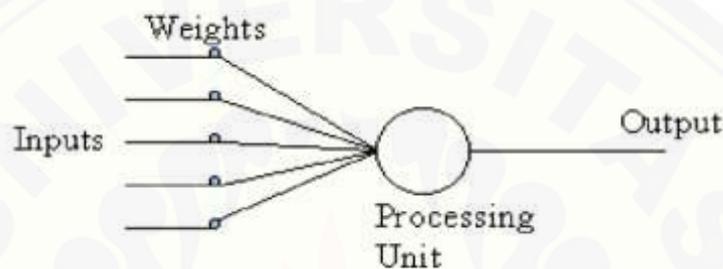
Gambar 2.3 Pola Variasi *Trend* Permintaan Daya Listrik (Ullah, 2014)

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*.

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



Gambar 2.4 Bentuk Dasar *Neuron* (Noviana, 2008)

Keterangan untuk gambar 2.4 di atas yaitu sebagai berikut:

1. *Input* merupakan masukan yang digunakan baik saat mempelajari mengenali suatu pola atau objek ataupun saat melakukan pelatihan.
2. *Weight* merupakan beban yang keadaanya selalu berubah setiap kali diberikan *input* sebagai proses pelatihan.
3. *Processing Unit* merupakan suatu tempat untuk mengenali suatu aktivitas atau objek yang mengalami perubahan dengan menggunakan pembebanan yang ditentukan.
4. *Output* merupakan suatu keluaran dari proses pelatihan atau pembelajaran terhadap suatu objek..

Metode neural network merupakan salah satu metode yang terbaik dalam permalan beban. Dibandingkan dengan metode yang lain, neural network memiliki nilai eror peramalan yang lebih kecil. Seperti pada penelitian Peramalan Beban puncak Listrik Jangka Pendek Menggunakan Metode Jaringan Syaraf tiruan (Triwulan, 2013). Pada penelitian tersebut membandingkan hasil peramalan beban

menggunakan JST dan koefisien beban, hasilnya lebih baik menggunakan JST karena eror persen yang di hasilkan JST rata-rata sebesar 0,12 % dengan akurasi 99,88%. Sedangkan eror persen yang dihasilkan oleh metode koefisien beban sebesar 1,85% dengan akurasi sebesar 98,15 %. Pada penelitian lain yaitu Peramalan Beban Jangka Panjang Sistem Kelistrikan Kota Palu Menggunakan Metode Logika Fuzzy (Massarang, 2015), penelitian tersebut membandingkan hasil peramalan fuzzy dengan RUPTL PLN tahun 2015-2024. Hasil peramalan tersebut memiliki selisih yang cukup besar yaitu sebesar 9,8820% .

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 (2005), 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*

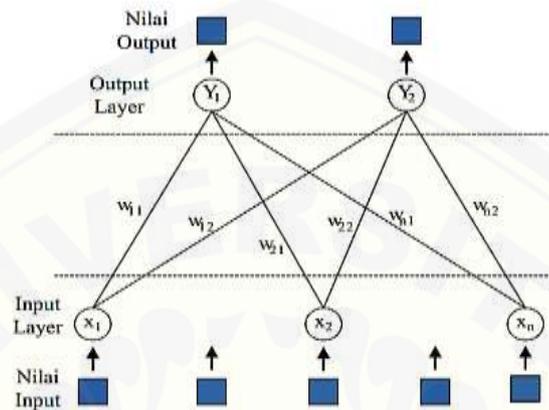
Berikut ini merupakan arsitektur suatu jaringan yang sering dipakai dalam menentukan konfigurasi *neural network*.

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

Suatu jaringan yang tersusun atas 1 layar *input* dan juga 1 layar *output* disebut juga dengan jaringan layer 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 layer tersembunyi.

JST layer 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 2.5 dapat diamati

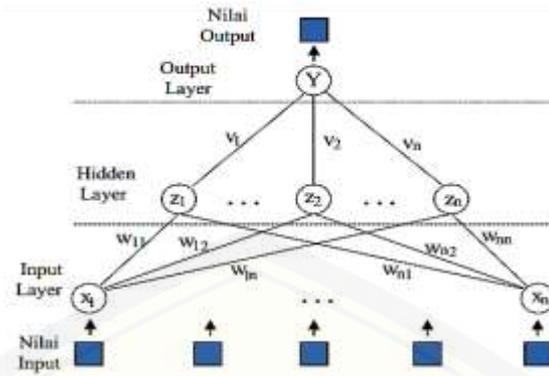
arsitektur *single layer network* yang memiliki n buah *input layer* ($X_1, X_2, X_3, \dots, X_n$) serta 2 buah *output layer* (Y_1, Y_2) dengan bobot masing-masing yang menghubungkan antara *input layer* dengan *output layer* ($W_{11}, W_{12}, W_{21}, W_{22}, W_{n1}, W_{n2}$).



Gambar 2.5 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 2.5 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 2.6 Multi Layer Network (Siregar, 2013)

2.7.2 Fungsi Transfer

Karakter dari *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 kategori jenisnya 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 mengerjakan 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.

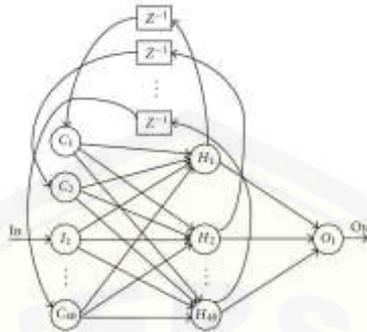
3. Peramalan

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 (Adianshar, 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 2.7 dibawah ini.



Gambar 2.7 Struktur dari *Recurrent Neural Network* (Jaroslaw S., 2011)

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*. (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 memperbaiki *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). *levenberg marquardt* merupakan salah satu dari beberapa metode yang digunakan dalam neural network dan yang paling sering digunakan dalam berbagai bidang aplikasi, seperti pengenalan pola, peramalan dan optimisasi. Hal ini dikarenakan metode ini menggunakan pembelajaran yang terbimbing. Pola masukan dan target diberikan sebagai pasangan data. Bobot-bobot awal dilatih dengan melalui tahap maju untuk mendapatkan error keluaran yang selanjutnya error ini digunakan dengan tahap mundur untuk memperoleh nilai bobot yang

sesuai agar dapat memperkecil nilai error sehingga target keluaran yang dikehendaki tercapai.

Tujuan dari model ini adalah untuk mendapatkan keseimbangan antara kemampuan jaringan mengenali pola yang digunakan selama proses pelatihan berlangsung serta kemampuan jaringan memberikan respon yang benar terhadap pola masukan yang berbeda dengan pola masukan selama pelatihan.

Persamaan matematika untuk menentukan bobot baru pada algoritma *levenberg marquardt* ialah sebagai berikut:

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

dimana: w_{new} : bobot baru

w_{old} : bobot lama

ϵ : vektor galat

Z : matriks hasil diferensiasi galat terhadap bobotnya.

Algoritme *levenberg marquardt* memiliki jumlah iterasi yang lebih baik dibandingkan dengan algoritme optimasi backpropagation yang lain. Namun, algoritme ini membutuhkan memori yang lebih banyak dibandingkan algoritme sejenisnya (Kurniawan, 2012).

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

1. Inisialisasi bobot dan bias dengan bilangan acak, *epoch* maksimum, dan minimal goal (performance yang dihitung dengan MSE).
2. Menentukan parameter yang dibutuhkan, antara lain:
 - a. Parameter *levenberg marquardt* yang nilainya harus lebih besar dari nol.
 - b. Parameter faktor Beta (β) yang digunakan sebagai parameter yang dikalikan atau dibagi dengan parameter *levenberg marquardt*.
3. Menghitung maju (*feedforward*) pada *hidden* dan output layer seperti langkah langkah pada algoritma *error back propagation*.
4. Menghitung nilai MSE.
5. Menghitung error dan total error jaringan. Rumus untuk error:

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

dimana: r : *input* ke- r

Rumus untuk menghitung total *error*:

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

dimana: e = vektor kesalahan berukuran Nx1 yang terdiri dari e_n r = 1, 2, 3, ..., n

6. Menghitung matriks Jacobian J(x)

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

$$J = \left[\frac{\partial er}{\partial w} \right] \dots\dots\dots(6.6)$$

$$\begin{bmatrix} \frac{\partial e_{11}(x)}{\partial x_1} & \frac{\partial e_{11}(x)}{\partial x_2} & \frac{\partial e_{11}(x)}{\partial x_1} & \dots & \frac{\partial e_{11}(x)}{\partial x_n} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{\partial e_{k1}(x)}{\partial x_1} & \frac{\partial e_{k2}(x)}{\partial x_2} & \frac{\partial e_{k3}(x)}{\partial x_2} & \dots & \frac{\partial e_{k1}(x)}{\partial x_n} \\ \frac{\partial e_{11}(x)}{\partial x_1} & \frac{\partial e_{11}(x)}{\partial x_2} & \frac{\partial e_{11}(x)}{\partial x_1} & \dots & \frac{\partial e_{11}(x)}{\partial x_n} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{\partial e_{k2}(x)}{\partial x_1} & \frac{\partial e_{k2}(x)}{\partial x_2} & \frac{\partial e_{k2}}{\partial x_3} & \dots & \frac{\partial e_{k2}}{\partial x_n} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{\partial e_{kp}(x)}{\partial x_1} & \frac{\partial e_{kp}}{\partial x_2} & \frac{\partial e_{kp}}{\partial x_3} & \dots & \frac{\partial e_{kp}}{\partial x_n} \end{bmatrix}$$

x = fungsi bobot-bobot jaringan dan bias dari keseluruhan jaringan.

$$X = [v_{11}, v_{12}, \dots, v_{ij}; v_{01}, v_{02}, \dots, v_{0j}; w_{11}, w_{12}, \dots, w_{jk}; w_{01}, w_{02}, \dots, w_{0k}] \dots(6.7)$$

7. sedangkan gradient dapat dihitung dengan

$$g = J^T J \dots\dots\dots(6.8)$$

8. Dalam hal ini **J** merupakan sebuah matrik jacobian yang berisikan turunan pertama dari *error* jaringan terhadap bobot dan bias jaringan. Perubahan pembobot dapat dihitung dengan :

$$\Delta x = [J(x)^T J(x) + \mu I]^{-1} * Gradient \dots\dots\dots(6.9)$$

9. Setelah didapatkan nilai Δx tahap selanjutnya adalah pengoreksian bobot dengan rumus:

$$X = X + \Delta X \dots\dots\dots(6.10)$$

10. Menghitung maju (*feedforward*) dengan bobot dan bias yang baru.
11. Menghitung MSE jaringan dengan bobot dan bias yang baru. Kemudian tes kondisi berhenti. Jika *epoch* atau iterasi masih berlanjut maka akan terdapat 2 kemungkinan kondisi berikut :
- Jika MSE naik $\mu \times \beta$
 - Jika MSE turun $\frac{\mu}{\beta}$

Keterangn :

- μ adalah konstanta *learning* bertujuan untuk melihat pengaryh perubahan terhadap proses *learning*. Dari sini dapat disimpulkan perubahan μ dapat memperlambat iterasi, perubahannya itu juga tergantung pada nilai awal saat random bobot dan bias
- β adalah faktor pengali di sini terjadi proses iterasi apabila semakin cepat proses iterasi hal ini disebabkan oleh nilai μ semakin kecil karena dibagi oleh nilai β yang semakin besar sehingga proses invers matrik semakin cepat.

Dari hasil pengujian dapat diambil kesimpulan bahwa semakin kecil konstanta learning μ jumlah iterasi semakin kecil dan semakin besar nilai faktor pengali β dapat mempercepat iterasi.

12. Kemudian melakukan kembali langkah 5 sampai langkah 8.

BAB 3. METODE PENELITIAN

3.1 Prosedur Penelitian

Pada metodologi penelitian ini, langkah-langkah yang dilakukan dalam pelaksanaan penelitian peramalan beban puncak jangka panjang pada GI Jember dengan menggunakan metode *recurrent neural network 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 studi peramalan beban ditinjau dari transformator pada GI Jember 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 GI Jember. Data yang akan digunakan untuk peramalan beban ialah seperti di bawah ini yaitu:

- a. Karakteristik dan spesifikasi transformator pada GI Jember
- b. Data beban puncak yang ada di GI Jember tersebut merupakan data beban historis yang akan digunakan pada penelitian ini dari tahun 2011 hingga tahun 2017.

3. Pembuatan Sistem Peramalan *Neural Network* dengan berbasis *Levenberg Marquardt*

Tahap ketiga adalah membuat sistem peramalan beban yang akan menggunakan *neural network 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 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 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 adalah:

- 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*).

5. Analisis

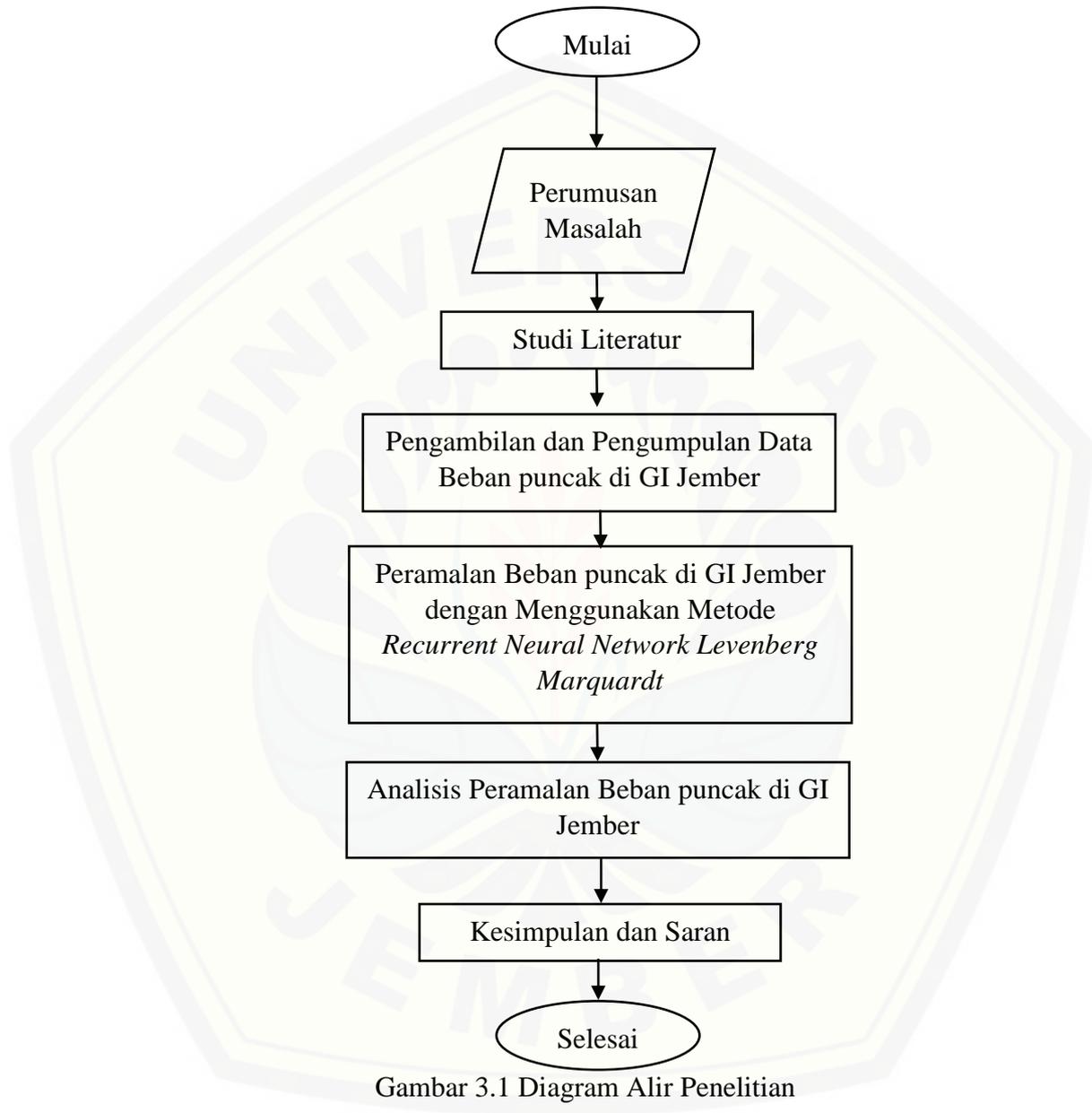
Pada tahap keenam ini yaitu melakukan analisis terhadap data yang diperoleh dari data yang sebelumnya. Dalam penelitian ini dilakukan peramalan tentang pertumbuhan dan penambahan beban puncak di GI Jember. Hasil data yang diperoleh yaitu berupa data peramalan beban puncak.

6. 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 dimaksud untuk pertimbangan atas pengembangan selanjutnya. Pengambilan kesimpulan dan saran dilakukan setelah menganalisa data yang telah didapat.

3.2 Diagram Alir Penelitian

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



Gambar 3.1 Diagram Alir Penelitian

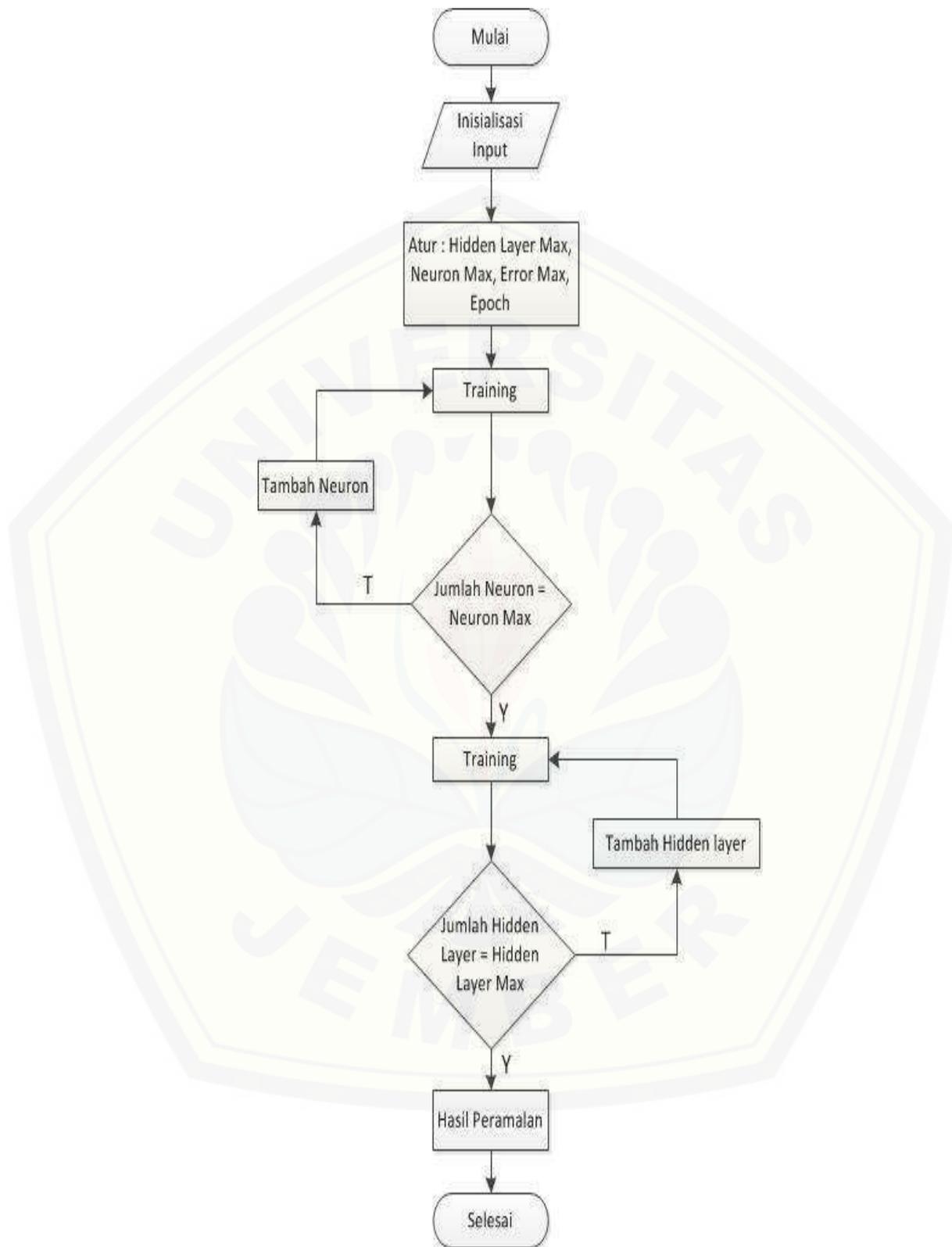
Dari diagram alir penelitian pada gambar 3.1 di atas dapat diamati bahwa proses diagram alir penelitian dimulai yaitu dari perumusan masalah yaitu, sampaikan transformator mencapai batas maksimal transmisi energi listrik pada Rayon Jember dan bagaimana sistem peramalan metode *recurrent neural network levenberg marquardt*. Adapun yang literatur berupa jurnal penelitian yang terkait sudah dilakukan oleh peneliti, buku panduan milik PLN, buku literatur tentang pentransmisi energi listrik, konduktor, maupun transformator dan buku standar PLN (SPLN) yang dikeluarkan secara resmi oleh pihak PLN.

Tahap selanjutnya yaitu melakukan pengambilan dan pengumpulan data. Adapun data yang digunakan yaitu berupa data sekunder milik PT. PLN (Persero) APJ Jember GI Jember. Data yang diambil ada dua macam yaitu data mengenai pembebanan listrik GI Jember yang berupa data beban puncak yang terjadi mulai dari data tahun 2011 hingga tahun 2017. Data yang kedua berupa data yang mengenai spesifikasi dan karakteristik transformator, data yang kedua ini diperoleh dari GI Jember.

Setelah itu dilakukan peramalan beban puncak dengan metode *recurrent neural network levenberg marquardt* dengan menggunakan *software* Matlab. Setelah mendapatkan hasil peramalan, selanjutnya yaitu melakukan analisis perhitungan dan peramalan beban puncak. Data hasil peramalan dan perhitungan akan dianalisis untuk sampai kapan transformator yang ada pada GI Jember akan mencapai batas maksimal pentransmisi energi listrik. Dan yang terakhir adalah 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 ke depan, dilakukan suatu metode peramalan beban puncak, metode peramalan yang akan digunakan pada penelitian kali ini yaitu dengan menggunakan metode *recurrent neural network levenberg marquardt*. Di bawah ini disajikan gambar diagram alir dari peramalan beban dengan menggunakan *software* Matlab.



Gambar 3.2 Diagram Alir Peramalan Beban

Proses dari diagram alir pada gambar 3.2 diatas dimulai dengan inialisasi input yang berupa data beban puncak di Transformator GI Jember dan memasukkannya dalam serangkaian program. Input data yang berupa beban puncak dalam satuan Amp yang jumlahnya cukup besar akan di normalisasikan sehingga dapat diproses oleh software matlab. Selanjutnya memasukkan jumlah maksimal hidden layer yaitu dua, dan memasukkan jumlah neuron maskimal tiap layer sejumlah 10 buah, serta memasukkan nilai epoch. Setelah parameter yang diperlukan sudah di masukkan, selanjutnya yaitu tahap training atau tahap pelatihan jaringan. Pada tahap training ini data akan dilatih untuk mengenali target data yang diinginkan dan untuk meminimalisir nilai eror. Adapun proses dari pelatihan ini jaringan akan berjalan atau running dengan menggunakan satu hidden layer dan satu neuron pada awalnya. Kemudian jumlah neuron akan bertambah satu secara otomatis apabila proses training pertama sudah selesai, sehingga jumlah neuron pada pelatihan kedua sejumlah 2 buah. Begitupun seterusnya hingga jumlah neuron mencapai batas maksimal yang sudah ditentukan yaitu 10 buah. Apabila jumlah neuron sudah mencapi 10 buah maka secara otomatis jumlah hidden layer pada pelatihan berikutnya akan bertambah satu. Begitu seterusnya hingga tercapai jumlah maksimal hidden layer yang telah ditentukan yaitu 2 buah. Apabila jumlah hidden layer pada jaringan sudah berjumlah 2 buah dan nilai neuronnya sejumlah 10 maka proses training menggunakan metode *Levenberq Marquardt* sudah selesai.

3.4 Peramalan Beban Menggunakan *Software* Matlab

Matlab ialah singkatan 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 puncak di GI Jember dan *output* yaitu data peramalan beban puncak pada transformator GI Jember. 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 keakuratan 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 lapis *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* (bukan hanya pada range 0-1 atau pada range -1-1 (Siang, 2009). Fungsi parameter yang digunakan pada sistem peralaman 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 pelatihan pada penelitian ini menggunakan *trainlm* (*train levenberg Marquadt*). Merupakan fungsi pelatihan jaringan yang memperbarui nilai bobot dan bias sesuai dengan pengoptimalan *levenberg-Marquadt*. Sistem *training* ini merupakan algoritma *backpropagation* tercepat di toolbox, dan sangat disarankan sebagai algoritma supervisi pilihan pertama, meskipun memerlukan lebih banyak memori daripada algoritma lainnya. Berikut ini merupakan perintah yang digunakan untuk melakukan *training* :

```
[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* pada perintah tersebut 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.



Gambar 3.3 Diagram Alir Perhitungan

Proses dari diagram alir perhitungan diatas dimulai saat diperoleh nilai peramalan beban untuk satu tahu kedepan. Nilai tersebut berupa nilai daya dalam satuan Amp yang dikurangi oleh nilai daya yang dihasilkan oleh Trasformator GI Jember. Setelah didapat nilai arus maka akan dibandingkan dengan nilai dari kapasitas hantar maksimum dari Trasformator Gardu Induk di Jember. Apabila nilai dari arus yang didapatkan belum mencapai atau melebihi dari nilai kuat hantar arus maksimum dari ketentuan RUPTL Tahun 2016-2025, maka akan diulangi proses perhitungannya dari hasil peramalan satu tahun kedepannya lagi. Jika nilai arus sudah mencapai atau melebihi dari kapasitas hantar maksimum kabel maka perhitungan sudah selesai.

BAB 5. PENUTUP

5.1 Kesimpulan

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

1. Hasil Simulasi peramala beban puncak pada transformator Gardu Induk Jember menggunakan JST *Levenberg Marquardt* pada pengujian bulan Januari 2016 samapi Desember 2017 memiliki nilai MAPE (*Mean Average Prediction Error*) bernilai 16,11166595%.
2. Konfigurasi untuk membangun jaringan pelatihan *recurrent neural network* terbaik yang digunakan sebagai jaringan peramalan beban ditahun berikutnya menggunakan *hidden layer* ke 1 dengan neuron sebesar 18 dan *error* sebesar 2,0535E-31.
3. Perkiraan beban puncak diperkirakan baru melewati kapasitas transformator pada Gardu Induk Jember pada tahun 2023 beban puncak yaitu pada bulan Maret sebesar 5031,254Amp, Juni sebesar 4657,63Amp, dan September sebesar 4162,68Amp yaitu melewati maksimal 4041,171Amp dari 70% dari ketentuan RUPTL 2016-2025 kapasita trafo yaitu 5773,67Amp.

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. Untuk memperoleh pertimbangan yang lebih banyak tentang peramalan beban, dapat menggunakan pola pelatihan, parameter, dan dapat menggabungkan dua metode yang berbeda.
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.
- Baskoro, D. 2017. Peramalan Jangka Pendek Beban Tenaga Listrik dan Konsumsi Bahan Bakar Gas pada PT.PJB UP Gresik Menggunakan Metode *Constructive Backpropagation Neural Network* Serta Prediksi Kebutuhan Bahan Bakar Gas Pembangkit. Jember: Program Studi Teknik Elektro Fakultas Teknik Universitas Jember.
- Defit, S. 2013. Perkiraan Beban puncak Jangka Pendek dengan Metode *Adaptive Neuro Fuzzy Inference System*. *SAINTIKOM*. 12(3): 165-176.
- Dwisatya, R. 2015. Prediksi Beban puncak 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 puncak Jangka Pendek Menggunakan Metode Jaringan Syaraf Tiruan. *Setrum*. 1(1): 35-40.
- Jaroslawa, S., K. Panczerz, 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 7 Juni 2018].
- 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 Djiteng. 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.

- Menteri Energi dan Sumber Daya Mineral. 2016 *Rencana Usaha Penyediaan Tenaga Listrik: PT Perusahaan Listrik Negara (persero) Tahun 2016 s.d 2025*. No. 5899 K/20/MEM/2016. Jakarta : KeMenterian Energi dan Sumber Daya Mineral.
- Mulyadi, A. (2017). *Desain Sistem Transfer Beban Otomatis dari Sumber PLN ke PLTS pada Waktu Beban puncak (WBP)*. 2(4): 73-77.
- 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 puncak 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.
- Rahman Arfiansyah, Ade Gafar Abdullah, Dadang Lukman Hakim. 2012. *Prakiraan Beban puncak Pada kelistrikan Indonesia Menggunakan Algoritma Adaptive Neuro Fuzzy Inference System*. *ELECTRANS*. 11(2): 18-26.
- SatuEnergi. 2015. *Dasar Listrik*. <http://www.satuenergi.com/2015/12/bagaimana-listrik-dihantarkan.html>. [Diakses pada 7 Juni 2018].
- Siang, J. J. 2009. *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*. Yogyakarta: Andi.
- 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. 2014. *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.
- 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 Oktober 2018].
- Za'im Mukhammad. 2014. Analisis Transformator Daya 3 Fasa 150 KV/ 20 KV pada Gardu Induk Ungaran PLN Distribusi Semarang. *EDUEL* 3(2): 9-16.

LAMPIRAN

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

```
clear;
clc;
load data1.mat;

%% Preprocessing
P=data1(:,1:36)';
T=data1(:,37:48)';

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

%% Define Parameters
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
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
disp([' Hidden Layer 1, 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
layer1=11;
break;
end
end

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

%parameter
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
disp([' Hidden Layer II, Jumlah Neuron = ' num2str(l2)]);

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

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

%Hitung mse
error=mse(Rec-tn);
%error(l2)=error; %cek
%Ambil Network yg terbaik
if modmse > error;
layerRec=l2;
modmse=error;
hl=2;
save training.mat;
end
else
layer2=12;
break;
end
end
```

```

end

%% Mengambil network yg terbaik

load training.mat;
an= sim(netRec,pn)
bs= [layerRec modmse]
a = poststd(an,meant,stdt);
disp('Network terbaik dengan error terkecil');
disp(['Hidden Layer ke ' num2str(hl), ', Neuron = '
num2str(layerRec), ', Error = ' num2str(modmse)]);

%Regression Diagram
plotregression(a,T);

%Pengujian

an=trastd(ujil,meanp,stdp);
cn1=sim(net,an);
c1=poststd(cn1,meant,stdt);

```

B. Listing Program Graphical User Interface (GUI) MATLAB

```

%%GUIskripsi

function varargout = GUIskripsi(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  '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 nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});

```

```
end
% End initialization code - DO NOT EDIT

% --- Executes just before GUIskripsi is made visible.
function GUIskripsi_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)
% varargin   command line arguments to GUIskripsi (see
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={3779 3516 3756 3879 3770 3700 3607 3828 4527 3828 3895
4579};
B={3883 3665 3725 3590 3873 3647 3415 3745 3834 3536 3825
4175};
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={2848.214044 1921.547432 3971.51332 4252.831019 4062.525798
4205.532216 4082.004976 3491.507618 3799.86596 2921.162887
3620.623712 3647.936006};
```

```

L={2784.828096 1737.718656 3835.946793 4183.979743 4140.850498
    3834.028868 4139.67959 3446.869806 4415.002493 3331.476557
    3157.849781 3723.341109};
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={24.63048308 45.34848032 5.737841317 9.637303906 7.759304978
13.66303286 13.16897632 8.790292104 16.06216126 23.68957975
7.044320612 20.33334776};
F={28.28153242 52.58612125 2.978437406 16.54539673 6.915840378
5.128293615 21.2204858 7.960752836 15.15395129 5.784034035
17.44183578 10.81817703};
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={16,11166595};
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('datauji', '-mat');
plot(myData.DataPLOTGUIPerb);
title('Grafik Pengujian Beban');
xlabel('Tahun 2016-2017, Bulan ke-');
ylabel('Beban (Amp)');
grid on

set(handles.text4,'enable','on')

% --- 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
% End initialization code - DO NOT EDIT

% --- Executes just before GUIskripsitab2 is made visible.
function GUIskripsitab2_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)
% varargin   command line arguments to GUIskripsitab2 (see
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={3461.505636 2070.611527 3840.36864 3771.162275 3937.573522
3681.734201 3886.340565 3415.970492 4235.596791 3039.54351
3247.325566 4035.425069};

I={2835.530706 2070.070429 4116.198127 3402.601783 3924.505127
3987.427264 3697.75324 3218.753229 4675.383248 3380.325807
3163.512316 3672.940281};

J={2969.331523 2013.188174 3794.486568 3857.079167 3964.216013
3800.892473 3514.680772 3735.179618 3883.820321 3004.154587
3098.293156 3952.181437};

K={2284.359183 2071.883334 4686.63691 3480.318469 3623.87325
4466.862559 3487.539683 3788.891862 4633.193453 3538.471465
3409.526207 3206.142684};

L={2707.19129 2580.996316 3782.964877 3727.825296 3878.554013
3803.68821 3313.134507 4129.567362 3772.314853 2715.961762
3039.968354 4154.614796};

M={2687.752014 2373.121143 5031.254563 2686.415472 3605.340216
4657.627005 3455.935551 3921.16079 4162.678523 3529.125398
3562.579785 3642.39898};

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('dataramalbeban', '-mat');
plot(myData1.DataPLOTGUI);
```

```

title('Grafik Peramalan Beban');
xlabel('Tahun 2018-2025, Bulan ke-');
ylabel('Beban (Amp)');
grid on

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

%gamebebanpuncak

function varargout = gamebebanpuncak(varargin)

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @gamebebanpuncak_OpeningFcn, ...
                  'gui_OutputFcn',   @gamebebanpuncak_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

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

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

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

```

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

% --- Outputs from this function are returned to the command
line.
function varargout = gamebebanpuncak_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)

I={2638.285225 3014.086512 5306.890275 2337.815008 3452.897424
4418.913291 3677.240819 3546.23767 4507.831654 3210.489334
3582.970486 4142.95473};

Dataku4=[I'];
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('datagame3', '-mat');
plot(myData1.datagame3);
title('Grafik Peramalan Beban');
xlabel('Tahun 2019');
ylabel('Beban (Amp)');
grid on

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

% --- Executes on button press in checkbox1.
function checkbox1_Callback(hObject, eventdata, handles)
% hObject handle to checkbox1 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see
GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox1
```

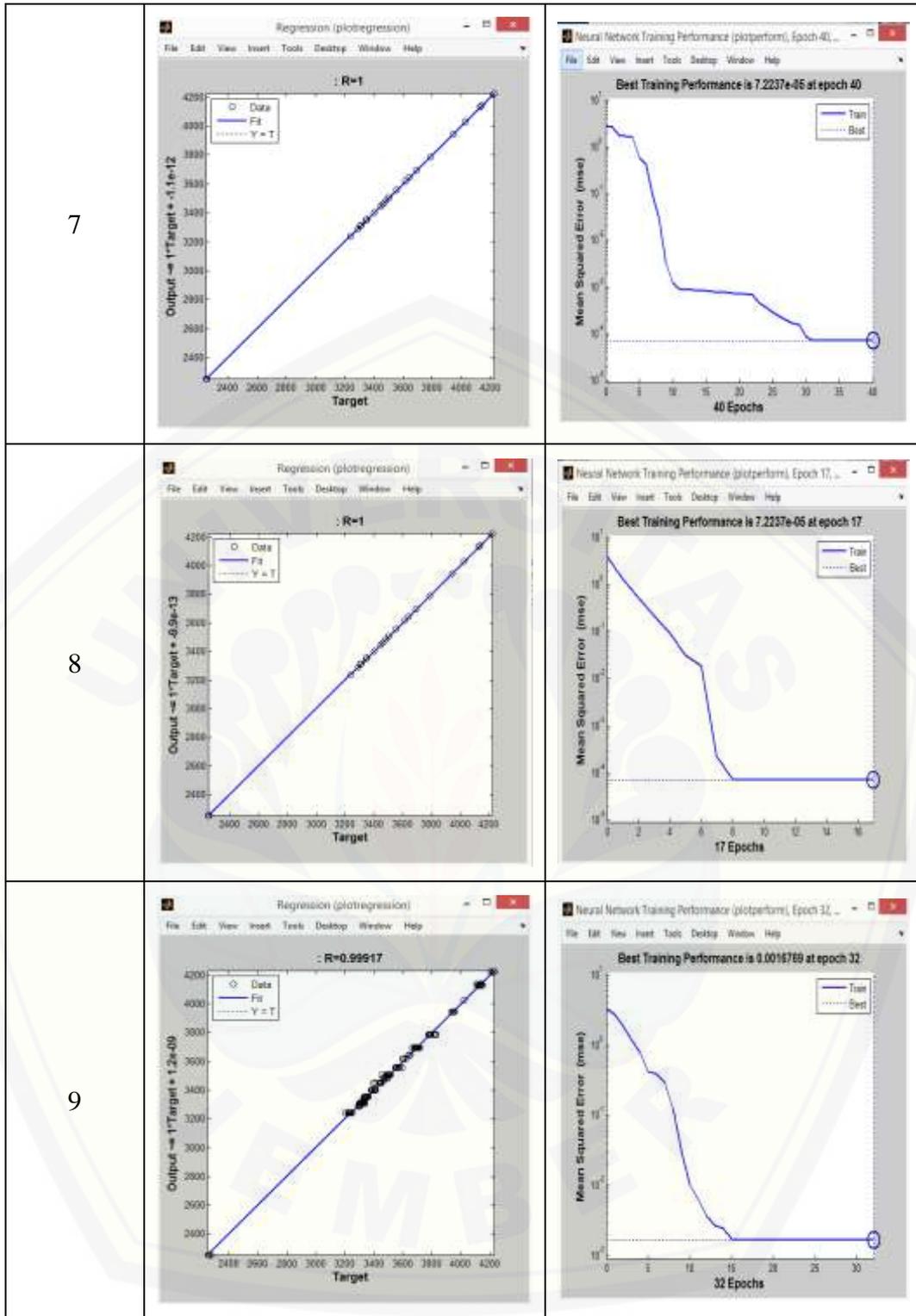
B. Tabel Data Beban puncak Trasformator Gardu Induk Jember

No	Bulan	Beban (Amp)						
		2011	2012	2013	2014	2015	2016	2017
1	Januari	2558	2752	3047	3464	3314	3779	3883
2	Februari	2860	2834	3226	3290	3308	3516	3665
3	Maret	3304	3501	3312	3291	3789	3756	3725
4	April	2646	2723	3386	2254	3557	3879	3590
5	Mei	2611	3454	3083	3945	4130	3770	3873
6	Juni	2928	2776	2988	3485	3348	3700	3647
7	Juli	3340	2856	3941	3402	3355	3607	3415
8	Agustus	2664	3187	3103	3240	3619	3823	3745
9	September	3552	3036	3297	3449	4140	4527	3834
10	Oktober	2845	3353	3594	3507	4223	3828	3536
11	November	2971	3036	3107	3694	3642	3895	3825
12	Desember	2991	3407	3661	3694	4028	4579	4175

C. Gambar Hasil Percobaan *Training Recurent Neural Network*

Training	Regression	Performance
1	<p>Regression (plotregression) : R=1 Output = 1*Target + -9.4e+23</p>	<p>Neural Network Training Performance (plotperform), Epoch 18... Best Training Performance is 7.2227e-05 at epoch 18</p>
2	<p>Regression (plotregression) : R=1 Output = 1*Target + -1.2e+12</p>	<p>Neural Network Training Performance (plotperform), Epoch 9, ML... Best Training Performance is 5.6934e-31 at epoch 9</p>
3	<p>Regression (plotregression) : R=1 Output = 1*Target + -1.4e+12</p>	<p>Neural Network Training Performance (plotperform), Epoch 9, ML... Best Training Performance is 5.7126e-31 at epoch 9</p>

<p>4</p>		
<p>5</p>		
<p>6</p>		



<p>10</p>		
<p>11</p>		
<p>12</p>		

13

