



**PERAMALAN JANGKA PENDEK BEBAN TENAGA LISTRIK PADA
PT. PJB UP GRESIK MENGGUNAKAN METODE *CONSTRUCTIVE
BACKPROPAGATION NEURAL NETWORK* SERTA PREDIKSI
KEBUTUHAN BAHAN BAKAR GAS PEMBANGKITAN**

SKRIPSI

oleh

**Dovy Risko Baskoro
NIM 121910201090**

**JURUSAN TEKNIK ELEKTRO STRATA 1
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2017**



**PERAMALAN JANGKA PENDEK BEBAN TENAGA LISTRIK PADA
PT. PJB UP GRESIK MENGGUNAKAN METODE *CONSTRUCTIVE
BACKPROPAGATION NEURAL NETWORK* SERTA PREDIKSI
KEBUTUHAN BAHAN BAKAR GAS PEMBANGKITAN**

SKRIPSI

diajukan guna memenuhi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Teknik Elektro (S1)
dan mencapai gelar Sarjana Teknik

oleh

**Dovy Risko Baskoro
NIM 121910201090**

**JURUSAN TEKNIK ELEKTRO STRATA 1
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2017**

PERSEMBAHAN

Dengan penuh rasa syukur saya ucapkan kepada Allah SWT atas segala karunia-Nya sehingga saya dapat menyelesaikan skripsi ini. Skripsi ini merupakan pembelajaran tentang bagaimana menghargai dan menjalani sebuah proses. Apapun itu hasilnya, proses pantas diberi penghargaan yang lebih daripada hasil, namun hasil adalah bukti, dan bukti adalah segalanya. Dengan penuh rasa syukur dan dengan ketulusan hati saya buktikan dan persembahkan hasil proses saya selama perkuliahan ini kepada:

1. Kedua orang tua saya Bapak Jumat dan Alm. Ibu Niwati yang tercinta. terima kasih yang sangat berlimpah karena telah mendidik, membesarkan, memberi cinta dan kasih sayang serta doa yang tiada putus hingga saya dapat menyelesaikan skripsi ini, dan saya persembahkan gelar yang saya dapatkan untuk mereka.
2. Dosen Pembimbing Utama Bapak Bambang Sri Kaloko, serta Bapak Dosen Pembimbing Anggota Bapak Dedy Kurnia Setiawan atas kesabaran dan keikhlasan dalam membimbing saya menyelesaikan skripsi ini.
3. Seluruh dosen Teknik Elektro Universitas Jember yang telah memberikan bekal ilmu pengetahuan dan bimbingan selama mengikuti pendidikan di Jurusan Teknik Elektro Fakultas Teknik Universitas Jember
4. Saudara-saudara ku di Teknik Elektro Universitas Jember.
5. Almamater Universitas Jember.

MOTO

“Sometimes you win, some times you learn. Trust the process”

Tujuan pendidikan itu untuk mempertajam kecerdasan, memperkuat kemauan
serta memperhalus perasaan.

(Tan Malaka)

Barang siapa bersungguh-sungguh, sesungguhnya kesungguhannya itu
adalah untuk dirinya sendiri.

(QS Al-Ankabut [29]: 6)

Terbentur, terbentur, terbentur, ... , terbentuk.

(Tan Malaka)

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Dovy Risiko Baskoro

NIM : 121910201090

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Peramalan Jangka Pendek Beban Tenaga Listrik Pada PT. PJB UP Gresik Menggunakan Metode *Constructive Backpropagation Neural Network* Serta Prediksi Kebutuhan Bahan Bakar Gas Pembangkitan” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun, 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 ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 20 Juni 2017

Yang menyatakan,

Dovy Risiko Baskoro

NIM 121910201090

SKRIPSI

**PERAMALAN JANGKA PENDEK BEBAN TENAGA LISTRIK PADA
PT. PJB UP GRESIK MENGGUNAKAN METODE *CONSTRUCTIVE
BACKPROPAGATION NEURAL NETWORK* SERTA PREDIKSI
KEBUTUHAN BAHAN BAKAR GAS PEMBANGKITAN**

Oleh

Dovy Risiko Baskoro
NIM 121910201090

Pembimbing

Dosen Pembimbing Utama : Dr. Bambang Sri Kaloko, S.T., M.T.
Dosen Pembimbing Anggota : Dedy Kurnia Setiawan, S.T., M.T.

PENGESAHAN

Skripsi berjudul “Peramalan Jangka Pendek Beban Tenaga Listrik Pada PT. PJB UP Gresik Menggunakan Metode *Constructive Backpropagation Neural Network* Serta Prediksi Kebutuhan Bahan Bakar Gas Pembangkitan” telah diuji dan disahkan pada:

Hari, tanggal : Selasa, 20 Juni 2017

Tempat : Fakultas Teknik Universitas Jember.

Tim Penguji:

Ketua,

Anggota I,

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

Dedy Kurnia Setiawan, S.T., M.T.
NIP 198006102005011003

Anggota II,

Anggota III,

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

Dr. Azmi Saleh, S.T., M.T.
NIP 197106141997021001

Mengesahkan
Dekan Fakultas Teknik,

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

Peramalan Jangka Pendek Beban Tenaga Listrik Pada PT. PJB UP Gresik Menggunakan Metode *Constructive Backpropagation Neural Network* Serta Prediksi Kebutuhan Bahan Bakar Gas Pembangkitan (Short Term Forecasting Power Supply At PT. PJB UP Gresik Using Constructive Backpropagation Neural Network Method And Prediction of Fuel Generation Fuel Needs).

Dovy Risiko Baskoro

*Jurusan Teknik Elektro, Fakultas Teknik
Universitas Jember
E-mail: bazkoro158@gmail.com*

ABSTRAK

Aktivitas manusia dalam penggunaan listrik dari waktu ke waktu akan terus mengalami peningkatan. Hal ini diakibatkan karena listrik sudah menjadi bagian penting bagi kemajuan peradaban manusia di berbagai bidang, baik bidang ekonomi, teknologi, sosial dan budaya manusia. Peramalan beban listrik dibutuhkan karena tenaga listrik tidak dapat disimpan dalam skala besar, karenanya tenaga ini harus disediakan pada saat dibutuhkan. Akibatnya muncul persoalan dalam menghadapi kebutuhan daya listrik yang tidak tetap dari waktu ke waktu. Berdasarkan uraian tersebut maka dalam penelitian ini akan dirancang sebuah penelitian yang berjudul “Peramalan Jangka Pendek Beban Tenaga Listrik Pada PT. PJB UP Gresik Menggunakan Metode *Constructive Backpropagation Neural Network* Serta Prediksi Kebutuhan Bahan Bakar Gas Pembangkitan”. Penggunaan metode *Constructive Backpropagation Neural Network* pada penelitian ini diharapkan mampu meningkatkan akurasi dari sistem peramalan beban tenaga listrik. Hasil yang didapatkan memiliki rata-rata error 1,7% untuk peramalan beban dan 0,6% untuk peramalan bahan bakar.

Kata Kunci : peramalan beban, *constructive backpropagation neural network*, *quadratic least square*

Peramalan Jangka Pendek Beban Tenaga Listrik Pada PT. PJB UP Gresik Menggunakan Metode *Constructive Backpropagation Neural Network* Serta Prediksi Kebutuhan Bahan Bakar Gas Pembangkitan (Short Term Forecasting Power Supply At PT. PJB UP Gresik Using Constructive Backpropagation Neural Network Method And Prediction of Fuel Generation Fuel Needs).

Dovy Risiko Baskoro

*Jurusan Teknik Elektro, Fakultas Teknik
Universitas Jember
E-mail: bazkoro158@gmail.com*

ABSTRACT

Human activity in the use of electricity over time will continue to increase. This is due to electricity has become an important part for the progress of human civilization in various fields, both economic, technological, social and human culture. Power load forecasting is required because electrical power can not be stored on a large scale, therefore this power should be provided when needed. As a result, problems arise in the face of the need for electrical power is not fixed from time to time. Based on the description then in this study will be designed a study entitled "Forecasting Short Term Power Supply At PT. PJB UP Gresik Using Constructive Backpropagation Neural Network Method And Prediction of Fuel Generation Fuel Needs ". The use of Constructive Backpropagation Neural Network method in this research is expected to increase the accuracy of power load forecasting system. The results obtained have an average error of 1.7% for load forecasting and 0.6% for forecasting of fuel.

Keywords: load forecasting, constructive backpropagation neural network, quadratic least square

RINGKASAN

Peramalan Jangka Pendek Beban Tenaga Listrik Pada PT. PJB UP Gresik Menggunakan Metode *Constructive Backpropagation Neural Network* Serta Prediksi Kebutuhan Bahan Bakar Gas Pembangkitan; Dovy Risko Baskoro; 121910201090; 2017; 75 halaman; Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Aktivitas manusia dalam penggunaan listrik dari waktu ke waktu akan terus mengalami peningkatan. Hal ini diakibatkan karena listrik sudah menjadi bagian penting bagi kemajuan peradaban manusia di berbagai bidang, baik bidang ekonomi, teknologi, sosial dan budaya manusia. Peningkatan kebutuhan listrik tersebut mengharuskan pihak penyedia listrik dapat menyalurkan kebutuhan listrik konsumen agar stabilitas multibidang di masyarakat dapat terjamin. Adanya gangguan dalam pasokan listrik dapat mengganggu aktivitas manusia di berbagai bidang. Oleh karena itu kebutuhan reliabilitas antara pasokan listrik dengan kebutuhan listrik merupakan hal yang penting.

Peramalan beban listrik dibutuhkan karena tenaga listrik tidak dapat disimpan dalam skala besar, karenanya tenaga ini harus disediakan pada saat dibutuhkan. Akibatnya muncul persoalan dalam menghadapi kebutuhan daya listrik yang tidak tetap dari waktu ke waktu. Apabila daya yang dikirim dari bus-bus pembangkit jauh lebih besar daripada permintaan daya pada bus-bus beban, maka akan timbul persoalan pemborosan energi pada perusahaan listrik, terutama untuk pembangkit termal. Sedangkan apabila daya yang dibangkitkan dan dikirimkan lebih rendah atau tidak memenuhi kebutuhan beban konsumen maka akan terjadi pemadaman lokal pada bus-bus beban, yang akibatnya merugikan pihak konsumen. Oleh karena itu diperlukan penyesuaian antara pembangkit dengan permintaan daya.

Berdasarkan uraian tersebut maka dalam penelitian ini akan dirancang sebuah penelitian yang berjudul “Peramalan Jangka Pendek Beban Tenaga Listrik Pada PT. PJB UP Gresik Menggunakan Metode *Constructive Backpropagation Neural*

Network Serta Prediksi Kebutuhan Bahan Bakar Gas Pembangkitan”. Penggunaan metode *Constructive Backpropagation Neural Network* pada penelitian ini diharapkan mampu meningkatkan akurasi dari sistem peramalan beban tenaga listrik. Sehingga dalam pelaksanaannya mampu jadi salah satu cara yang efektif dan cepat guna untuk peramalan beban, sehingga akurasi dan kualitas dalam sistem pembangkitan dan distribusi kelistrikan serta perencanaan dalam penggunaan gas sebagai bahan bakar menjadi lebih baik dan efisien.

Peramalan jangka pendek beban tenaga listrik menggunakan 48 input dengan metode *constructive backpropagation neural network* tahun 2016 mendapatkan hasil rata-rata *error* sebesar 2,07%. *Error* terkecil sebesar 0,2% dengan hasil peramalan 21024670 Watt pada bulan Oktober Minggu ketiga. *Error* terbesar dengan nilai 4,9% dengan hasil peramalan 18921563 Watt pada bulan September minggu pertama. Peramalan jangka pendek beban tenaga listrik menggunakan 24 input dengan metode *constructive backpropagation neural network* tahun 2016 mendapatkan hasil rata-rata *error* sebesar 1,7%. *Error* terkecil sebesar 0,03% dengan hasil peramalan 17862013.6 Watt pada bulan Agustus Minggu ke-4. *Error* terbesar dengan nilai 4,2% dengan hasil peramalan 18793678.9 Watt pada bulan September minggu pertama. Peramalan bahan bakar gas menggunakan metode *quadratic least square* mendapatkan hasil rata-rata *error* sebesar 0,6%. *Error* tekecil sebesar 0,6% dengan hasil *flowrate* 23,7 KNM³/H pada bulan Agustus minggu kedua. *Error* terbesar dengan nilai 4,4% dengan hasil *flowrate* 24,3 KNM³/H pada bulan November minggu pertama.

PRAKATA

Puji syukur ke hadirat Allah SWT. Atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi berjudul “Peramalan Jangka Pendek Beban Tenaga Listrik Pada PT. PJB UP Gresik Menggunakan Metode *Constructive Backpropagation Neural Network* Serta Prediksi Kebutuhan Bahan Bakar Gas Pembangkitan”. 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 beberapa pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Ibu Dr. Ir. Entin Hidayah, M.UM., selaku Dekan Fakultas Teknik Universitas Jember;
2. Bapak Dr. Bambang Sri Kaloko, S.T., M.T., selaku Ketua Jurusan Teknik Elektro Universitas Jember;
3. Bapak Dr. Bambang Sri Kaloko, S.T., M.T. selaku Dosen Pembimbing Utama, dan Bapak Dedy Kurnia Setiawan, ST., MT. selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi ini;
4. Bapak Dr. Triwahju Hardianto, S.T., M.T. selaku dosen penguji I;
5. Bapak Dr. Azmi Saleh, S.T., MT. selaku dosen penguji II;
6. Seluruh Dosen Teknik Elektro yang telah memberikan bekal ilmu pengetahuan dan bimbingan selama mengikuti pendidikan di Jurusan Teknik Elektro Fakultas Teknik Universitas Jember;
7. Para staf karyawan dan karyawan serta teknisi Fakultas Teknik Universitas Jember yang telah memberikan bantuan selama mengikuti pendidikan di Fakultas Teknik Universitas Jember;
8. Kedua orang tua saya Bapak Jumat dan Alm. Ibu Niwati tercinta, saudaraku Anton Wahyudi dan Andi Suryadi yang telah memberikan dorongan, semangat, kasih sayang, perhatian, kesabaran dan doa yang tak pernah putus

demi mempermudah saya mencapai kesuksesan awal dalam perguruan tinggi serta terselesaikannya skripsi ini;

9. Saudara-saudara ku Teknik Elektro 2012 dan juga warga HME lainnya yang sangat membantu dan telah memberi semangat, semoga kekompakan kita tetap terjaga dan bisa mencapai kesuksesan bersama;
10. Keluarga besar MAHADIPA yang telah memberikan banyak pembelajaran selama ini;

Penulis menyadari bahwa skripsi ini jauh dari kesempurnaan, untuk itu penulis menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi. Akhir kata, semoga skripsi ini dapat berguna bagi pembaca dan bagi penulis sendiri pada khususnya semoga Allah SWT memberikan yang terbaik untuk kita semua. Amin

Jember, 20 Juni 2017

Penulis

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Aktivitas manusia dalam penggunaan listrik dari waktu ke waktu akan terus mengalami peningkatan. Hal ini diakibatkan karena listrik sudah menjadi bagian penting bagi kemajuan peradaban manusia di berbagai bidang, baik bidang ekonomi, teknologi, sosial dan budaya manusia. Peningkatan kebutuhan listrik tersebut mengharuskan pihak penyedia listrik dapat menyalurkan kebutuhan listrik konsumen agar stabilitas multibidang di masyarakat dapat terjamin. Adanya gangguan dalam pasokan listrik dapat mengganggu aktivitas manusia di berbagai bidang. Oleh karena itu kebutuhan reliabilitas antara pasokan listrik dengan kebutuhan listrik merupakan hal yang penting (Khair, 2011).

Untuk memenuhi kebutuhan tersebut secara kuantitas dan kualitas maka dibutuhkan perencanaan sistem tenaga listrik yang tepat. Sebagai dasar dalam perencanaan, baik perencanaan operasi maupun perencanaan sistem pengembangan tenaga listrik, salah satu hal yang penting adalah peramalan yang tepat untuk mengetahui kebutuhan tenaga listrik dalam kurun waktu tertentu. Peramalan beban meliputi peramalan beban puncak dan peramalan kebutuhan energi listrik. Peramalan berdasarkan rentang waktu dapat dikategorikan menjadi peramalan jangka pendek, jangka menengah dan jangka panjang (Meindra, 2014).

Peramalan beban listrik dibutuhkan karena tenaga listrik tidak dapat disimpan dalam skala besar, karenanya tenaga ini harus disediakan pada saat dibutuhkan. Akibatnya muncul persoalan dalam menghadapi kebutuhan daya listrik yang tidak tetap dari waktu ke waktu. Apabila daya yang dikirim dari bus-bus pembangkit jauh lebih besar daripada permintaan daya pada bus-bus beban, maka akan timbul persoalan pemborosan energi pada perusahaan listrik, terutama untuk pembangkit termal. Sedangkan apabila daya yang dibangkitkan dan dikirimkan lebih rendah atau tidak memenuhi kebutuhan beban konsumen maka akan terjadi pemadaman lokal pada bus-bus beban, yang akibatnya merugikan pihak konsumen. Oleh

karena itu diperlukan penyesuaian antara pembangkit dengan permintaan daya (Mulyadi, 2010).

PLTU (Pembangkit Listrik Tenaga Uap) merupakan salah satu metode pembangkitan listrik yang ada di Indonesia. PLTU pada umumnya menggunakan batubara sebagai bahan bakar. Namun dalam perkembangannya, PLTU juga mulai beralih menggunakan gas sebagai bahan bakar, seperti yang ada pada PLTU milik PT PJB UP Gresik. Dalam penerapannya PLTU memiliki keunggulan dapat dioperasikan menggunakan berbagai jenis bahan bakar (padat, cair, dan gas), PLTU juga dapat dioperasikan dengan berbagai mode pembebanan. Selain itu PLTU juga memiliki kelemahan yang salah satunya adalah PLTU sangat tergantung pada tersedianya pasokan bahan bakar (Astawan, 2014).

Pembangkit listrik bertenaga uap yang menggunakan gas sebagai bahan bakar pembangkitan biasa disebut sebagai PLTGU (Pembangkit Listrik Tenaga Gas dan Uap). Penggunaan bahan bakar yang efektif dan efisien tentu saja disesuaikan dengan kebutuhan beban yang ada, sehingga tidak ada energi yang terbuang ataupun justru malah kekurangan bahan bakar karena perencanaan beban yang kurang baik. Oleh karena itu perlu dirancang sebuah metode yang baik dalam peramalan beban dan perencanaan penggunaan bahan bakar untuk pembangkitan.

Peramalan beban tenaga listrik telah diteliti oleh Ricky Ardian Pradana dalam jurnalnya pada tahun 2015. Dalam peramalannya menggunakan metode *backpropagation neural network*. Sedangkan penelitian yang dilakukan oleh Maju Binoto dalam jurnalnya tahun 2015, juga menggunakan metode *backpropagation neural network*, dalam kedua penelitian tersebut juga dilakukan peramalan terhadap energi listrik yang akan terjual dan daya yang tersambung. Namun pada penelitiannya tidak dilakukan peramalan terhadap bahan bakar yang akan digunakan untuk pembangkitan energi. Metode penelitian yang digunakan juga menggunakan metode *feedforward backpropagation* dimana dalam penentuan dan pembuatannya *hidden layer* masih dilakukan secara *manual* untuk mencapai target *error* yang telah ditentukan.

Berdasarkan uraian tersebut maka dalam penelitian ini akan dirancang sebuah penelitian yang berjudul “Peramalan Jangka Pendek Beban Tenaga Listrik dan

Konsumsi Bahan Bakar Gas Pada PT. PJB UP Gresik Menggunakan Metode *Constructive Backpropagation Neural Network*". Penggunaan metode *Constructive Backpropagation Neural Network* pada penelitian ini diharapkan mampu meningkatkan akurasi dari sistem peramalan beban tenaga listrik. Sehingga dalam pelaksanaannya mampu jadi salah satu cara yang efektif dan cepat guna untuk peramalan beban, sehingga akurasi dan kualitas dalam sistem pembangkitan dan distribusi kelistrikan serta perencanaan dalam penggunaan gas sebagai bahan bakar menjadi lebih baik dan efisien.

Pemilihan PLTGU pada PT PJB UP Gresik sebagai pusat penelitian karena peran dan fungsi PLTGU Gresik itu sendiri. Mengingat wilayah Gresik memiliki cukup banyak pabrik dan perusahaan skala besar ataupun menengah yang membutuhkan daya listrik secara kontinyu sehingga diperlukan perencanaan pembangkitan yang baik berdasarkan peramalan beban yang telah dilakukan. Karena jika besar konsumsi listrik tidak diperkirakan maka dapat mempengaruhi kesiapan dari unit pembangkit untuk menyediakan pasokan energi listrik kepada konsumen. Selain itu karena pada perkembangannya PLTU PT PJB UP Gresik sudah meninggalkan batubara dalam penggunaan bahan bakar dan menggunakan gas sebagai bahan bakar pembangkitan sehingga menjadi PLTGU.

Peramalan jumlah konsumsi bahan bakar gas pada penelitian ini dilakukan karena berkaitan dengan pola atau sistem pengadaan bahan bakar yang telah ada pada PLTGU di PT.PJB UP Gresik. Pola pengadaan atau pembelanjaan gas yang telah ada ialah menggunakan acuan beban maksimal yang pernah diminta oleh PLN dalam periode waktu tertentu. Jadi jumlah bahan bakar yang disiapkan untuk pembangkitan dialokasikan untuk mencukupi beban maksimal tersebut. Pola pembelanjaan seperti ini kurang efektif dari segi biaya berkaitan dengan perubahan harga bahan bakar gas dan biaya produksi atau pembangkitan. Tidak menutup kemungkinan anggaran harga gas ketika pengadaan jauh lebih mahal dibandingkan ketika bahan bakar gas sudah digunakan untuk pembangkitan.

Apabila hal tersebut terjadi tentunya akan meningkatkan biaya produksi dan dapat mengurangi keuntungan, karena harga penjualan listrik tetap dan tidak mengikuti peningkatan harga bahan bakar gas. Jadi langkah lebih baik jika pola

pengadaan bahan bakar gas untuk pembangkitan sudah direncanakan mengikuti peramalan beban yang akan dipenuhi sesuai kebutuhan agar anggaran pembelanjaan bahan bakar bisa seimbang dengan harga jual daya listrik yang dikirim.

1.2 Rumusan Masalah

Sesuai dengan latar belakang yang telah dipaparkan, maka dalam penelitian ini dapat dirumuskan beberapa masalah yaitu :

1. Bagaimana mendesain *Constructive Backpropagation Neural Network* untuk peramalan beban pada PT PJB UP Gresik?
2. Bagaimana perancangan sistem karakteristik *Input Output* untuk peramalan jumlah konsumsi bahan bakar gas pada PT PJB UP Gresik?
3. Bagaimana tingkat akurasi hasil peramalan beban dan jumlah bahan bakar gas jika dibandingkan dengan hasil data pada PT PJB UP Gresik?

1.3 Tujuan Penelitian

Tujuan dilakukannya penelitian ini adalah sebagai berikut :

1. Mendapatkan perancangan sistem peramalan jangka pendek beban dan jumlah konsumsi gas pada PT PJB UP Gresik.
2. Mengetahui tingkat akurasi hasil peramalan beban dan jumlah konsumsi gas pada PT PJB UP Gresik menggunakan metode *Constructive Backpropagation Neural Network* dan metode Karakteristik *Input Output* .

1.4 Manfaat Penelitian

Manfaat penelitian ini ialah agar peramalan jangka pendek beban dan jumlah konsumsi gas pada PT PJB UP Gresik menggunakan metode *Constructive Backpropagation Neural Network* dalam penerapannya dapat menjadi salah satu cara atau metode untuk membuat perencanaan pembangkitan listrik dan pengadaan bahan bakar gas pada PT PJB UP Gresik. Dengan adanya penelitian ini, juga diharapkan dapat menjadi kajian terhadap penelitian selanjutnya agar ditemukan teknologi atau metode yang tepat dalam peramalan beban dan konsumsi kebutuhan bahan bakar pembangkitan.

1.5 Batasan Masalah

Berdasarkan uraian rumusan masalah di atas, maka pembahasan penelitian ini dibatasi oleh:

1. Data beban sesungguhnya yang digunakan sebagai acuan hasil peramalan dibatasi data bulan Juli sampai November tahun 2016.
2. Peramalan beban dilakukan sebagai peramalan mingguan.
3. Simulasi peramalan beban dilakukan dengan menggunakan *software Matlab*

1.6 Sistematika Penulisan

Penulisan penelitian ini tersusun secara sistematis serta terdiri dari beberapa bagian yang saling berhubungan dengan harapan menjadi mudah dipahami dan diambil manfaatnya. Adapun uraian singkat mengenai sistematika penulisan penelitian ini ialah sebagai berikut.

BAB 1. PENDAHULUAN

Pada bab ini berisi tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan.

BAB 2. TINJAUAN PUSTAKA

Bab dua mencakup konsep serta teori-teori serta rujukan penelitian lain yang mendasari dalam pembuatan tugas akhir.

BAB 3. METODOLOGI PENELITIAN

Tempat dan waktu penelitian, alat dan bahan, tahap penelitian, dan desain penelitian adalah bagian-bagian yang menyusun bab tiga. Desain penelitian mencakup desain konfigurasi *neural network*, termasuk blok diagram sistem secara keseluruhan dan *flowchart*.

BAB 4. HASIL DAN PEMBAHASAN

Pada bab ini dibahas dan dianalisis hasil dari peramalan beban dan juga hasil dari peramalan bahan bakar.

BAB 5. KESIMPULAN DAN SARAN

Bab ini terdiri atas kesimpulan dan saran dari hasil pengujian serta analisis penelitian yang telah dilakukan.

BAB 2. TINJAUAN PUSTAKA

2.1 Pembangkit Termal

Pembangkit Termal merupakan pembangkit tenaga listrik yang melibatkan proses panas dalam pembangkitan tenaga listriknya. Tipe pembangkitan ini membutuhkan bahan bakar yang berasal dari bahan bakar fosil. Pembangkit Listrik Tenaga Gas (PLTG), Pembangkit Listrik Tenaga Uap (PLTU) dan Pembangkit Listrik Tenaga Diesel (PLTD) merupakan beberapa contoh dari pembangkit-pembangkit yang merupakan pembangkit termal.

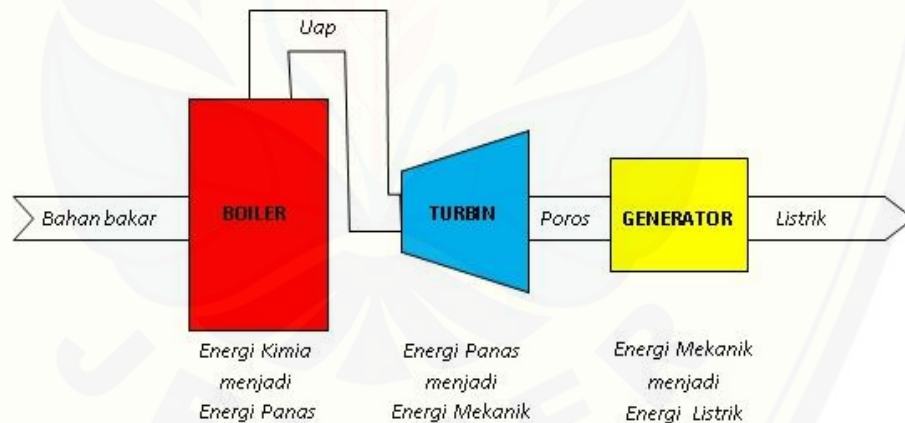
Salah satu pembangkit termal yang digunakan di Indonesia adalah PLTU yang dioperasikan oleh PJB UP Gresik. Terdapat empat buah PLTU yang dioperasikan oleh PJB UP Gresik saat ini yang digunakan untuk memenuhi kebutuhan listrik. Empat PLTU ini dapat memberikan sumbangan daya sebesar ± 600 MW.

2.1.1 PLTGU

Pembangkit Listrik Tenaga Gas dan Uap (PLTGU) adalah gabungan antara PLTG dengan PLTU, dimana panas dari gas buang dari PLTG digunakan untuk menghasilkan uap yang digunakan sebagai fluida kerja di PLTU. Dan bagian yang digunakan untuk menghasilkan uap tersebut adalah HRSG (*Heat Recovery Steam Generator*). PLTGU merupakan suatu instalasi peralatan yang berfungsi untuk mengubah energi panas (hasil pembakaran bahan bakar dan udara) menjadi energi listrik yang bermanfaat. Pada dasarnya, sistem PLTGU ini merupakan penggabungan antara PLTG dan PLTU. PLTU memanfaatkan energi panas dan uap dari gas buang hasil pembakaran di PLTG untuk memanaskan air di HRSG (*Heat Recovery Steam Generator*), sehingga menjadi uap jenuh kering. Uap jenuh kering inilah yang akan digunakan untuk memutar sudu (baling-baling). Gas yang dihasilkan dalam ruang bakar pada Pusat Listrik Tenaga Gas (PLTG) akan menggerakkan turbin dan kemudian *generator*, yang akan mengubahnya menjadi energi listrik. Penggunaan bahan bakar menentukan tingkat efisiensi pembakaran dan prosesnya.

2.1.2 Proses Konversi Energi PLTGU

Pertama, turbin gas berfungsi menghasilkan energi mekanik untuk memutar kompresor dan *rotor generator* yang terpasang satu poros, tetapi pada saat *start up* fungsi ini terlebih dahulu dijalankan oleh penggerak mula (*prime mover*). Penggerak mula ini dapat berupa *diesel*, motor listrik atau *generator* turbin gas itu sendiri yang menjadi motor melalui mekanisme SFC (*Static frequency Converter*). Setelah kompresor berputar secara kontinu, maka udara luar terhisap hingga dihasilkan udara bertekanan pada sisi *discharge* (tekan) kemudian masuk ke ruang bakar. Kedua, proses selanjutnya pada ruang bakar, jika *start up* menggunakan bahan bakar cair (*fuel oil*) maka terjadi proses pengkabutan (*atomizing*) setelah itu terjadi proses pembakaran dengan penyala awal dari busi, yang kemudian dihasilkan api dan gas panas bertekanan. Gas panas tersebut dialirkan ke turbin sehingga turbin dapat menghasilkan tenaga mekanik berupa putaran.



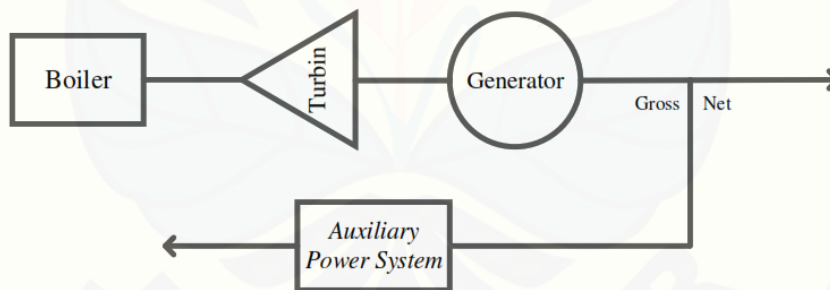
Gambar 2.1 Proses Konversi Energi Pada PLTGU

2.2 Karakteristik Input-Output

Dalam pengoperasian sebuah pembangkit secara terus menerus, pastinya akan mempengaruhi performa dari pembangkit itu sendiri. Pembangkit dapat mengalami penurunan performa baik sedikit demi sedikit maupun secara drastis. Untuk itu dibutuhkan suatu acuan untuk mengetahui apakah performa pembangkit masih dalam keadaan stabil atau mengalami penurunan. Dalam hal

ini, karakteristik *input-output* dapat diterapkan untuk mengetahui keadaan performa pembangkit termal.

Dasar dari permasalahan pengoperasian pembangkit secara ekonomis adalah mengatur karakteristik *input-output* dari unit pembangkit termal itu sendiri. Karakteristik unit pembangkit termal terdiri dari boiler, turbin, generator dan *auxiliary power system* seperti yang ditunjukkan pada gambar 2.2. *Output* pembangkit selain disalurkan melalui transmisi juga akan digunakan oleh *auxiliary power system* atau sistem tenaga bantu untuk beroperasi. Pada gambar tersebut juga terdapat *gross* atau *input* kotor dan juga *net* yang direpresentasikan sebagai *output* bersih. *Input* kotor merupakan *input* total yang diukur dalam biaya/jam sedangkan *output* bersih merupakan daya *output* yang dihasilkan. Dengan menggunakan karakteristik *input-output* pada masing-masing pembangkit, maka dapat ditentukan pengoperasian optimum secara ekonomis sejumlah *unit* pembangkit untuk menentukan total biaya operasinya paling minimal.



Gambar 2.2 Karakteristik Unit Pembangkit

Karakteristik *input-output* pembangkit termal merupakan suatu karakteristik yang menggambarkan hubungan antara *input* dan *output* yang dimiliki oleh pembangkit. *Input* yang dimaksudkan adalah bahan bakar (liter/jam) yang dipakai oleh pembangkit. Sedangkan *output* yang dihasilkan adalah berupa daya (MW). Pada umumnya karakteristik *input-output* pembangkit termal didekati dengan fungsi polinomial orde dua, yaitu (AM. Ilyas dkk, 2010) :

$$H_i = \alpha_i + \beta P + \gamma P^2 \dots\dots\dots(2.1)$$

dengan :

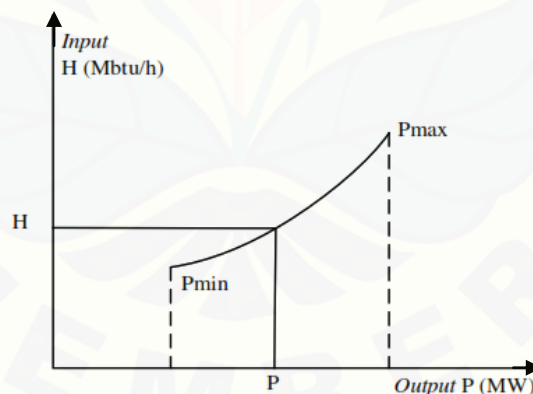
H_i = *input* bahan bakar pembangkit termal ke- i (liter/jam)

P_i = *output* pembangkit termal ke- i (MW)

α_i, β, γ = konstanta *input* - *output* pembangkit termal ke- i .

Penentuan parameter α_i, β, γ membutuhkan data yang berhubungan dengan *input* bahan bakar H_i dan *output* pembangkit P_i .

Gambar 2.3 merupakan karakteristik *input–output* dari *unit* pembangkit termal yang ideal. Kurva ideal digambarkan sebagai kurva *non-linier* yang kontinyu. Pada grafik terdapat variabel *input* H yang direpresentasikan sebagai energi panas yang dibutuhkan dengan satuan Mbtu/h. Sedangkan *output* simbolkan dengan P yang berarti daya dengan satuan MW. Pada gambar ditunjukkan sebuah kurva hubungan *input* dan *output* yang dibatasi oleh P_{min} dan P_{max} .



Gambar 2.3 Karakteristik *Input-Output* Unit Pembangkit Termal Ideal

Data karakteristik *input–output* diperoleh dari perhitungan desain atau dari pengukuran. Jika digunakan data pengukuran akan diperoleh kurva yang tidak kontinyu. Dan pada pembangkit termal mempunyai batas operasi minimum dan maksimum. Batasan ini biasanya disebabkan oleh kestabilan pembakaran dan masalah desain generator. Pada umumnya *unit* pembangkit termal tidak dapat beroperasi dibawah 30% dari kapaitas desain (Neny, 2014).

2.2.1 Metode *Quadratic Least Square Regression*

Metode *least square* menyatakan bahwa “jumlah kuadrat selisih dari nilai sebenarnya dengan nilai yang terhitung, dikalikan jumlah pengukuran adalah minimum”. Metode *least square* merupakan metode estimasi parameter sistem yang meminimumkan fungsi kriteria jumlah kuadrat kesalahan prediksi (Cahyo Adi dkk,2011). Salah satu cabang dari metode *least square* adalah metode *quadratic least square regression* merupakan salah satu metode pendekatan yang dapat digunakan untuk menyelesaikan persoalan pada karakteristik *input–output* pembangkit termal. Metode ini banyak digunakan untuk regresi ataupun pembentukan persamaan dari titik–titik data diskrit dalam suatu pemodelan dan juga dapat digunakan untuk analisis.

Metode ini termasuk dalam metode–metode pendekatan sesatan terdistribusi. Berdasarkan karakteristik kerjanya yang melakukan pengurangan sesatan menyeluruh yang terukur berdasarkan interval pendekatan keseluruhan sesuai dengan order pendekatan yang meningkat. Metode *quadratic least square regression* juga memainkan peranan penting dalam teori statistik, karena metode ini seringkali digunakan dalam penyelesaian masalah–masalah yang melibatkan kumpulan data yang tersusun secara acak.

Pada tugas akhir ini, metode *quadratic least square regression* digunakan untuk melakukan regresi atau pencocokan kurva pada permasalahan karakteristik *input- output* pembangkit termal yang diharapkan dapat membentuk persamaan matematis tertentu. Karakteristik *input–output* pembangkit termal merupakan fungsi polinomial orde dua, sehingga digunakan persamaan berikut :

- a. Untuk mencari konstanta (α, β, γ)

$$\alpha = \frac{\{[\sum(x^2y) * \sum(x^2) - \sum(xy) * \sum(x^3)]\}}{\{[\sum(x^2) * \sum(x^4) - [\sum(x^3)]^2]\}} \dots\dots\dots(2.2)$$

$$\beta = \frac{\{[\sum(xy) * \sum(x^4) - \sum(x^2y) * \sum(x^3)]\}}{\{[\sum(x^2) * \sum(x^4) - [\sum(x^3)]^2]\}} \dots\dots\dots(2.3)$$

$$y = \left[\frac{\sum y_i}{n} \right] - \left\{ b * \left[\frac{\sum x_i}{n} \right] \right\} - \left\{ a * \left[\frac{\sum x^2 i}{n} \right] \right\} \dots\dots\dots(2.4)$$

b. Persamaan Kuadrat Regresi Statistik

$$\sum x^2 = (\sum x_i^2) - \left[\frac{\sum x_i^2}{n} \right] \dots\dots\dots(2.5)$$

$$\sum xy = (\sum x_i y_i) - \left[\frac{\sum x_i * \sum y_i}{n} \right] \dots\dots\dots(2.6)$$

$$\sum x^3 = (\sum x_i^3) - \left[\frac{\sum x_i * \sum x_i^2}{n} \right] \dots\dots\dots(2.7)$$

$$\sum x^2 y = (\sum x_i^2 y_i) - \left[\frac{\sum x_i^2 * \sum y_i}{n} \right] \dots\dots\dots(2.8)$$

$$\sum x^4 = (\sum x_i^4) - \left[\frac{(\sum x_i^2)^2}{n} \right] \dots\dots\dots(2.9)$$

dengan keterangan :

Yi = nilai individu untuk setiap gabungan *variable*

Xi = nilai individu untuk setiap gabungan *variable*

n = jumlah data

2.3 Karakteristik Beban Tenaga Listrik

Secara umum beban yang dilayani oleh sistem distribusi elektrik ini dibagi dalam beberapa sektor yaitu sektor perumahan, sektor industri, sektor komersial dan sektor usaha. Masing-masing sektor beban tersebut mempunyai karakteristik-karakteristik yang berbeda, sebab hal ini berkaitan dengan pola konsumsi energi pada masing-masing konsumen di sektor tersebut. Karakteristik beban yang banyak disebut dengan pola pembebanan pada sektor perumahan ditunjukan oleh adanya fluktuasi konsumsi energi elektrik yang cukup besar. Hal ini disebabkan konsumsi energi elektrik tersebut dominan pada malam hari. Sedang pada sektor industri fluktuasi konsumsi energi sepanjang hari akan hampir sama, 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.

1. Faktor Beban (*load factor*)

Faktor beban adalah perbandingan antara beban rata – rata terhadap beban puncak yang diukur dalam suatu periode tertentu. Beban rata – rata dan beban puncak dapat dinyatakan dalam kilowatt, kilovolt – amper, amper dan sebagainya, tetapi satuan dari keduanya harus sama. Faktor beban dapat dihitung untuk periode tertentu biasanya dipakai harian, bulanan atau tahunan.

Beban puncak yang dimaksud disini adalah beban puncak sesaat atau beban puncak rata-rata dalam interval tertentu (demand maksimum), pada umumnya dipakai demand maksimum 15 menit atau 30 menit. Definisi dari faktor beban ini dapat dituliskan dalam persamaan berikut ini:

$$\text{Faktor beban (Fb)} = \frac{\text{beban rata-rata dalam periode tertentu}}{\text{beban puncak dalam periode tersebut}} \dots\dots\dots(2.10)$$

Faktor beban dapat diketahui dari kurva bebannya. Sedangkan untuk perkiraan besaran faktor beban di masa yang akan datang dapat didekati dengan kata data statistik yang ada berdasarkan jenis bebannya. Bila diterapkan pada pusat pembangkit maka di dapat, menurut definisi :

$$\text{Faktor beban (Fb)} = \frac{P_{\text{rata-rata}}}{P_{\text{puncak}}} = \frac{P_{\text{rata-rata}}}{P_p} \times \frac{T}{T} \dots\dots\dots(2.11)$$

dengan :

T = periode waktu

Prata-rata = Beban rata – rata dalam periode T

P_p= beban puncak yang terjadi dalam periode T pada selang waktu tertentu (15 menit atau 30 menit).

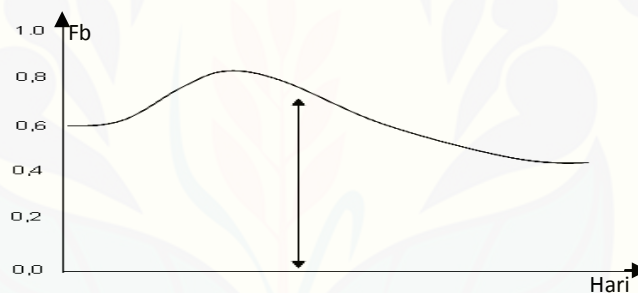
Bila P_r dan P_p dalam kW dan T dalam jam. Bila T dalam setahun, maka didapat faktor beban tahunan, bila dalam satu bulan didapat faktor beban bulanan dan bila harian, faktor beban harian.

2. Beban Harian

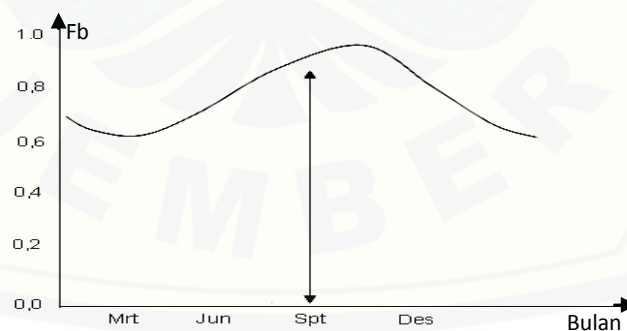
Faktor beban harian, bervariasi menurut karakteristik dari daerah beban tersebut, apakah daerah pemukiman, daerah industry, perdagangan ataupun gabungan dari bermacam pemakai/pelanggan, juga bagaimana keadaan cuaca atau juga apakah hari libur dan sebagainya.

3. Faktor Beban harian rata – rata

Faktor beban harian rata – rata, gambar berikut merupakan dasar dari pada faktor beban tahunan total.



Gambar 2.4 Beban Harian



Gambar 2.5 Beban Bulanan

Selanjutnya, dapat dilihat beban puncak bulanan rata – rata terhadap beban puncak tahunan, lihat gambar 2.6. misalkan P_{pt} = puncak tahunan (annual load faktor), maka ini dapat dihitung sebagai berikut :

$$F_{btahunan} = F_{bharian} \times \frac{P_{ph}}{P_{pb}} \times \frac{P_{pb}}{P_{pt}} \dots\dots\dots(2.12)$$

Keterangan :

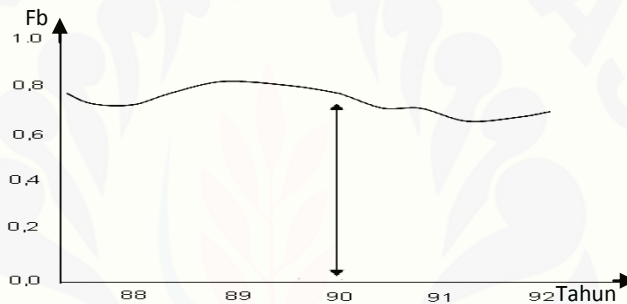
Fbt = faktor beban tahunan

Fbh = faktor beban harian

Pph = beban puncak rata – rata harian

Ppb = beban puncak rata – rata bulanan

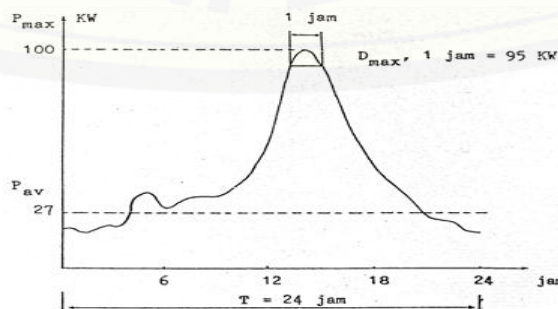
Ppt = beban puncak rata – rata tahunan



Gambar 2.6 Kurva Beban Tahunan

4. Beban Puncak (Peak Load)

Beban Puncak (P_{max}) adalah nilai terbesar dari pembebanan sesaat pada suatu *interval demand* tertentu. Untuk dapat memperjelas pengertian mengenai *Demand* (D), *Maximum Demand* (D_{max}) dan Beban Puncak (P_{max}) dapat dilihat pada Gambar 2.7 dibawah ini.



Gambar 2.7 Perubahan kebutuhan maksimum terhadap waktu

Interval Demand : $T = 24$ jam

Demand = P_{av} : $D = 27$ kW

Maximum Demand : D_{max} , 1 jam = 95 kW

Beban Puncak : $P_{max} = 10$ kW

5. Beban Terpasang (*Connected Load*)

Beban terpasang dari suatu sistem adalah jumlah total daya dari seluruh peralatan sesuai dengan KW atau KVA yang tertulis pada papan nama (*name plat*) peralatan yang akan dilayani oleh sistem tersebut.

$$\text{Jadi : } P_L = \sum_{i=1}^n P_i \dots\dots\dots(2.13)$$

Keterangan :

P_i = *rating* KVA dari alat i

n = jumlah alat yang terhubung ke sistem.

6. Faktor Keragaman (*Diversity Factor*)

Faktor keragaman (f_{div}) didefinisikan sebagai perbandingan antara jumlah beban maksimum dari masing masing unit beban yang ada pada suatu sistem terhadap beban maksimum sistem secara keseluruhan.

$$\begin{aligned} \text{Jadi faktor keragaman : } F_{div} &= \frac{D_{max\ 1} + D_{max\ 2} + \dots + D_{max\ n}}{D_{max\ (1+2+\dots+n)}} \dots\dots\dots(2.14) \\ &= \frac{\sum_{i=1}^n D_{max}}{D_{max\ s}} \end{aligned}$$

Keterangan :

$D_{max\ i}$ = beban maksimum beban unit ke i

$D_{max\ s}$ = beban maksimum sistem

Bila $D_{\max i}$ untuk seluruh unit bersamaan waktunya maka f_{div} akan berharga 1, tetapi bila tidak f_{div} akan lebih besar dari 1.

7. Faktor Keserempakan (*Coincidence Factor*)

Faktor keserempakan (f_{cf}) adalah kebalikan dari faktor keragaman, yang didefinisikan sebagai perbandingan antara beban maksimum dari suatu kumpulan beban dari sistem terhadap jumlah beban maksimum dari masing-masing unit beban.

$$\text{Jadi : } F_{div} = \frac{D_{\max 1} + D_{\max 2} + \dots + D_{\max n}}{D_{\max (1+2+\dots+n)}} = \frac{D_s}{\sum_{i=1}^n D_{\max i}} = \frac{1}{(f_{div})} \dots \dots \dots (2.15)$$

8. Faktor Kebutuhan (*Demand Factor*)

Faktor kebutuhan didefinisikan sebagai perbandingan antara beban puncak suatu sistem terhadap beban terpasang yang dilayani oleh sistem. Nilai f_d pada prinsipnya lebih kecil atau sama dengan satu. Bisa saja terjadi lebih besar dari satu, yaitu saat terjadi dibeban lebih.

9. Faktor Beban (*Load Factor*)

Faktor beban adalah perbandingan antara beban rata-rata selama interval tertentu dengan beban puncak yang terjadi pada interval yang sama.

dimana: p_{av} = beban rata-rata dan P_{\max} = beban puncak.

Faktor beban adalah perbandingan antara beban rata – rata terhadap beban puncak dalam periode tertentu. Beban rata – rata dan beban puncak dapat dinyatakan dalam kilowatt, kilovolt – amper, amper dan sebagainya, tetapi satuan dari keduanya harus sama. Faktor beban dapat dihitung untuk periode tertentu biasanya dipakai harian, bulanan atau tahunan.

$$\text{Faktor beban (Fb)} = \frac{\text{beban rata – rata dalam periode tertentu}}{\text{beban puncak dalam periode tertentu}} \dots \dots \dots (2.16)$$

2.4 Peramalan Beban Tenaga Listrik

Peramalan pada dasarnya merupakan suatu dugaan atau prakiraan mengenai terjadinya suatu kejadian atau peristiwa dimasa yang akan datang. Dalam kegiatan perencanaan peramalan merupakan kegiatan mula dari proses tersebut.

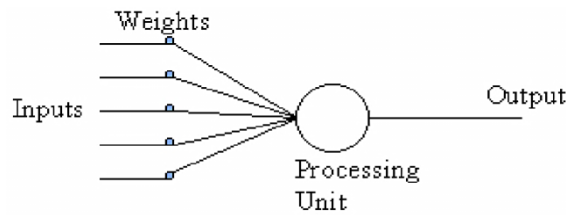
Ramalan di bidang tenaga elektrik pada dasarnya merupakan ramalan kebutuhan energi elektrik dan ramalan beban tenaga elektrik (watt). Keduanya sering disebut dengan istilah *Demand and Load Forecasting*. Hasil peramalan ini dipergunakan untuk membuat rencana pemenuhan kebutuhan maupun pengembangan penyediaan tenaga elektrik setiap saat secara cukup dan baik serta terus menerus. Secara garis besar pembuatan ramalan kebutuhan tenaga elektrik dapat dibagi dalam tiga tahap, yaitu :

- a. Pengumpulan dan penyiapan data.
- b. Pengolahan dan analisa data.
- c. Penentuan metoda dan pembuatan model.

2.5 Neural Network (NN)

Secara umum *Neural Network* (NN) adalah jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan jaringan syaraf manusia. NN ini merupakan sistem adaptif yang dapat merubah strukturnya untuk memecahkan masalah berdasarkan informasi eksternal maupun internal yang mengalir melalui jaringan tersebut. Secara sederhana NN adalah sebuah alat pemodelan data statistik *non-linear*. NN dapat digunakan untuk memodelkan hubungan yang kompleks antara *input* dan *output* untuk menemukan pola-pola pada data.

Secara mendasar, sistem pembelajaran merupakan proses penambahan pengetahuan pada NN yang sifatnya kontinuitas sehingga pada saat digunakan pengetahuan tersebut akan dieksploitasikan secara maksimal dalam mengenali suatu objek. *Neuron* adalah bagian dasar dari pemrosesan suatu *Neural Network* (Noviana, 2008). Dibawah ini merupakan bentuk dasar dari suatu *neuron*.



Gambar 2.8 Bentuk dasar *neuron* (Noviana, 2008)

Keterangan Gambar 2.8 di atas adalah sebagai berikut:

1. *Input*= merupakan masukan yang digunakan baik saat pembelajaran maupun dalam mengenali suatu objek.
2. *Weight* = beban yang selalu berubah setiap kali diberikan *input* sebagai proses pembelajaran.
3. *Processing Unit*= merupakan tempat berlangsungnya proses pengenalan suatu objek berdasarkan pembebanan yang diberikan.
4. *Output* = keluaran dari hasil pengenalan suatu objek.

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

1. Perangkat mampu untuk mengenali suatu objek secara *non-linier*.
2. Mempermudah pemetaan *input* menjadi suatu hasil tanpa mengetahui proses sebenarnya.
3. Mampu melakukan pengadaptasian terhadap pengenalan suatu objek.
4. Perangkat yang memiliki toleransi terhadap suatu kesalahan dalam pengenalan suatu objek.
5. *Neural network* mampu diimplementasikan pada suatu *hardware* atau perangkat keras.

Suatu sistem *neural network* mempunyai beberapa parameter yang menentukan keberhasilan sistem, yaitu:

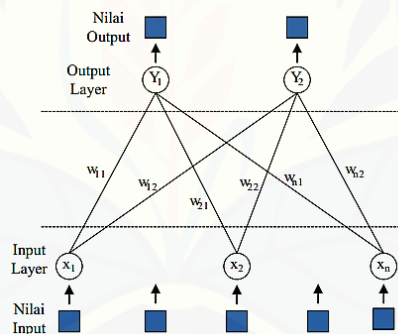
1. Pola hubungan antar *neuron* (disebut arsitektur jaringan).
2. Metode untuk menentukan bobot penghubung (disebut metode *training/learning/algorithm*).
3. Fungsi aktivasi.

2.5.1 Arsitektur Neural Network

Beberapa arsitektur jaringan yang sering dipakai dalam neural network antara lain:

1. Jaringan Layar Tunggal (*single layer network*)

Jaringan dengan lapisan tunggal terdiri dari 1 *layer input* dan 1 *layer output*. Setiap *neuron* yang terdapat di dalam *layer input* selalu terhubung dengan setiap *neuron* yang terdapat pada *layer output*. Jaringan ini hanya menerima *input* kemudian secara langsung akan mengolahnya menjadi *output* tanpa harus melalui *hidden layer*. Contoh algoritma JST yang menggunakan metode ini yaitu *adaline* dan *perceptron*. Pada Gambar 2.9 diperlihatkan arsitektur *single layer network* dengan n buah *input layer* (X_1, X_2, \dots, X_n) dan dua buah *output layer* (Y_1, Y_2) dengan bobot masing-masing yang menghubungkan antara *input layer* dengan *output layer* ($W_{11}, W_{12}, W_{21}, W_{22}, W_{n1}, W_{n2}$).

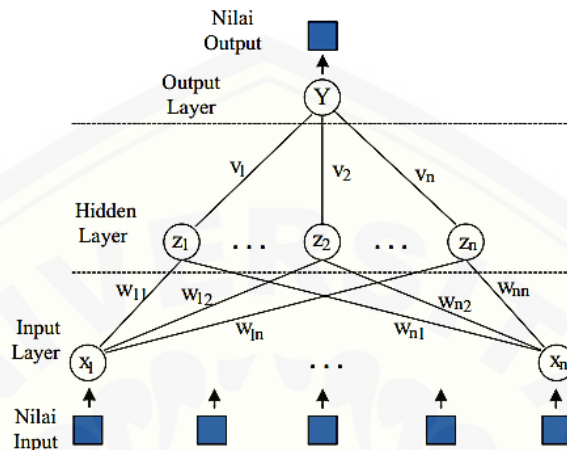


Gambar 2.9 Arsitektur *single layer network* (Siregar, 2013)

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

Jaringan dengan lapisan jamak memiliki ciri khas tertentu yaitu memiliki 3 jenis *layer* yaitu *layer input*, *layer output*, dan *layer* tersembunyi. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang kompleks dibandingkan jaringan dengan lapisan tunggal. Namun, proses pelatihan sering membutuhkan waktu cenderung lama. Contoh algoritma *neural network* yang menggunakan metode ini yaitu *backpropagation*. Pada Gambar 2.10 diperlihatkan arsitektur *multi layer network* dengan n buah *input layer* (X_1, \dots, X_n), n buah *hidden layer* (Z_1, Z_2, \dots, Z_n) dan sebuah *output layer* (Y) dengan bobot yang

menghubungkan *input layer* dengan *hidden layer* ($W_{11}, W_{12}, W_{1n}, W_{n1}, W_{n2}, W_{nn}$) dan bobot yang menghubungkan *hidden layer* dengan *output layer* (V_1, V_2, \dots, V_n).



Gambar 2.10 Multi layer network (Siregar, 2013)

2.5.2 Fungsi Transfer

Karakter dari *neural network* tergantung atas bobot dan fungsi *input – output* (fungsi transfer) yang mempunyai ciri tertentu untuk setiap unit. Fungsi ini terdiri dari 3 kategori yaitu :

1. *Linear Units*, Aktifitas *output* adalah sebanding dengan jumlah bobot *output*.
2. *Threshold Units*, *Output* diatur satu dari dua tingkatan tergantung dari apakah jumlah *input* adalah lebih besar atau lebih kecil dari nilai ambang.
3. *Sigmoid Units*, *Output* terus menerus berubah-ubah tetapi tidak berbentuk *linear*. *Unit* ini mengandung kesamaan yang lebih besar dari sel syaraf sebenarnya dibandingkan dengan *linear* dan *threshold unit*, namun ketiganya harus dipertimbangkan dengan perkiraan kasar.

Untuk membuat *neural network* agar dapat melakukan beberapa kerja khusus, harus dipilih *unit-unit* yang akan dihubungkan antara satu dengan yang lain dan harus bisa mengatur bobot dari hubungan tersebut secara tepat. Hubungan tersebut akan menentukan apakah mungkin suatu *unit* mempengaruhi *unit* yang lain (Nasution, 2009).

2.5.3 Aplikasi Neural Network

Saat ini *neural network* banyak dipakai dalam berbagai sistem. Beberapa aplikasi *neural network* adalah sebagai berikut:

1. Pengenalan Pola (*Pattern Recognition*)

Neural network dapat dipakai untuk mengenali pola (misal huruf, angka, suara atau tanda tangan) yang sudah sedikit berubah. Hal ini mirip dengan otak manusia yang masih mampu mengenali orang yang sudah beberapa waktu tidak dijumpainya (mungkin wajah atau bentuk tubuhnya yang sudah sedikit berubah).

2. *Signal Processing*

Neural network (model *Adaline*) dapat dipakai untuk menekan *noise* dalam saluran telpon.

3. Peramalan

Neural network juga dapat dipakai untuk meramalkan apa yang akan terjadi di masa yang akan datang berdasarkan pola kejadian yang ada di masa yang lampau. Peramalan ini dapat dilakukan mengingat kemampuan *neural network* dapat membuat generalisasi dari apa yang sudah ada sebelumnya. *Neural network* juga dapat menyelesaikan masalah dalam bidang kontrol dan kedokteran. (Siregar, 2013).

2.5.4 Backpropagation Neural Network

Backpropagation 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.

2.5.5 *Constructive Backpropagation Neural Network (CBNN)*

CBNN (*Constructive Backpropagation Neural Network*) digunakan pada simulasi ini dikarenakan CBNN memiliki keunggulan yaitu dalam sistemnya bisa memasukkan beberapa *hidden layer* dan *neuron* sesuai yang diinginkan dan *program* tersebut bisa dijalankan sekaligus tanpa harus mengganti satu persatu *hidden layer* dan neuronnya.

Selain itu sistem ini mempunyai kemampuan untuk melakukan penyesuaian arsitektur jaringan dengan kompleksitas permasalahan yang dihadapi selama proses pembelajaran. Dengan aturan pembelajaran tersebut akan memperkecil *error* besaran sebenarnya dengan estimasi yang akan diperoleh. Rangkaian CBNN ini yang digunakan pada simulasi yang kemudian hasilnya akan dianalisis.

Algoritma *backpropagation* terdiri dari duabagian antara lain sebagai berikut: (Oktaorora, 2011)

a. Algoritma Pelatihan *Backpropagation (Training Proccess)*

Di dalam proses pelatihan *backpropagation* terdapat tiga tahap. Tahap pertama ialah tahap maju (*feed-forward*). Pada tahap ini seluruh proses awal inisialisasi bobot-bobot *input* dilakukan. Pada tahap ini juga ditentukan angka pembelajaran (α), nilai toleransi *error* dan jumlah *epoch* (siklus setiap pola pelatihan) yang diperlukan selama proses komputasi berlangsung.

Setelah semua proses inisialisasi dilakukan, maka langkah selanjutnya ialah proses maju. Setiap unit masukan x_i akan mengirimkan sinyal masukan ke lapisan tersembunyi. Setelah dihitung dengan menggunakan fungsi aktivasi maka keluarannya akan dikirimkan ke lapisan di atasnya, yaitu lapisan *output*. Setelah nilai keluaran (y_k) diperoleh, maka dibandingkan dengan target keluaran sebenarnya (t_k). Selisih $y_k - t_k$ disebut dengan *error* (δ_k). Jika nilai *error* lebih

kecil atau sama dengan dari nilai ambang maka proses iterasi dihentikan, tetapi jika tidak maka nilai *error* tersebut digunakan untuk memodifikasi bobot-bobot untuk mengoreksi kesalahan yang terjadi.

Tahap kedua adalah tahap mundur atau *backpropagation*. Pada tahap ini, nilai *error* (δ_k) yang diperoleh pada di lapisan *output* digunakan untuk mengoreksi bobot-bobot yang ada pada lapisan tersembunyi yang berhubungan langsung dengan lapisan *output*. Setelah itu nilai *error* (δ_j) di setiap *unit* pada lapisan tersembunyi juga dihitung untuk mengoreksi bobot-bobot yang menghubungkan lapisan *input* dengan lapisan tersembunyi. Tahap ketiga adalah tahap pengoreksian bobot. Setelah seluruh bobot pada lapisan *input* dan lapisan tersembunyi dimodifikasi sesuai dengan besar faktor *error*nya, maka ketiga fase ini diulang secara terus menerus sampai kondisi berhenti dipenuhi. Kondisi berhenti yang dimaksud adalah jika jumlah *epoch* yang ditetapkan tercapai atau jika nilai *error* jaringan telah sama dengan atau lebih kecil dari nilai toleransi *error* yang ditetapkan sebelumnya. Pada tahap pelatihan, jaringan diharapkan dapat melatih seluruh data pelatihan yang diberikan untuk mendapatkan bobot akhir jaringan yang akan digunakan pada tahap pengujian. Struktur algoritma pelatihan *backpropagation* antara lain:

1. Inisialisasi bobot-bobot

Mentukan angka pembelajaran (α). Tentukan pula nilai toleransi error yang diinginkan dan set maksimal *epoch* jika ingin membatasi jumlah *epoch* yang digunakan.

2. Selama kondisi berhenti tidak terpenuhi, lakukan langkah ke-2 sampai 9.

3. Setiap pasangan pola pelatihan, lakukan langkah ke-3 sampai ke-8.

Tahap maju (Feedforward)

4. Tiap-tiap unit *input* (x_i , $i = 1, 2, 3, \dots, o$) menerima sinyal *input* dan meneruskannya ke tiap-tiap unit pada lapisan tersembunyi.

5. Tiap-tiap unit di lapisan tersembunyi (z_j , $j = 1, 2, 3, \dots, p$) menjumlahkan sinyal *input* yang berbobot, yaitu:

$$z_{net_j} = v_{j0} + \sum_{i=1}^n x_i v_{ji} \dots \dots \dots (2.17)$$

Keterangan : x : input

z : lapisan tersembunyi

Fungsi aktivasi untuk menghitung sinyal *output*, yaitu:

$$z_j = v_{j0} + \sum_{i=1}^n x_i v_{ji} \dots \dots \dots (2.18)$$

dan mengirimkan sinyal tersebut ke semua unit lapisan di atasnya (lapisan *output*).

6. Tiap-tiap unit di lapisan *output* ($y_k, k = 1, 2, 3, \dots, m$) menjumlahkan sinyal *input* yang berbobot, yaitu:

$$y_{net_k} = w_{k0} + \sum_{j=1}^p z_j w_{kj} \dots \dots \dots (2.19)$$

Fungsi aktivasi untuk menghitung sinyal *output*, yaitu:

$$y_k = w_{k0} + \sum_{j=1}^p z_j w_{kj} \dots \dots \dots (2.20)$$

Keterangan : x : input

y : output

z : lapisan tersembunyi

w : bobot

Tahap mundur(Backpropagation)

7. Tiap-tiap unit *output* y_k menerima pola target t_k untuk menghitung error (δ_k), yaitu:

$$\delta_k = (t_k - y_k) f'(y_{net_k}) = (t_k - y_k) y_k (1 - y_k) \dots \dots \dots (2.21)$$

Kemudian menghitung nilai koreksi bobot yang nantinya digunakan untuk memperbaiki nilai bobot antara lapisan tersembunyi dan lapisan *output* (w_{jk}), yaitu:

$$\Delta w_{jk} = \alpha \delta_k z_j \dots \dots \dots (2.22)$$

Menghitung juga koreksi bias yang digunakan untuk memperbaiki nilai bias antara lapisan tersembunyi dan lapisan *output* (wk_0), yaitu:

$$\Delta w_{k0} = \alpha \delta_k \dots \dots \dots (2.23)$$

8. Tiap-tiap unit pada lapisan tersembunyi ($z_j, j = 1, 2, 3, \dots, p$) menjumlahkan sinyal-sinyal *input* dari lapisan *output*, yaitu:

$$\delta_{net_j} = \sum_{k=1}^n \delta_k w_{jk} \dots \dots \dots (2.24)$$

Mengalikan nilai ini dengan fungsi aktivasi untuk menghitung *error* pada lapisan tersembunyi (δ_j), yaitu:

$$\delta_j = \delta_{net_j} f'(z_{net_j}) = \delta_{net_j} z_j (1 - z_j) \dots \dots \dots (2.25)$$

Kemudian hitung koreksi bobot untuk memperbaiki nilai bobot antara lapisan *input* dan lapisan tersembunyi (v_{ji}), yaitu:

$$\Delta v_{ji} = \alpha \delta_j x_i \dots \dots \dots (2.26)$$

Kemudian menghitung koreksi bias untuk memperbaiki nilai bobot antara lapisan *input* dan lapisan tersembunyi (v_{j0}), yaitu:

$$\Delta v_{j0} = \alpha \delta_j \dots \dots \dots (2.27)$$

Tahap pengoreksian bobot

9. Tiap unit keluaran ($y_k, k = 1, 2, 3, \dots, m$) memperbaiki bobot dan bias, yaitu:

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj}, (k = 1, 2, \dots, m; j = 0, 1, \dots, p) \dots (2.28)$$

Tiap-tiap unit tersembunyi memperbaiki bobot dan bias, yaitu:

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji}, (j = 1, 2, \dots, p; i = 0, 1, \dots, n) \dots \dots (2.29)$$

BAB 3. METODOLOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

Penelitian peramalan jangka pendek beban tenaga listrik dan konsumsi bahan bakar gas menggunakan metode *constructive backpropagation neural network* dilaksanakan pada PLTGU *unit* 3 dengan kapasitas 200 MW pada PT PJB UP Gresik Jl. Harun Tohir no.1 Gresik, Jawa Timur.

Periode waktu penelitian untuk pengambilan data pada PT PJB Gresik dilaksanakan pada bulan Maret 2017 dengan data beban dan bahan bakar yang diperoleh ialah data pada periode Januari 2014 hingga November 2016.

3.2 Alat dan Bahan

Pada penelitian ini menggunakan sebuah *personal computer* sebagai media penunjang pelaksanaan penelitian dan dilengkapi aplikasi MATLAB R2014b yang nantinya digunakan untuk pengerjaan peramalan sistem yang akan dijalankan

3.3 Tahap Penelitian

Tahapan yang dilakukan pada penelitian Peramalan Jangka Pendek Beban Tenaga Listrik dan Konsumsi Bahan Bakar Gas Pada PT. PJB UP Gresik Menggunakan Metode *Constructive Backpropagation Neural Network* sebagai berikut :

1. Studi literatur

Tahap pertama yang dilakukan dalam memulai penelitian ini ialah dengan mencari dan memilih literatur yang berkaitan dengan penelitian ini, dengan harapan literatur yang telah didapat mampu membantu dalam proses perancangan dan pelaksanaan penelitian agar hasil yang didapatkan sesuai dengan yang diharapkan.

2. Perancangan sistem peramalan

Persiapan perancangan sistem peramalan dimulai dengan membuat dan menentukan hubungan antara beban, kebutuhan gas, serta kemampuan

pembangkitan pada sistem yang nantinya akan digunakan sebagai metode peramalan pada PT PJB UP Gresik. Selain itu juga dilakukan perancangan konfigurasi *constructive backpropagation neural network* yang digunakan sebagai metode dalam peramalan.

3. Pengambilan data *input* dan target.

Pada perancangan tahap ini data *input* berupa beban sesungguhnya tenaga listrik wilayah Gresik dan juga data jumlah penggunaan gas sebagai bahan bakar pada PLTU di PT PJB UP Gresik. Data yang akan digunakan untuk peramalan beban adalah sebagai berikut.

- a. Data beban listrik tahun 2014 - 2016.
 - b. Data penggunaan gas sebagai bahan bakar pembangkitan tahun 2014 - 2016
 - c. Data beban listrik dan data penggunaan gas yang akan digunakan merupakan data historis.
 - d. Data akan dibagi menjadi dua bagian, yaitu data pelatihan (*training*) dan data pengujian (*testing*).
- ### 4. Konfigurasi peramalan dengan metode *constructive backpropagation neural network*.

Tahap keempat adalah pembuatan sistem peramalan dengan metode *constructive backpropagation neural network*. Pembuatan *neural network* menggunakan beban dengan membangun jaringan dan inisialisasi awal jaringan *constructive backpropagation neural network* menggunakan bantuan *software matlab*.

5. *Training neural network*.

Setelah sistem perencanaan konfigurasi peramalan menggunakan *neural network* selesai, maka selanjutnya dilakukan *training* terhadap hasil dari metode *constructive backpropagation neural network* dan akan muncul grafik serta data sebagai hasil peramalan tersebut. Tujuan dari pelatihan (*training*) adalah untuk melakukan proses pembelajaran dengan *constructive backpropagation* untuk menentukan nilai bobot baru. Data *input* dan data target dilatih untuk mendapatkan suatu nilai bobot baru. Parameter yang diperhatikan dalam inisialisasi pada jaringan adalah:

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

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

6. *Testing*

Proses pengujian (*testing*) adalah proses melakukan tes pembelajaran dengan menggunakan bobot terlatih untuk melakukan peramalan. Dalam proses pengujian, data yang diuji adalah data beban dan data konsumsi gas PT PJB UP Gresik. Sedangkan untuk pengujian peramalan bahan bakar menggunakan metode *quadratic least square* yang dikombinasikan dengan *Graphical User Interface* (GUI)

7. Analisis data dan pembuatan laporan

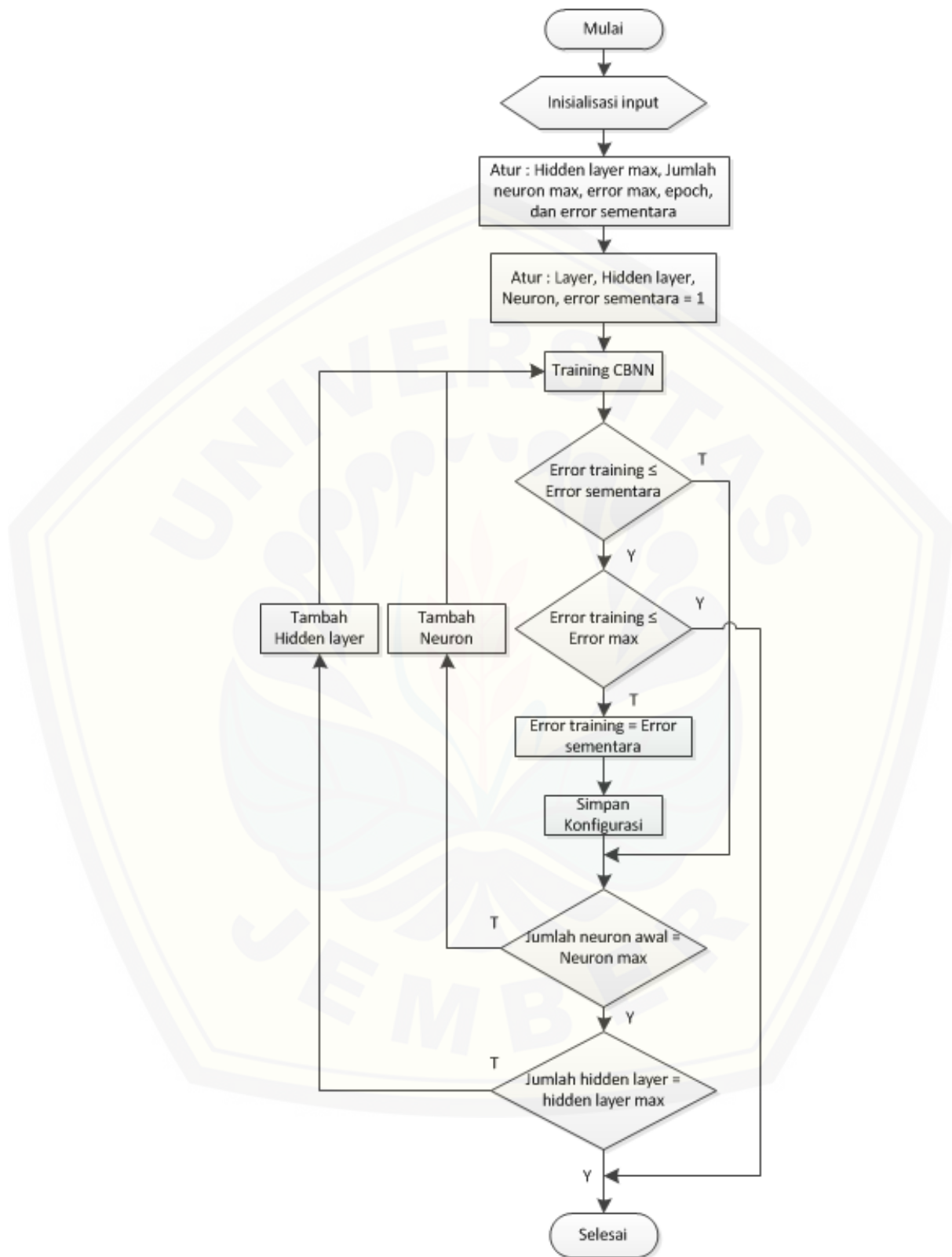
Tahap ketujuh adalah melakukan analisis pada data yang diperoleh dari tahap sebelumnya. Data yang diperoleh adalah data peramalan beban listrik dan bahan bakar mingguan.

8. Penarikan Kesimpulan

Tahap keenam adalah tahap terakhir pada penelitian ini yaitu memberikan kesimpulan dari pengujian dan analisa data yang diperoleh.

3.4 Perancangan Sistem *Training Neural Network*

Gambar 3.1 berikut ini merupakan diagram alir yang menunjukkan dan menjelaskan perancangan proses *training constructive backpropagation neural network* yang digunakan sebagai metode utama untuk melakukan peramalan beban dan bahan bakar gas pada PT PJB Gresik



Gambar 3.1 Diagram Alir Proses Simulasi

Proses dari diagram alir di atas dimulai dengan inisialisasi *input*, lalu mengatur jumlah *neuron* dan *hidden layer* awal yang digunakan pada program. Kemudian program di jalankan untuk mendapatkan hasil *training* yang diharapkan sesuai dengan target dan mendapatkan nilai *error* pada pelatihan tersebut, hasil *training* yang sesuai target bergantung pada nilai *error* dan pola yang didapatkan, semakin mendekati nilai *error* yang diinginkan maka hasil *training* semakin mendekati target.

Jika setelah proses *training* nilai *error* yang didapatkan sama dengan nilai *error* maksimal yang telah ditentukan maka pelatihan dilanjutkan dengan menambah jumlah *neuron*. Jika nilai *error* tidak sama dengan *error* maksimalnya, maka ditinjau terlebih dahulu nilai *error* yang didapat. Apabila *error* yang didapatkan nilainya lebih kecil dari *error* maksimal maka pelatihan selesai dan akan berhenti. Jika nilai *error* yang didapatkan lebih besar dari *error* maksimalnya maka akan dilanjutkan untuk menambah jumlah *neuron*.

Jika jumlah *neuron* tidak sama dengan jumlah *neuron* maksimalnya maka pelatihan akan kembali ke awal saat melakukan *training Constructive Backpropagation Neural Network* untuk mendapatkan nilai *error*, tetapi jika jumlah *neuron* sudah mencapai jumlah *neuron* maksimal maka proses *training* dilanjutkan dengan menambah jumlah *hidden layer*. Seperti saat menambah jumlah *neuron*, maka saat penambahan jumlah *hidden layer* tidak mencapai *hidden layer* maksimal maka pelatihan kembali ke awal untuk mendapatkan nilai *error*, tetapi jika jumlah *hidden layer* sama dengan jumlah *hidden layer* maksimal maka pelatihan akan berhenti dan diambil hasil yang terbaik dari pelatihan yang dilakukan. Hasil terbaik yang diambil lalu disimpan ialah hasil pelatihan dengan *error* terkecil beserta struktur yang terbentuk selama proses pelatihan *constructive backpropagation neural network*.

Constructive Backpropagation Neural Network (CBNN) digunakan pada simulasi ini dikarenakan CBNN memiliki keunggulan yaitu dalam sistemnya bisa memasukkan beberapa *hidden layer* dan *neuron* sesuai yang diinginkan dan program tersebut bisa dijalankan sekaligus tanpa harus mengganti satu persatu *hidden layer* dan neuronnya.

3.5 Perancangan Desain *Constructive Backpropagation Neural Network*

Berikut ini merupakan tabel yang berisikan perancangan arsitektur yang digunakan dalam proses *training constructive backpropagation neural network*.

Tabel 3.1 Arsitektur Jaringan Peramalan Beban Listrik

Arsitektur <i>Maximum</i> Jaringan <i>Constructive Backpropagation Neural Network</i>	
Epoch	: 1000
Jumlah Hidden Layer Max	: 3
Jumlah Neuron Hidden Layer Max	: 10
Fungsi Hidden Layer	: logsig
Fungsi Layer Output	: purelin
Fungsi Pelatihan	: trainr
Learning Rate	: 0.1

Pembuatan struktur dari *constructive backpropagation neural network* ke dalam program harus diatur terlebih dahulu, dalam hal ini pengaturannya adalah dengan memasukkan data yang sudah diurutkan sebagai *input* dan target, lalu penentuan banyaknya *hidden layer* ini berdasarkan pada *trial* dan *error* agar didapatkan jaringan yang terbaik dan sesuai untuk pelatihan data peramalan beban tenaga listrik ini.

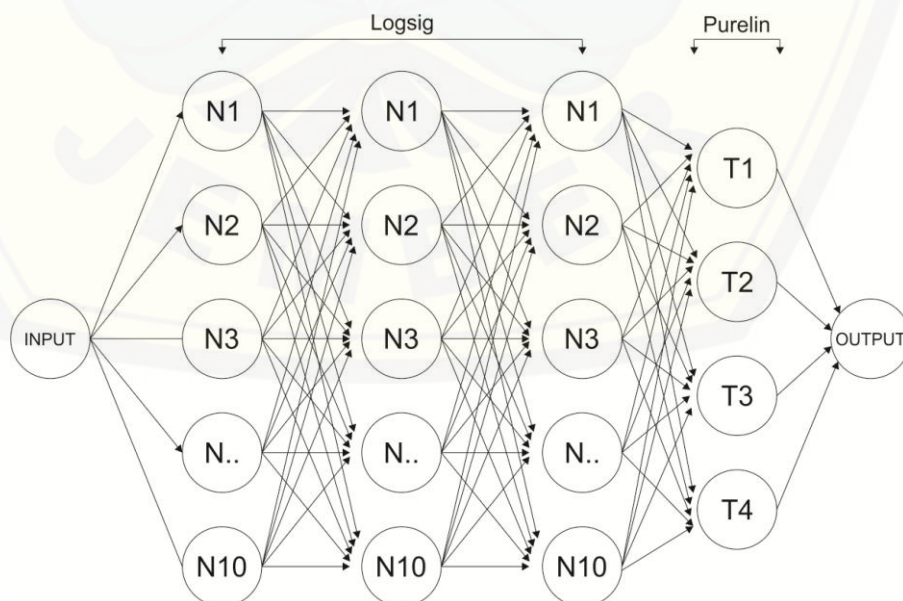
Berikut ini merupakan beberapa parameter pelatihan yang nantinya akan dijalankan agar mampu mendapatkan hasil yang optimal, parameter tersebut antara lain sebagai berikut :

1. *net.trainParam.show*: dipakai untuk menampilkan frekuensi perubahan mse (*default*: setiap 25 *epochs*), pada program yang dibuat frekuensi perubahannya setiap 50 *epoch*.
2. *net.trainParam.epochs* : dipakai untuk menentukan jumlah *epoch* maksimum pelatihan (*default*: 100 *epochs*), pada program yang dibuat untuk pelatihan pertama maksimum 1000 *epochs*.
3. *net.trainParam.goal*: dipakai untuk menentukan batas nilai mse agar iterasi dihentikan. Iterasi akan dihentikan jika mse lebih kecil dari batas

yang ditentukan dalam `net.trainParam.goal` atau jumlah *epoch* mencapai batas yang ditentukan dalam `net.trainParam.epochs`. Pada program pelatihan yang dijalankan, error yang ingin dicapai adalah 0.

4. `net.trainParam.lr` : dipakai untuk menentukan laju pemahaman ($\alpha = \text{learning rate}$). *Default* = 0.01. Semakin besar nilai α , semakin cepat pula proses pelatihan. Akan tetapi jika α terlalu besar, maka algoritma menjadi tidak stabil.

Data dan parameter pelatihan di atas digunakan untuk mendapatkan hasil yang baik, namun pola pembagian data pelatihan dan pengujian harus ditekankan agar jaringan mendapat data pelatihan yang cukup. Jumlah data yang kurang untuk proses pelatihan akan menyebabkan jaringan yang akan terbentuk mungkin tidak dapat mempelajari taburan data dengan baik. Sebaliknya, data yang terlalu banyak untuk proses pelatihan akan melambatkan proses pemusatan (konvergensi). Masalah *overtraining* (data pelatihan yang berlebihan) akan menyebabkan jaringan cenderung untuk menghafal data yang dimasukkan daripada melakukan generalisasi. Gambar 3.2 merupakan tampilan desain konfigurasi maksimal *hidden layer* dan *neuron* yang digunakan pada *neural network*.

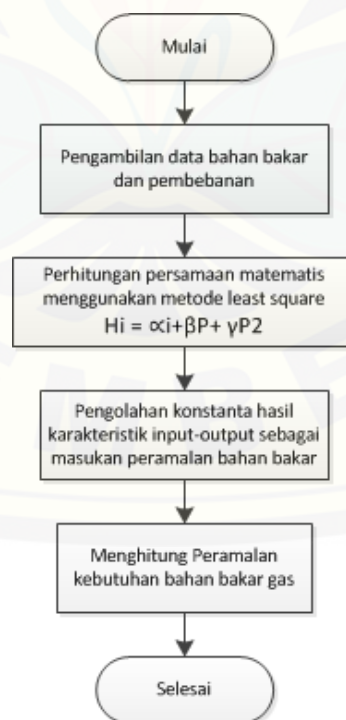


Gambar 3.2 Desain Konfigurasi *Maximum Neural Network*

Pada gambar 3.2 tersebut terdapat 48 *input* untuk pelatihan menggunakan *input* keseluruhan data, dan 24 *input* untuk korelasi data dengan masing – masing memiliki 4 buah target. pada sistem peramalan yang digunakan terdapat 4 buah *layer*, terdiri dari 3 buah *hidden layer* dan 1 *layer output*. Berdasarkan konfigurasi maksimal yang sudah dibentuk, yaitu menggunakan 3 buah *hidden layer* dengan 10 *neuron*, maka sistem akan mendapatkan 30 buah jaringan dalam sekali proses *training* yang selanjutnya akan dibandingkan pada setiap iterasinya untuk mendapatkan *error* yang terbaik.

3.6 Peramalan Kebutuhan Bahan Bakar Gas Menggunakan Karakteristik *Input Output*

Karakteristik *input–output* merupakan karakteristik yang ada pada tiap–tiap *unit* pembangkit yang menggambarkan hubungan antara daya dan bahan bakar. Untuk mengetahui karakteristik *input–output* ini dibutuhkan konstanta–konstanta formulasi konsumsi bahan bakar (α, β, γ).



Gambar 3.3 Alur peramalan bahan bakar gas

Langkah – langkah yang dilakukan yaitu :

1. Melakukan pengambilan data *input-output* pada pembangkit
2. Menentukan Konstanta (α, β, γ) dengan menghitung persamaan karakteristik *input-output* pembangkit dengan metode *quadratic least square* berdasarkan persamaan $H_i = \alpha_i + \beta P + \gamma P^2$.
3. Hasil konstanta karakteristik *input-output* yang sudah didapatkan dijadikan sebagai masukan untuk menentukan penggunaan bahan bakar gas berdasarkan hasil peramalan beban.
4. Menghitung kebutuhan bahan bakar gas berdasarkan hasil peramalan beban.

3.6 Alur Penelitian

Pada pembuatan simulasi peramalan beban tenaga listrik dan kebutuhan bahan bakar gas, proses pembuatan simulasinya dirancang sesuai dengan diagram alir yang ada pada Gambar 3.5 dibawah ini.



Gambar 3.4 Alur Penelitian

BAB 5. PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dan analisis data pada bab sebelumnya maka dalam penelitian ini dapat disimpulkan bahwa :

1. Peramalan jangka pendek beban tenaga listrik menggunakan metode *constructive backpropagation neural network* tahun 2016 mendapatkan hasil rata-rata *error* sebesar 2,07%. *Error* terkecil sebesar 0,2% dengan hasil peramalan 21024670 Watt pada bulan Oktober Minggu ketiga. *Error* terbesar dengan nilai 4,9% dengan hasil peramalan 18921563 Watt pada bulan September minggu pertama.
2. Peramalan bahan bakar gas menggunakan metode *quadratic least square* mendapatkan hasil rata-rata *error* sebesar 0,6%. *Error* tekecil sebesar 0,6% dengan hasil *flowrate* 23,7 KNM³/H pada bulan Agustus minggu kedua. *Error* terbesar dengan nilai 4,4% dengan hasil *flowrate* 24,3 KNM³/H pada bulan November minggu pertama.
3. Konfigurasi *neural network* terbaik yang digunakan sebagai sistem peramalan ialah berhenti pada *hidden layer* ke-2 dan *neuron* = 10 dengan *error* sebesar 3,28e-03, fungsi pelatihan menggunakan *trainr*, *logsig* untuk *layer input*, dan *purelin* untuk *layer output*.

5.2 Saran

Saran yang dianjurkan untuk pengembangan penelitian lebih lanjut dengan harapan mampu memperbaiki kekurangan dan mendapatkan hasil peramalan dengan *error* yang lebih kecil lagi adalah sebagai berikut:

1. Mempertimbangkan parameter lain sebagai input pembelajaran seperti perubahan dan pergeseran beban puncak dan penambahan jumlah penduduk dan industri atau pembangunan.
2. Sebagai pembanding bisa digunakan metode lain dalam peramalan beban



DAFTAR PUSTAKA

- Pratama, Ricky Ardian. 2016. "Peramalan Beban Listrik Jangka Panjang Provinsi D.I. Yogyakarta Menggunakan *Neural Network Back Propagation*" Universitas Negeri Surabaya. JTE. Vol. 05. No. 03.
- Binoto, Maju. 2015. "Peramalan Energi Listrik Yang Terjual dan Daya Listrik Tersambung Pada Sistem Ketenagalistrikan Untuk Jangka Panjang Di Solo Menggunakan Model Artificial Neural Network". Akademi Teknologi Solo. SNATIF Vol. 2.
- Rolyadely, Rendy. 2014. "Prediksi Pemakaian Listrik Dengan Pendekatan *Back Propagation*". STMIK Banjarbaru. JUTISI Vol. 3. No. 1
- Dwisatya, Ramadani. 2014. "Prediksi Beban Listrik Jangka Pendek Berbasis Algoritma Feed Forward Backpropagation Dengan Mempertimbangkan Variasi Tipe Hari". Universitas Telkom.
- Toreh, G. E. J. 2013. "Peramalan Beban Listrik Untuk Penjadwalan Sistem Pembangkit". UNSRAT.
- Handayani, Irma. Alimudin, Suhendar. 2012. "Peramalan Beban Tenaga Listrik Jangka Pendek Menggunakan Metode Jaringan Syaraf Tiruan". UNTIRTA. Vol. 1, No. 1.
- Sari, Dinar Atika. "Peramalan Kebutuhan Beban Jangka Pendek Menggunakan Jaringan Syaraf Tiruan *Back Propagation*". Universitas Diponegoro.
- Muslimin. 2015 "Peramalan Beban Listrik Jangka Menengah Pada Sistem Kelistrikan Kota Samarinda". Universitas Mulawarman. JITI Vol. 14. No. 2.
- Anggraeni, Rina. 2016. "Analisis Karakteristik *Input-Output* dan Optimasi Biaya Pembangkitan Menggunakan Metode *Quadratic Least Square Regression* dan Metode *Dynamic Genetic Algorithm*". Universitas Jember.
- Larasati, Bunga Fitriana. 2014. "Analisis Kualitas Daya Dengan Pemasangan D-Statcom Menggunakan Kendali *Constructive Backpropagation Neural Network* Pada Sistem Distribusi 20kV APJ PLN Jember. Universitas Jember.
- Daman Suswanto. 2009. Sistem Distribusi Tenaga Listrik. Universitas Negeri Padang.

LAMPIRAN

A. Listing Program Constructive Backpropagation Neural Network

```

clear;
clc;
load input.mat;

%% Preprocessing
P=watt(:,1:48)';
T=watt(:,49:52)';

[pn,meanp,stdp,tn,meant,stdt] = prestd(P,T);
%% Define Parameters
max_hl1 = 10;
max_hl2 = 10;
max_hl3 = 10;
error = 1;
errormax = 0;
modmse = 2e2;
epochs = 1000;

% Pelatihan hidden layer 1
for l1=1:max_hl1;
    if error > errormax ;
        net=newff(minmax(pn),[l1 4], {'logsig','purelin'});

        %parameter
        net.trainFcn = 'trainr';
        net.trainParam.lr = 0.1;
        net.trainParam.epochs = epochs;
        net.trainParam.goal = errormax;
        net.trainParam.show = 50;
        net.trainParam.max_fail = 200;
        net.trainParam.mc = 0.5;
        net.performFcn = 'mse';

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

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

        %Simulasi Hasil
        Wr= sim(netWr,pn);

        %Hitung mse
        error=mse(Wr-tn);
        %error(l1)=error; %cek
        %Ambil Network yg terbaik
    end
end

```

```
if modmse > error;
    layerWr=11;
    modmse=error;
    hl=1;
    save training.mat;
end
else
    layer1=11;
    break;
end
end

% Pelatihan hidden layer 2
for l2=1:max_hl2;
    if error > errormax ;
        net=newff(minmax(pn),[l1 l2 4], {'logsig','logsig','purelin'});

        %parameter
        net.trainFcn = 'trainr';
        net.trainParam.lr = 0.1;
        net.trainParam.epochs = epochs;
        net.trainParam.goal = errormax;
        net.trainParam.show = 50;
        net.trainParam.max_fail = 200;
        net.trainParam.mc = 0.5;
        net.performFcn = 'mse';

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

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

        %Simulasi Hasil
        Wr= sim(netWr,pn);

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

```

end

% Pelatihan hidden layer 3
for l3=1:max_hl3;
    if error_ > errormax;
        net=newff(minmax(pn),[l1 l2 l3 4],
{'logsig','logsig','logsig','purelin'});

        %parameter
        net.trainFcn = 'trainr';
        net.trainParam.lr = 0.1;
        net.trainParam.epochs = epochs;
        net.trainParam.goal = errormax;
        net.trainParam.show = 50;
        net.trainParam.max_fail = 200;
        net.trainParam.mc = 0.5;
        net.performFcn = 'mse';

        %Menampilkan Jumlah Neuron
        disp([' Hiden Layer III, Jumlah Neuron      =      '
num2str(l3)]);

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

        %Simulasi Hasil
        Wr= sim(netWr,pn);

        %Hitung mse
        error=mse(Wr-tn);
        %error(l1)=error; %cek
        %Ambil Network yg terbaik
        if modmse > error;
            layerWr=l3;
            modmse=error;
            hl=3;
            save training.mat;
        end
        else
            layer3=l3;
            break;
        end
    end

end

%% Mengambil network yg terbaik
load training.mat;
an = sim(netWr,pn)
bs= [layerWr modmse]
a = poststd(an,meant,stdt);
H = [(1:size(P,2))' T' a' (T'-a')];

```

```
sprintf('%2d %9.2f %7.2f %5.2f\n',H');
disp('Network terbaik dengan error terkecil');
disp(['Hidden Layer ke ' num2str(hl),', Neuron = '
num2str(layerWr),', Error = ' num2str(modmse)]);

%% Plot
[m1,a1,r1] = postreg(a,T);
plot([1:size(P,2)]',T,'bo',[1:size(P,2)]',a','r*');
title('hasil pengujian dengan data pelatihan: Target (o),
Output(*)');
xlabel('data ke-'); ylabel('T/o');

%% Transform ke simulink
%gensim(netWr)

%Pengujian
%qn = trastd(q,meanp, stdp);
%bn = sim(net,qn);
%b = poststd(bn,meant, stdt);

%% load bobot
disp('Layer 1');
IW11=netWr.IW{1,1}
LW11=netWr.LW{1,1}
b1=netWr.b{1}

disp('Layer 2');
LW21=netWr.LW{2,1}
LW22=netWr.LW{2,2}
b2=net.b{2}

disp('Layer 3');
IW31=netWr.IW{3,1}
LW32=netWr.LW{3,2}
b3=net.b{3}
```

B. Listing Program Quadratic Least Square

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

x = [...
    str2num(get(handles.edit1,'String'))
    str2num(get(handles.edit2,'String'))
    str2num(get(handles.edit3,'String'))
    str2num(get(handles.edit4,'String'))
    str2num(get(handles.edit5,'String'))
    str2num(get(handles.edit6,'String'))
    str2num(get(handles.edit7,'String'))
    str2num(get(handles.edit8,'String'))
    str2num(get(handles.edit9,'String'))
    str2num(get(handles.edit10,'String'))
    str2num(get(handles.edit11,'String'))
    str2num(get(handles.edit12,'String'))
    str2num(get(handles.edit13,'String'))
    str2num(get(handles.edit14,'String'))
    str2num(get(handles.edit15,'String'))
    str2num(get(handles.edit16,'String'))
    str2num(get(handles.edit17,'String'))
    str2num(get(handles.edit18,'String'))];

y = [...
    str2num(get(handles.bahanbkr1,'String'))
    str2num(get(handles.bahanbkr2,'String'))
    str2num(get(handles.bahanbkr3,'String'))
    str2num(get(handles.bahanbkr4,'String'))
    str2num(get(handles.bahanbkr5,'String'))
    str2num(get(handles.bahanbkr6,'String'))
    str2num(get(handles.bahanbkr7,'String'))
    str2num(get(handles.bahanbkr8,'String'))
    str2num(get(handles.bahanbkr9,'String'))
    str2num(get(handles.bahanbkr10,'String'))
    str2num(get(handles.bahanbkr11,'String'))
    str2num(get(handles.bahanbkr12,'String'))
    str2num(get(handles.bahanbkr13,'String'))
    str2num(get(handles.bahanbkr14,'String'))
    str2num(get(handles.bahanbkr15,'String'))
    str2num(get(handles.bahanbkr16,'String'))
    str2num(get(handles.bahanbkr17,'String'))
    str2num(get(handles.bahanbkr18,'String'))];

N = [...
    str2num(get(handles.edit40,'String'))];
```

```

%% Sx dan Sy
Sx=sum(x);
Sy=sum(y);

%% Sx^2
for i=1:N
    a(i)=x(i)*x(i);
    b(i)=x(i)*y(i);
    c(i)=x(i)*x(i)*x(i);
    d(i)=x(i)*x(i)*x(i)*x(i);
    e(i)=x(i)*x(i)*y(i);
end
Sx2=sum(a);
Sxy=sum(b);
Sx3=sum(c);
Sx4=sum(d);
Sx2y=sum(e);

%% Mencari Sxx Sxy dll
Sxx=Sx2-(Sx^2/N);
Sxy=Sxy-(Sx*Sy/N);
Sxx2=Sx3-(Sx*Sx2/N);
Sx2y=Sx2y-(Sx2*Sy/N);
Sx2x2=Sx4-(Sx2*Sx2/N);

%% Koefisien a, b, dan c
a=((Sx2y*Sxx)-(Sxy*Sxx2))/((Sxx*Sx2x2)-Sxx2*Sxx2);
b=((Sxy*Sx2x2)-(Sx2y*Sxx2))/((Sxx*Sx2x2)-Sxx2*Sxx2);
c=Sy/N - (b*(Sx/N))-(a*(Sx2/N));

fprintf('Persamaannya QSLR : %.4f x^2 + %.4f x + %.4f\n', a,b,c);
set(handles.hasil1,'String',a)
set(handles.hasil2,'String',b)
set(handles.hasil3,'String',c)

%% Ploting
axes(handles.axes1)

%Data
%figure(1)
hold on
h1=scatter(x , y , 50,'r','filled');
hold off

%Hasil Regresi
p=[a b c];
x=linspace(100,200);
y=polyval(p,x);

```



```
plot(x,y)
xlabel('Daya (MW)');
ylabel('Bahan Bakar (KNM3/jam)');

%Tombol RESET
% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.edit1, 'string', '');
set(handles.edit2, 'string', '');
set(handles.edit3, 'string', '');
set(handles.edit4, 'string', '');
set(handles.edit5, 'string', '');
set(handles.edit6, 'string', '');
set(handles.edit7, 'string', '');
set(handles.edit8, 'string', '');
set(handles.edit9, 'string', '');
set(handles.edit10, 'string', '');
set(handles.edit11, 'string', '');
set(handles.edit12, 'string', '');
set(handles.edit13, 'string', '');
set(handles.edit14, 'string', '');
set(handles.edit15, 'string', '');
set(handles.edit16, 'string', '');
set(handles.edit17, 'string', '');
set(handles.edit18, 'string', '');

set(handles.bahanbkr1, 'string', '');
set(handles.bahanbkr2, 'string', '');
set(handles.bahanbkr3, 'string', '');
set(handles.bahanbkr4, 'string', '');
set(handles.bahanbkr5, 'string', '');
set(handles.bahanbkr6, 'string', '');
set(handles.bahanbkr7, 'string', '');
set(handles.bahanbkr8, 'string', '');
set(handles.bahanbkr9, 'string', '');
set(handles.bahanbkr10, 'string', '');
set(handles.bahanbkr11, 'string', '');
set(handles.bahanbkr12, 'string', '');
set(handles.bahanbkr13, 'string', '');
set(handles.bahanbkr14, 'string', '');
set(handles.bahanbkr15, 'string', '');
set(handles.bahanbkr16, 'string', '');
set(handles.bahanbkr17, 'string', '');
set(handles.bahanbkr18, 'string', '');

set(handles.edit40, 'string', '');
```

```
set(handles.hasil1,'string',num2str(0))
set(handles.hasil2,'string',num2str(0))
set(handles.hasil3,'string',num2str(0))

cla(handles.axes1,'reset')

% --- Executes during object creation, after setting all properties.
function pushbutton1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% --- Executes during object deletion, before destroying properties.
function pushbutton1_DeleteFcn(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

C. Data Beban dan Bahan Bakar PT PJB Gresik

No	Bulan	Minggu	Beban (Watt)			Bahan Bakar Gas (KNM3/H)		
			2014	2015	2016	2014	2015	2016
1	Januari	1	22388613	22388613.43	21028000	5102.204	5102.204	4614.06
		2	24290003	24290003.47	22120000	5584.337	5584.337	4821.42
		3	24647993	24647992.98	19982000	5629.581	5629.581	4387.72
		4	24135466	24135466.03	18080000	5474.873	5474.873	3863.768
2	Februari	5	25999890	19721444.38	18104000	5308.63	5138.293	3971.06
		6	25175901	22361826.97	18248000	5775.866	5280.263	4014.2
		7	24834998	22633190.42	19200000	5706.892	4735.774	4173.52
		8	24445943	22553090.67	19419800	5646.964	4712.563	3508.56
3	Maret	9	24945417	21903198.34	17296000	5730.62	5098.102	3931.22
		10	25623097	19780114.13	17846000	5935.575	5249.366	4373.88
		11	25203207	22457162.03	17212000	5831.803	5053.288	4219.12
		12	25430699	20170733.09	17334000	5892.82	5173.981	4178.6
4	April	13	23455221	19386000	17900000	5417.688	4497.771	3951.08
		14	23273731	18944000	16630000	5348.939	4620.88	3607.58
		15	22156582	19540000	16906000	5068.692	4311.12	3435.58
		16	22132871	19185200	16848000	5087.628	4232.841	3265.28
5	Mei	17	22212815	19588200	18092000	5086.237	4101.124	3962.1
		18	23967992	19615800	17570000	5469.096	4324.492	3592.068
		19	22644740	19401600	18010000	5242.583	4058.219	3819.66
		20	22636184	19401100	16870000	5225.856	4287.49	2949.5
6	Juni	21	21930047	19694720	16914200	4997.225	4342.557	3678.04
		22	23232927	18604000	17164000	5284.789	3784.68	3660.58
		23	21940912	19688000	16800000	5070.422	4194.892	3664.82
		24	20773093	19402000	17072800	4791.66	4156.18	3711.56
7	Juli	25	22631395	20180000	19314000	5084.275	4303.58	3726.56
		26	22351503	19256000	18003600	5024.99	3400.408	3653.56
		27	21972581	21710000	19802000	5004.18	4590.56	3704
		28	19364177	18482000	17129400	2073.081	2212.089	4297.76
8	Agustus	29	23219191	20986000	18776200	5237.375	5041.58	3199.62
		30	22842653	18316000	18217000	5137.405	3903.634	3483.12
		31	23219191	19934000	19970000	5237.375	4003.32	3744.9
		32	22842653	21946000	17856000	5137.405	4700.32	3848.06
9	September	33	23557736	21412000	18024000	5277.928	4525.42	3951.98
		34	21228418	17422000	17718000	4871.473	3673.599	3653.22

		35	21325821	19546000	20100000	4990.951	3676.554	3646.86
		36	15852171	16137800	17288000	4242.357	3372.394	4716.12
10	Oktober	37	22854560	20114000	20134000	5281.725	4172.864	4298.12
		38	24368369	21365600	18650800	5598.048	4493.843	3413.36
		39	26427368	20107600	21086000	5990.402	4908.82	4303.34
		40	24175898	21700000	17970000	5551.358	4610.98	3878.66
11	November	41	21203147	20628000	18684000	4927.15	4398.38	4135.1
		42	23631165	20476000	18475800	5505.592	4125.077	3560.92
		43	20699511	19582000	19324000	4852.418	4525.16	3642.44
		44	23148467	19268000		5468.892	4398.36	
12	Desember	45	20551141	18429000		4970.056	5858.178	
		46	24054047	17998000		5544.549	5058.18	
		47	21730702	18578000		5086.931	4363.57	
		48	20881958	18044000		4881.455	4259.08	

D. Robot Layer 1

No	Neuron									
	1	2	3	4	5	6	7	8	9	10
1	0.015	-0.26	0.047	-0.15	0.383	-0.03	0.240	0.224	-0.17	-0.20
2	0.256	0.465	0.137	-0.18	0.661	-0.26	0.232	0.332	0.508	-0.34
3	-0.01	-0.54	0.293	0.401	0.331	-0.16	0.19	0.499	0.465	-0.35
4	0.203	-0.14	-0.27	0.456	-0.33	0.292	0.468	0.181	0.572	0.448
5	0.179	-0.59	-0.03	0.248	-0.45	0.599	-0.01	-0.18	0.457	0.035
6	0.138	-0.26	-0.60	-0.53	0.105	0.281	0.099	-0.32	0.089	0.207
7	0.372	-0.14	0.172	-0.09	0.112	-0.05	0.106	-0.05	0.281	-0.07
8	0.059	-0.23	0	-0.28	-0.38	0.137	-0.07	-0.39	0.663	0.005
9	0.328	-0.01	-0.12	0.028	0.488	0.220	0.004	0.150	0.054	0.149
10	-0.37	-0.23	-0.46	0.064	0.140	0.126	-0.05	0.092	0.391	-0.01
11	-0.36	-0.20	-0.18	0.226	0.136	-0.04	-0.11	0.176	0.454	0.016
12	0.028	-0.02	-0.08	-0.34	0.150	0.201	0.332	0.015	0.033	0.300
13	0.525	-0.18	0.369	0.052	0.137	0.110	-0.09	-0.15	0.476	-0.15
14	-0.21	-0.46	-0.11	0.151	0.238	-0.13	-0.22	-0.00	0.348	0.239
15	0.030	-0.04	-0.18	0.233	0.265	-0.22	0.181	0.279	0.178	0.308
16	-0.16	-0.24	-0.51	-0.28	0.102	0.248	-0.28	0.041	0.058	-0.28
17	0.403	0.048	0.147	-0.12	-0.21	-0.09	-0.28	0.032	-0.22	0.044
18	0.074	-0.11	-0.06	0.096	0.101	-0.12	-0.19	0.036	-0.04	0.117
19	-0.17	0.320	-0.50	0.133	-0.08	-0.21	-0.13	-0.05	-0.13	0.433

20	-0.19	-0.18	-0.12	-0.00	0.025	0.073	-0.29	0.073	0.001	-0.19
21	-0.21	-0.00	0.043	-0.16	0.037	-0.29	0.157	-0.11	0.331	-0.22
22	0.062	-0.09	0.165	0.078	-0.31	-0.26	0.329	0.077	0.028	0.116
23	0.415	-0.00	0.274	0.649	0.193	-0.13	0.647	-0.26	-0.27	-0.08
24	0.233	0.540	-0.35	-0.05	0.011	-0.06	-0.15	0.092	-0.22	0.118
25	-0.31	0.212	0.070	-0.58	-0.20	0.352	-0.47	0.177	-0.31	0.068
26	-0.49	0.116	-0.63	0.206	0.209	-0.09	-0.14	0.429	0.331	-0.11
27	-0.21	-0.46	-0.11	0.151	0.238	-0.13	-0.22	-0.00	0.348	0.239
28	0.030	-0.04	-0.18	0.233	0.265	-0.22	0.181	0.279	0.178	0.308
29	-0.16	-0.24	-0.51	-0.28	0.102	0.248	-0.28	0.041	0.058	-0.28
30	0.403	0.048	0.147	-0.12	-0.21	-0.09	-0.28	0.032	-0.22	0.044
31	0.074	-0.11	-0.06	0.096	0.101	-0.12	-0.19	0.036	-0.04	0.117
32	-0.17	0.320	-0.50	0.133	-0.08	-0.21	-0.13	-0.05	-0.13	0.433
33	-0.19	-0.18	-0.12	-0.00	0.025	0.073	-0.29	0.073	0.001	-0.19
34	-0.21	-0.00	0.043	-0.16	0.037	-0.29	0.157	-0.11	0.331	-0.22
35	0.062	-0.09	0.165	0.078	-0.31	-0.26	0.329	0.077	0.028	0.116
36	0.415	-0.00	0.274	0.649	0.193	-0.13	0.647	-0.26	-0.27	-0.08
37	0.233	0.540	-0.35	-0.05	0.011	-0.06	-0.15	0.092	-0.22	0.118
38	-0.31	0.212	0.070	-0.58	-0.20	0.352	-0.47	0.177	-0.31	0.068
39	-0.49	0.116	-0.63	0.206	0.209	-0.09	-0.14	0.429	0.331	-0.11
40	-0.13	0.462	0.118	-0.17	-0.22	0.121	0.007	0.199	-0.36	-0.08
41	-0.25	-0.12	0.272	-0.00	0.038	0.089	-0.25	0.406	-0.11	-0.19
42	-0.10	-0.10	0.054	0.334	-0.22	0.028	-0.03	0.027	0.527	0.123
43	-0.08	-0.33	0.185	0.233	-0.17	0.482	-0.27	-0.06	0.099	-0.20
44	0.27	-0.13	-0.29	-0.29	0.468	-0.39	0.297	-0.23	-0.34	-0.44
45	0.006	0.023	0.305	-0.44	-0.63	0.008	-0.69	-0.18	0.055	-0.11
46	-0.14	-0.17	-0.06	0.520	0.035	-0.28	-0.01	-0.06	0.050	0.325
47	0.497	0.124	0.279	0.034	0.249	0.130	0.575	-0.11	-0.13	0.012
48	0.011	0.122	0.843	-0.45	0.227	-0.02	0.330	-0.38	0.442	0.224

E. Bobot Layer 2

No	Neuron									
	1	2	3	4	5	6	7	8	9	10
1	0.456	3.701	2.574	0.322	-0.11	0.466	2.094	3.158	-1.41	-2.24
2	-0.83	-0.75	1.538	-3.89	1.654	0.854	2.363	1.506	0.117	2.61
3	1.759	1.469	-0.77	-1.35	-1.73	-3.43	1.129	0.554	-0.09	-3.12
4	-3.48	-0.61	-2.43	1.183	0.635	-2.56	-1.97	0.072	1.090	-1.85
5	1.346	2.608	1.149	-2.92	-3.75	2.930	3.651	2.882	4.846	-3.27
6	-2.83	2.880	0.504	1.222	1.670	0.642	0.863	-3.24	-1.40	0.382
7	-2.54	-1.70	-4.47	-2.91	1.374	2.956	-2.20	-0.58	1.021	1.072
8	3.190	1.778	-3.58	-3.43	1.666	0.218	-1.50	2.731	2.891	-2.50
9	2.473	-2.27	-0.62	0.694	4.563	3.279	-1.07	2.650	4.172	-2.14
10	2.610	2.853	0.776	0.897	0.963	-2.55	-3.82	1.944	0.047	-0.72

F. Bobot layer Output

No	1	2	3	4
1	0.5861	2.0635	1.7873	-1.1053
2	2.7592	-0.0655	1.1276	0.6045
3	-0.1648	-1.301	-0.375	-1.086
4	-0.5064	-0.83	0.2144	0.982
5	-0.3489	1.7192	1.5997	0.9415
6	1.049	1.8359	-1.0145	2.6213
7	1.4481	1.0712	-0.5681	-0.7384
8	-1.0338	-0.2927	-0.2411	-1.0138
9	0.6363	-1.3049	1.8858	0.6651
10	-0.962	0.1748	-0.9931	0.26

G. Bias Layer

No	Bias layer 1	Bias layer 2	Bias layer Output
1	2.5026	-3.9441	-0.3788
2	1.9852	-8.0519	-0.7238
3	1.3616	0.7079	0.026
4	-0.5425	3.9765	-0.0141
5	-0.8627	-3.9934	
6	-0.6429	-1.1394	
7	1.0422	0.6988	
8	1.3073	-4.0359	
9	-2.556	-8.0264	
10	2.9887	2.7878	

H. Gambar Hasil Percobaan

