

# Perancangan Sistem Prediktor Daya Pada Panel *Photovoltaic* di *Buoy Weather Station*

Aini Prisilia Susanti dan Aulia Siti Aisjah

Jurusan Teknik Fisika FTI ITS Surabaya Kampus ITS Keputih Sukolilo Surabaya 6011

Telp : +6231-5947188 Fax : +6231-5923626

E-mail: auliasa@ep.its.ac.id

**Abstrak**—*Buoy weather station* merupakan stasiun informasi cuaca yang banyak dijumpai di pelabuhan, khususnya di Surabaya. Untuk mengoperasikannya diperlukan sumber daya listrik berupa panel *photovoltaic*. Efek fotolistrik pada PV mampu merubah energi cahaya menjadi energi listrik. Besarnya daya yang dihasilkan tergantung dari intensitas matahari, temperatur permukaan, dan keadaan geografis setempat. Untuk memprediksi daya keluaran per setengah jam yang dihasilkan oleh panel PV maka digunakan metode jaringan syaraf tiruan dengan algoritma *backpropagation* pada *software* Matlab. Variabel yang digunakan berupa data daya yang diperoleh dari tegangan dan arus yang dihasilkan oleh panel PV di daerah Surabaya. Data daya selama 3 hari per setengah jam tersebut dijadikan data input dan target pada Matlab. Hasil terbaik perancangan sistem prediksi daya keluaran panel PV menggunakan JST pada Matlab yaitu *Mean Square Error* (MSE) sebesar 0,0113 dan akurasi ketepatan prediksi sebesar 99,81%.

**Kata kunci**—*buoy weather station*, panel *photovoltaic*, jaringan syaraf tiruan, algoritma *backpropagation*, Matlab, GUI.

## I. PENDAHULUAN

LAUTAN di Indonesia memiliki peranan penting dalam kehidupan kita sehari-hari mengingat negara ini dikenal sebagai negara maritim. Kita membutuhkan nelayan untuk mencari ikan, transportasi laut untuk menghubungkan satu pulau dengan pulau yang lain, dan sumber daya alam yang terdapat di dasar laut yang kekayaannya tak terbatas. Namun tak jarang, laut menjadi lawan pada saat terjadi pasang bahkan tsunami yang sempat merenggut nyawa sebagian besar warga Aceh dan sekitarnya. Selama ini Indonesia memiliki suatu badan yang berfungsi sebagai pemantau kondisi keadaan laut yaitu Badan Meteorologi Klimatologi dan Geofisika (BMKG). BMKG memiliki stasiun cuaca dengan melakukan pemantauan seperti temperatur, kelembaban udara, tekanan udara, curah hujan, arah angin, dan intensitas matahari dimana didukung oleh sensor yang mampu memberikan informasi dan prediksi cuaca di lautan [1].

Dengan menggunakan perangkat atau *device* berupa *buoy weather station*, maka pencarian data cuaca yang dibutuhkan dapat dilakukan. *Buoy weather station* dapat terapung layaknya pelampung dengan sumber daya listrik berupa panel surya. Panel surya atau yang seringkali disebut sel *photovoltaic*, mampu menangkap energi matahari yang dikenal sebagai sumber cahaya terkuat yang dapat dimanfaatkan. Alat ini terdiri dari sel surya yang mengubah energi cahaya (foton) menjadi energi listrik [2]. Adapun karakteristik dari daya yang dihasilkan oleh solar sel ditentukan oleh intensitas cahaya,

temperatur matahari, dan faktor geografis dari suatu daerah [3]. Semakin banyak intensitas cahaya, semakin banyak energi matahari yang diserap, maka semakin tinggi pula arus yang dihasilkan sehingga diperoleh daya keluaran yang maksimum. Untuk mengetahui seberapa besar daya yang dibutuhkan untuk menjalankan *buoy weather*, maka dibutuhkan sistem prediksi daya keluaran yang dihasilkan oleh panel *photovoltaic*.

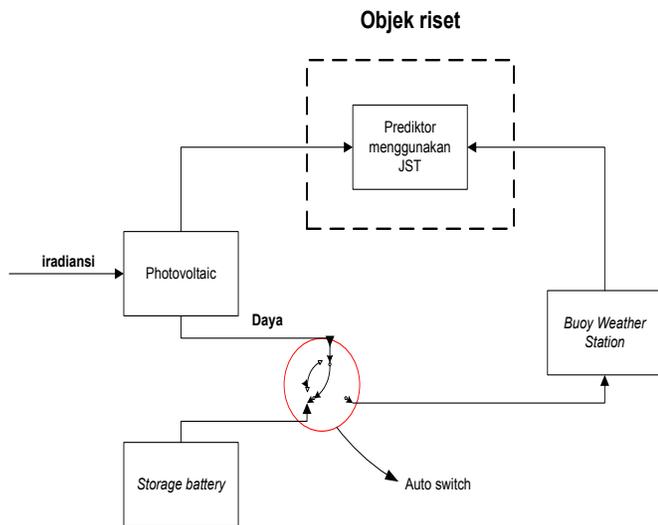
Berdasarkan teori yang telah disebutkan dalam jurnal-jurnal tentang *photovoltaic*, ada beberapa faktor yang mempengaruhi keluaran dari modul PV. Salah satu faktor utamanya yaitu besarnya iradiasi atau intensitas cahaya matahari yang diterima oleh permukaan modul PV. Banyaknya intensitas matahari yang ditangkap tergantung dari keadaan cuaca pada saat itu dan peletakkan modul *photovoltaic* sehingga mempengaruhi besarnya daya keluaran. Dimana semakin tinggi intensitas maka daya yang dihasilkan akan sebanding berubah yaitu semakin besar. Oleh sebab itu, pada tugas akhir ini dirancang suatu sistem yang berfungsi sebagai prediktor daya keluaran pada panel *photovoltaic* berdasarkan banyaknya intensitas matahari. Metode yang digunakan untuk perancangan sistem prediksi yaitu jaringan syaraf tiruan (JST) atau yang sering disebut *Artificial Neural Network* (ANN) dengan algoritma *backpropagation*. Dengan demikian, sistem prediktor daya keluaran pada panel *photovoltaic* dapat di aplikasikan pada *buoy weather station*.

## II. METODE PENELITIAN

### A. Alur Kinerja Sistem

Dalam pengerjaan suatu penelitian diperlukan diagram blok sistem sebagai gambaran umum tentang permasalahan dan tujuan yang diinginkan. Diagram blok disusun berdasarkan langkah-langkah dari tugas akhir yang bertujuan menjelaskan keseluruhan komponen sampai objek yang dijadikan suatu riset. Diagram blok sistem umumnya memiliki variabel input, proses, dan variabel output disertai dengan parameter-parameter pendukungnya.

Komponen utama yang digunakan dalam tugas akhir ini adalah modul *photovoltaic*. *Photovoltaic* mampu mengkonversi energi foton dari cahaya matahari menjadi energi listrik. Teknologi PV mengandalkan efek fotolistrik pertama kali di jelaskan oleh ilmuwan asal Prancis, Edmund Becquerel, pada tahun 1839 [2]. Gambar 1 merupakan diagram blok sistem dari tugas akhir ini yaitu sebagai berikut :



Gambar 1. Diagram blok sistem

Besarnya daya yang dihasilkan oleh modul PV dipengaruhi intensitas matahari atau yang sering disebut iradiansi. Dalam hal ini iradiansi berperan sebagai variabel input dan daya sebagai variabel output. Sedangkan peneliti melakukan riset pada perancangan sistem prediktor daya menggunakan metode jaringan syaraf tiruan. Kemampuan modul PV dalam memenuhi daya yang dibutuhkan oleh *buoy weather station* sangat penting, disini peneliti mengasumsikan *buoy weather station* yang digunakan membutuhkan daya sebesar 10 watt sebagai catu daya supaya dapat beroperasi. Oleh sebab itu, sistem prediktor ini bertujuan meng-informasikan seberapa besar daya yang dihasilkan modul PV untuk setengah jam selanjutnya. Jika daya pada PV tidak mampu memenuhi kebutuhan *buoy weather station*, maka *storage battery* akan menggantikan fungsi PV sebagai *power supply*.

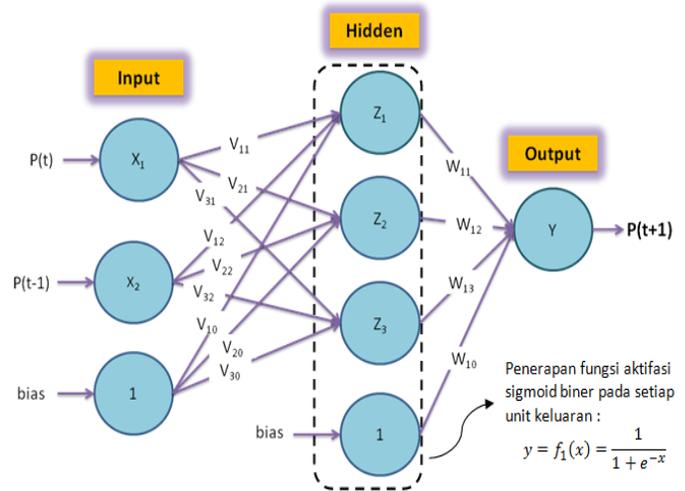
**B. Penyusunan Algoritma Backpropagation**

Ada beberapa tahap yang perlu diperhatikan sebelum dilakukan pemodelan :

1. Penentuan data input dan output untuk pemahaman *backpropagation*.
2. Penentuan jumlah *layer* atau lapisan pada arsitektur *backpropagation*.

Maka dapat ditentukan pemodelan dari jaringan *backpropagation* dengan data input berupa daya pada saat waktu tertentu  $P(t)$  dan  $P(t-1)$ , dan output yang diinginkan berupa  $P(t+1)$ . Arsitektur *backpropagation* sesuai dengan yang diinginkan dapat di modelkan pada Gambar 2.

Dalam suatu jaringan syaraf tiruan terdapat nilai bobot dan bias. Bobot dan bias merupakan kemampuan suatu jaringan dalam pelatihan yang berpengaruh terhadap output yang dihasilkan. Jumlah bobot dipengaruhi oleh banyaknya jumlah node atau neuron pada *hidden layer*. Semakin banyak jumlah neuron dalam pelatihan maka semakin banyak pula bobot-bobot yang dihasilkan.



Gambar 2. Tampilan arsitektur *backpropagation* dalam penelitian.

Gambar 2 merupakan arsitektur *backpropagation* yang digunakan dalam penelitian kali ini. Terdapat 2 jenis bobot pada arsitektur tersebut yaitu bobot dari input menuju *hidden layer* biasa disebut *input weight* (IW) yang ditunjukkan pada gambar yaitu  $V_{11}, V_{21}, V_{31}, V_{12}, V_{22}, V_{32}$ . Sedangkan yang menentukan bobot dari *hidden layer* menuju ke lapisan output disebut dengan *layer weight* (LW) yaitu pada  $W_{11}, W_{12},$  dan  $W_{13}$ . Bias pertama pada  $V_{10}, V_{20},$  dan  $V_{30}$ . Bias kedua  $W_{10}$ .

**C. Pengambilan Data**

Pengambilan data dilakukan di area Surabaya menggunakan avometer sebagai alat yang digunakan untuk mengukur daya keluaran yang dihasilkan oleh modul *photovoltaic*. Seperti yang sudah dibahas sebelumnya bahwa data daya diperoleh dari tegangan dan arus yang dihasilkan oleh modul PV tersebut. *Photovoltaic* yang digunakan memiliki spesifikasi tertentu dan luas permukaan yang berpengaruh pada daya yang dihasilkan. Waktu pengambilan dilakukan selama 3 hari terhitung tanggal 14 - 16 April 2013 pukul 07.00 - 18.00 WIB setiap setengah jam sekali. Daya keluaran yang diperoleh selanjutnya di-gunakan sebagai 2 data input. Data daya yang sudah diperoleh dapat dilihat pada lampiran. Berdasarkan data yang diperoleh, *photovoltaic* mampu menghasilkan daya dengan nilai yang besar yaitu antara pukul 10.00 - 11.30 WIB tergantung dari kondisi cuaca dan intensitas matahari setempat.

**D. Pemrograman Backpropagation dengan Simulasi Matlab**

Jaringan syaraf tiruan yang menerapkan algoritma *backpropagation* (BPA) dapat diimplementasikan pada bahasa Matlab. *Software* ini mampu menurunkan angka iterasi pada performansi algoritma.

**1) Inisialisasi Jaringan**

Langkah pertama yang dilakukan untuk memprogram *backpropagation* dengan Matlab yaitu membuat inisialisasi jaringan. Untuk jaringan yang terdiri dari 2 masukkan, sebuah *hidden layer* (lapisan tersembunyi) yang terdiri dari 3 unit, dan sebuah keluaran, pola jaringan seperti ini dapat disingkat pola

2-3-1. Data yang digunakan untuk pengujian dapat dilihat pada lampiran. Model matriks yang digunakan dalam Ms.Excel perlu diperhatikan karena akan dibuat perintah pada M-file untuk memanggil data dari Excel. Jika matriks yang digunakan salah maka sistem tidak akan berjalan. Data yang diperoleh yaitu berupa daya selama 3 hari dimana terdiri dari 2 input dan 1 target. Matriks yang digunakan pada Ms.Excel yaitu 2\*48 yang berarti pada data input ada 2 row (input 1 dan input 2) dan 48 column (48 data daya). Sedangkan untuk target matriks yang digunakan yaitu 1\*48.

### 2) Parameter Pelatihan Backpropagation

Ada beberapa parameter pelatihan yang dapat di atur sebelum dilakukan pelatihan. Dengan memberi nilai yang sudah ditentukan untuk dimasukkan pada parameter-parameter tersebut maka dapat diperoleh hasil yang optimal. Parameter-parameter yang digunakan adalah sebagai berikut [10]:

- `net.trainParam.lr`  
Digunakan untuk menentukan laju pemahaman ( $\alpha = learning\ rate$ ). Defaultnya = 0.01, semakin besar nilai  $\alpha$  maka semakin cepat proses pelatihan. Akan tetapi jika  $\alpha$  terlalu besar, maka algoritma menjadi tidak stabil dan mencapai titik minimum lokal.
- `net.trainParam.lr_inc`  
Lanjutan dari laju pemahaman dengan menentukan nilai kenaikannya (inc : increase).
- `net.trainParam.mc`  
Penambahan momentum bertujuan untuk menghindari perubahan bobot yang terlalu besar akibat perbedaan data.
- `net.trainParam.goal`  
Digunakan untuk menentukan batas nilai MSE agar iterasi dihentikan. Iterasi berhenti jika MSE kurang dari batas yang ditentukan dalam parameter ini atau jumlah epoch mencapai batas yang ditentukan.
- `net.trainParam.epochs`  
Digunakan untuk menentukan jumlah epoch maksimum pelatihan (default : 100 epoch).

### 3) Penentuan Metode Modifikasi sebagai Fungsi Pelatihan

Metode modifikasi pada algoritma *backpropagation* bertujuan untuk mempercepat fungsi pelatihan yang relatif lambat. Metode yang digunakan harus disesuaikan dengan kebutuhan dalam melatih suatu jaringan. Ada beberapa metode yang sering digunakan sebagai fungsi pelatihan, di antaranya [10]:

- **Variabel laju pemahaman (traingda,traingdx)**  
Metode penurunan gradien dengan momentum (traingda,traingdx). Performansi dari algoritma ini sangat sensitif terhadap besarnya nilai *learning rate*. Jika diberi nilai *learning rate* yang besar maka hasil dari algoritma akan beresilasi dan menjadi tidak stabil dalam pelatihan. Sedangkan, jika nilai *learning rate* terlalu maka proses iterasi membutuhkan waktu yang cukup lama.

- **Metode Levenberg Marquardt (trainlm)**  
Dikenal sebagai metode pembelajaran dengan proses penurunan tercepat karena tidak memerlukan perhitungan matriks Hessian. Algoritma Levenberg-Marquardt yang digunakan yaitu sebagai berikut :

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \quad (3.1)$$

### 4) Pembagian Data

Pembagian data berdasarkan 3 parameter yaitu proses *training*, *validating*, dan *testing*. Bertujuan agar dapat mengetahui porsi dari masing-masing proses tersebut. Ada 2 cara yaitu dengan metode persentase dan pembagian menurut jumlah data yang ada. Data yang digunakan pada metode persentase lebih bersifat *random* atau acak pada saat menjalankan sistem. Misalnya data yang dimiliki sebanyak 100 data sampel, persentase yang diinginkan untuk *train* 70%, *validasi* 15%, dan *test* 15%. Jadi data yang diolah untuk *training* sebanyak 70 data, *validasi* 15 data, dan *testing* 15 data. Tetapi kelemahannya adalah tidak dapat diketahui data mana saja yang diolah karena bersifat acak.

Untuk mengetahui data mana saja yang digunakan pada saat pengolahan maka metode dengan pembagian menurut jumlah data diperlukan. Contohnya 100 data sample dengan pembagian data ke-1 sampai dengan data ke-70 digunakan sebagai data *training*, data ke-71 sampai 85 untuk *validasi*, dan data ke-86 sampai 100 untuk *testing*. Metode ini lebih terperinci dibandingkan dengan metode persentase. Oleh sebab itu, pada perancangan sistem prediktor daya menggunakan metode ini dengan jumlah data yang terkumpul sebanyak 48. Pembagian untuk *training* data ke-1 sampai dengan data ke-34, *validasi* data ke-35 sampai ke-41, dan *testing* data ke-42 sampai ke-48. Jadi dapat diketahui banyaknya data dan data apa saja yang akan diolah.

### 5) Penentuan Bobot dan Bias Awal

Bobot awal akan mempengaruhi kecepatan konvergensi dan apakah suatu jaringan yang dibuat mencapai titik minimum lokal atau global. Nilai bobot awal tidak boleh terlalu besar karena nilai turunan fungsi aktivasi yang digunakan menjadi sangat kecil. Oleh sebab itu, standar *backpropagation* bobot dan bias diisi dengan bilangan acak kecil. Bobot dan bias akan terus berubah setiap kali membentuk jaringan. Tetapi bobot dan bias awal dapat ditentukan dengan memberikan bobot dan bias tertentu, dengan cara memberikan nilai pada `net.IW`, `net.LW`, dan `net.b`. `net.IW` (*input weight*) yaitu unit masukan yang terhubung dengan lapisan tersembunyi paling bawah. `net.LW` (*layer weight*) digunakan untuk menyimpan bobot dari unit lapisan tersembunyi ke unit keluaran.

Sedangkan kode untuk memberikan nilai bias awal yaitu `net.b`. Jika jumlah bias yang dibuat pada arsitektur sebanyak 2 buah bias maka berikan nilai `net.b{1}` dan `net.b{2}`. Dengan nilai bobot dan bias awal yang sudah ditentukan dapat dimasukkan dalam *M-file* sehingga nilainya tidak berubah-ubah.

### III. ANALISIS DATA DAN PEMBAHASAN

#### A. Pengolahan dan Pengelompokan Data

Pengolahan pada *listing program* menentukan setiap input dan output dengan nilai range minimum dan maksimum. Dengan menggunakan kode normalisasi data maka secara otomatis data yang nilainya lebih dari 1 dapat dijadikan dengan range -1-1. Lalu pengelompokan data berdasarkan 3 parameter yaitu *training*, *validasi*, dan *testing* menggunakan perintah membagi data dengan menentukan sendiri data mana saja yang digunakan untuk pelatihan masing-masing parameter. Dengan ini dapat diketahui data ke-berapakah yang akan di olah dengan jaringan yang sudah dirancang.

Pengelompokan data bertujuan untuk mengetahui secara detail data yang akan digunakan untuk *training*, *validasi*, dan *testing*. Jumlah data yang dimiliki yaitu sebanyak 48 dengan pembagian data ke 1-34 untuk *training*, data ke-35; 37; 39; 41;43;45;47 untuk data *validasi*, dan data ke-36; 38; 40; 42; 44;46;48 untuk data *testing*.

#### B. Pelatihan dengan Variabel Peubah

Variabel yang di ubah-ubah yaitu jumlah neuron dan *training function*. Penentuan besarnya *learning rate* diubah-ubah sampai menemukan nilai *learning rate* yang sesuai dengan kebutuhan dalam melatih jaringan. Begitu pula dengan banyaknya jumlah epoch.

Pada pelatihan perancangan ini digunakan *learning rate* senilai 0,9 dan epoch sebanyak 300. Penentuan epoch 300 didasarkan pada saat proses pelatihan iterasi berhenti kurang dari 300. Oleh sebab itu tidak perlu epoch dengan nilai yang besar.

Sebelumnya telah dilakukan pelatihan dengan beberapa variasi keadaan. Dengan merubah banyaknya neuron dan fungsi pelatihan atau *training function* yang digunakan. Ada 6 keadaan untuk penentuan jumlah neuron dan *training function* terbaik dihubungkan dengan besarnya nilai *Mean Square Error* (MSE) yang dihasilkan serta nilai korelasi antara output dan target (regresi). Dari hasil pelatihan tersebut diperoleh persamaan output (daya prediksi) yang kemudian dapat diketahui besarnya error masing-masing.

Pada Tabel 1 menunjukkan hasil dari pelatihan jaringan dengan variasi keadaan. Dapat dilihat bahwa *training function* dengan metode *training* Levenberg-Marquardt mampu mempercepat proses iterasi dengan nilai MSE yang kecil dibandingkan penggunaan *train gradient descent adaptive* (*traingda*). Selain penggunaan *learning function*, kecepatan iterasi juga dipengaruhi oleh penentuan bobot dan bias awal sehingga jaringan secara langsung melakukan proses pelatihan tanpa harus menentukan bobot secara *random*. MSE terkecil diperoleh pada saat jumlah neuron 30, yaitu 0,0113. Perbandingan ketiga peubah jumlah neuron dengan *learning function* *trainlm* ditunjukkan berdasarkan persamaan output plot regresi pada saat proses pelatihan jaringan.

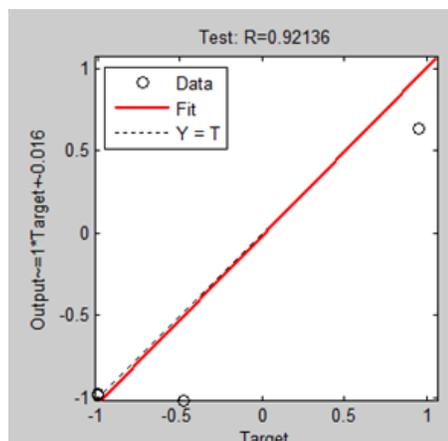
1. Jumlah neuron = 20

*Training function* = TRAINLM

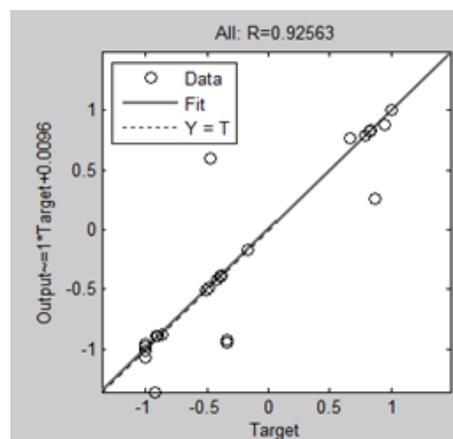
Dari hasil pelatihan diperoleh persamaan output untuk daya prediksi yaitu  $0,94 \cdot \text{Target} + 0,063$ . Hasil dari persamaan inilah yang akan dibandingkan dengan daya aktual atau target yang diinginkan.

Tabel 1.  
Tabel perbandingan dengan beberapa keadaan

Keadaan	Jumlah neuron	Berhenti di iterasi ke-	<i>Training function</i>	MSE
1.	20	6	TRAINGDA	0,0633
2.	20	6	TRAINLM	0,0180
3.	30	6	TRAINGDA	0,0375
4.	30	6	TRAINLM	0,0113
5.	40	40	TRAINGDA	0,0418
6.	40	8	TRAINLM	0,0145



Gambar 3. Plot regresi untuk n = 20 dan *training function* = *trainlm*



Gambar 4. Plot regresi untuk n = 30 dan *training function* = *trainlm*

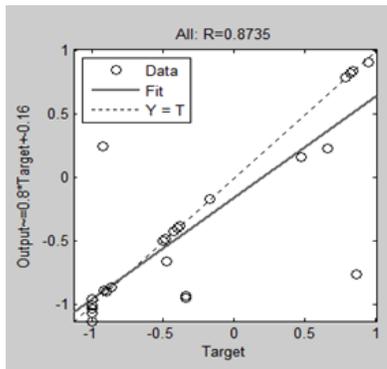
2. Jumlah neuron = 30

*Training function* = TRAINLM

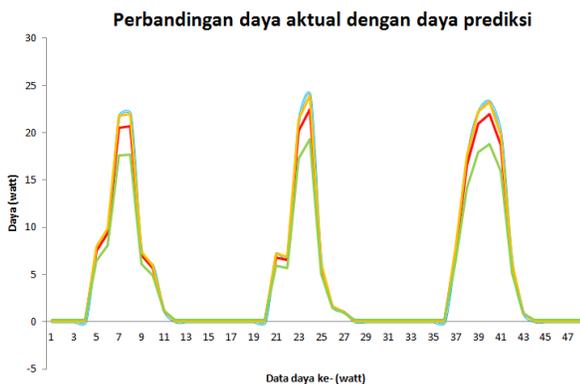
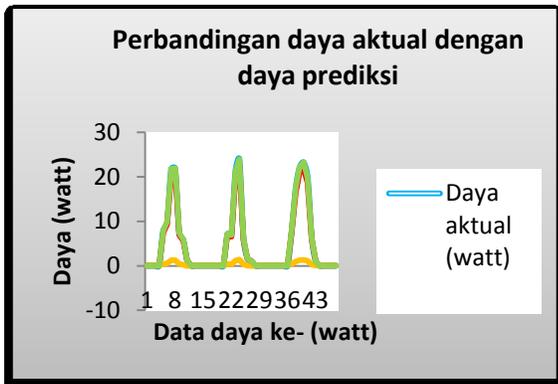
Dari hasil pelatihan diperoleh persamaan output untuk daya prediksi yaitu  $1 \cdot \text{Target} + 0,0096$ . Data target yang dimiliki dikalikan dengan 1 kemudian dijumlahkan dengan 0,0096 yang artinya daya prediksi keseluruhan hanya memiliki selisih sebesar 0,0096 dengan daya target atau daya aktual yang diinginkan.

3. Jumlah neuron = 40

*Training function* = TRAINLM



Gambar. 5. Plot regresi untuk n = 40 dan *training function* = trainlm



Gambar 6. Grafik perbandingan daya aktual dengan daya prediksi

Tabel 2.  
Hasil Perbandingan Daya Aktual dan Prediksi

Variasi keadaan	Rata-rata error (watt)	error (%)	Akurasi %
N=20	0,23978	6,163	93,837
N=30	0,0096	0,19	99,81
N=40	0,84906	20,39	79,61

Dari hasil pelatihan diperoleh persamaan output untuk daya prediksi yaitu  $0,8 \cdot \text{Target} + 0,16$ . Besar nilai regresi yang diperoleh yaitu 0.8735. Jika dibandingkan dengan parameter dengan jumlah neuron 20 dan 30, pelatihan dengan n = 40 memiliki nilai R yang paling kecil.

C. Perbandingan Daya Aktual dengan Daya Prediksi

Dari proses pelatihan dengan beberapa keadaan hingga diperoleh hasil yang terbaik, maka data daya aktual yang ada di-

substitusikan ke persamaan output yang ada pada plotting regresi. Maka diperoleh perbandingan daya aktual dan daya prediksi dengan *error* dari masing-masing keadaan dengan *trainlm* yang ditunjukkan pada Gambar 6. Grafik pada Gambar 6 menunjukkan garis biru yang mempresentasikan daya aktual, yang kemudian dibandingkan dengan garis merah yaitu daya prediksi dengan neuron sebanyak 20 dan MSE sebesar 0,0180, garis orange adalah daya prediksi neuron = 30 dan MSE = 0,0113, sedangkan garis hijau daya prediksi neuron = 40 dan MSE = 0,0145. Berdasarkan hasil persamaan output diperoleh nilai rata-rata *error* dan akurasi masing-masing variasi keadaan. Rata-rata error dan performansi ditunjukkan pada Tabel 2. Berdasarkan Tabel 6 diperoleh rata-rata *error*, persentase *error*, dan tingkat persentase akurasi dari masing-masing variasi keadaan. Hasil terbaik ditunjukkan dengan rata-rata error keseluruhan sebesar 0,0096 dengan persentase akurasi sebesar 99,81% pada saat neuron sebanyak 30 node.

IV. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan untuk merancang sistem prediksi daya keluaran pada panel *photovoltaic* terdapat beberapa kesimpulan yaitu *Trainlm* adalah fungsi pelatihan yang paling baik dibandingkan dengan *trainlm*. MSE terkecil yang menunjukkan hasil terbaik pada perancangan diperoleh dengan memasukkan jumlah neuron = 30 dan *training function* = *trainlm* yaitu 0,0113. Didapatkan error sebesar 0,0096 dengan persentase akurasi sebesar 99,81%. Perancangan ini mampu memprediksi daya yang dihasilkan oleh panel *photovoltaic*.

V. DAFTAR PUSTAKA

- [1] [http://www.bmkg.go.id/BMKG\\_Pusat/Profil/Tugas\\_dan\\_Fungsi.bmkg](http://www.bmkg.go.id/BMKG_Pusat/Profil/Tugas_dan_Fungsi.bmkg). [Diakses 30 Mei 2013].
- [2] Sedaghati, F., Nahavandi, A., Badamchizadeh, M.A., Ghaemi, S. & Fallah, M.A., "PV Maximum Power-Point Tracking by Using Artificial Neural Network," *Hindawi Publishing Corporation*, vol. 2012, 2012.
- [3] Hiyama, T. and Kouzuma, S., "Application of Neural Network to Prediction of Maximum Power from PV Modules," *IEEE*, pp. 349-354, 1993.
- [4] Anonim, "Sel Surya," 2009.
- [5] Tsai, H.L., Tu, C.S., & Su, Y.J., "Development of Generalized Photovoltaic Model Using MATLAB / SIMULINK," dalam *WCECS*, San Francisco, 2008.
- [6] Strong, S.J., "The Solar Electric House," dalam *Energy for the Environmentally Responsible, Energy-Independent Home*, 1993.
- [7] Hermawan, Arief, Jaringan Saraf Tiruan, Teori dan Aplikasi, Yogyakarta: ANDI, 2006.
- [8] Sinuhaji, F., "Jaringan Saraf Tiruan untuk Prediksi Keputusan Medis Penyakit Asma," *USU Repository*, 2009.
- [9] Nazari, J. and Ersoy, Okan K., "Implementation of back-propagation neural network with Matlab," *ECE Technical Report*, Indiana, 1992.
- [10] Siang, J.J., Jaringan Saraf Tiruan dan Pemrogramannya menggunakan Matlab, Yogyakarta: Andi, 2005.
- [11] Abhishek, K., Kumar, A., Ranjan, R. & Kumar, S., "A Rainfall Prediction Model using Artificial Neural Network," *IEEE*, 2012.
- [12] Touretzky, D. & Laskowski, K., "Artificial Neural Network," *Neural Network for Time Series Prediction*, pp. 2-42, 2006.