

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

Dalam bab ini akan dijelaskan analisis dan desain dari sistem yang akan dibangun. Hal ini dimaksudkan untuk memberikan gambaran yang jelas tentang alur kerja sistem dan desain sistem.

4.1 Analisis Sistem

Analisis sistem dimaksudkan untuk memberikan gambaran yang jelas tentang program aplikasi. Program aplikasi yang akan dibangun adalah sebuah sistem yang dapat meramalkan besar kenaikan permukaan air kedepannya berdasarkan data koefisien aliran, intensitas curah hujan, dan luas daerah aliran sungai sebelumnya. Sistem yang akan dibangun menggunakan metode *artificial neural network backpropagation*, yang mengambil studi kasus di Desa Tanjung Jaya, Kelurahan Surabaya, Kecamatan Sungai Serut, Kota Bengkulu.

Objek Desa Tanjung Jaya diangkat karena, merupakan salah satu Desa yang terletak di kelurahan Surabaya yang sering mengalami banjir setiap tahunnya, hal ini dikarenakan debit banjir yang ada lebih besar daripada daya tampung sungai itu sendiri. Masyarakat sendiri menyadari akan terjadinya banjir saat curah hujan yang tinggi terjadi di Desa Tanjung Jaya tersebut. Sehingga, sebagian dari masyarakat belum sempat melakukan tindakan siap siaga untuk mengatasi jika terjadi banjir. Oleh sebab itu, penulis mencoba untuk membuat suatu sistem peramalan kenaikan

permukaan air di Desa Tanjung Jaya dengan *artificial neural network backpropagation*.

4.1.1. Analisis Kebutuhan Sistem

Analisis sistem dilakukan dengan terlebih dahulu melihat kebutuhan dari sistem. Analisis kebutuhan sistem dilakukan untuk mengidentifikasi kebutuhan – kebutuhan yang diperlukan dalam membangun program aplikasi. Dalam penelitian ini dilakukan analisis kebutuhan masukan, kebutuhan proses, dan kebutuhan keluaran. Penjelasan lebih lanjut dari ketiga analisis kebutuhan tersebut adalah sebagai berikut:

1. Analisis kebutuhan masukan

Analisis kebutuhan masukan yang berupa data masukan dan perhitungan di dalam metode *backpropagation*. Data yang diambil berupa data intensitas curah hujan (I), data luas daerah aliran sungai (A) Sungai Serut, koefisien aliran (C) di Desa Tanjung Jaya. Penjelasan mengenai data masukan tersebut dijelaskan pada bagian analisis data. Sedangkan perhitungan yang digunakan mengacu pada landasan teori. Teori tentang perhitungan tersebut telah dibakukan yang ditujukan untuk mengolah perhitungan kenaikan permukaan air.

2. Analisis kebutuhan proses

Analisis kebutuhan proses yang berupa pemodelan data. Analisis ini dimaksudkan untuk menggambarkan proses-

proses yang terdapat di dalam program aplikasi melalui sebuah pemodelan. Pemodelan yang digunakan oleh penulis adalah pendekatan terstruktur dan pemakaian *Data Flow Diagram* (DFD) untuk menggambarkan aliran data.

3. Analisis kebutuhan keluaran

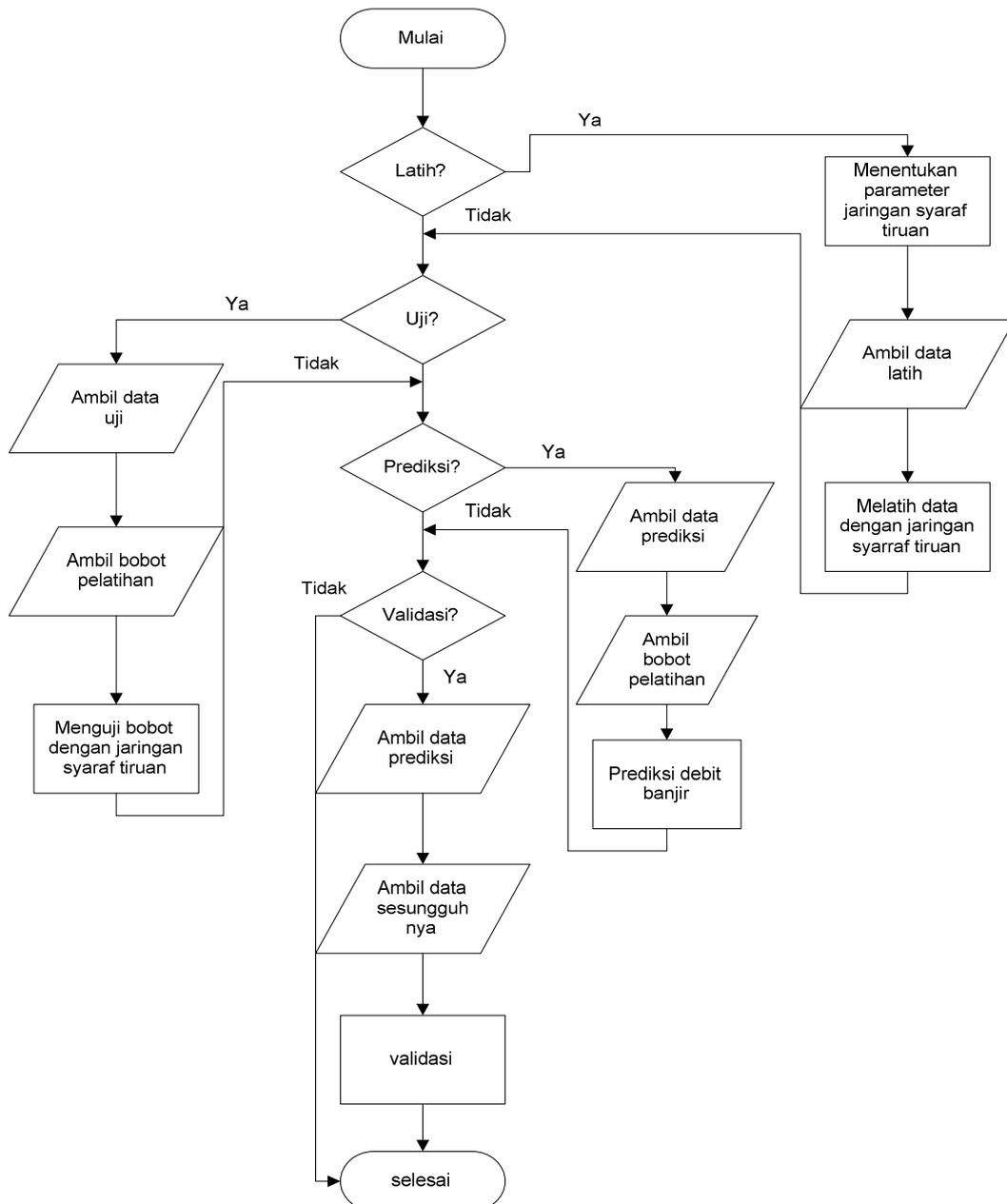
Analisis kebutuhan keluaran berupa tampilan program aplikasi yang akan dirancang pada perancangan antarmuka. Perancangan antarmuka program aplikasi ini menggunakan perangkat lunak pendukung MatLab R2008b. Program aplikasi ini nantinya akan dijalankan pada Windows *Seven* yang disesuaikan dengan sistem operasi yang digunakan oleh penulis dalam penelitian ini.

4.1.2. Analisis Data

Analisis data bertujuan untuk mengetahui data-data apa saja yang diperlukan dalam program aplikasi. Data yang diambil berupa data intensitas hujan yang terjadi selama 4 bulan (120 hari) yakni pada bulan November 2013 hingga Februari 2014 dengan rentang data selama satu jam, data koefisien aliran sungai di Desa Tanjung Jaya, dan data luas daerah aliran sungai Sungai Serut. Dimana data untuk *training* selama 3 bulan (bulan November 2013 – bulan Januari 2014) sebanyak 92 hari. Data untuk *testing* pada tanggal 1-4 Februari 2014. Dan data untuk *forecasting* pada tanggal 6 Februari 2014 untuk memprediksi kenaikan permukaan air tanggal 7 Februari

4.1.3 Analisis *Flowchart*

Bagan alir (*flowchart*) digunakan untuk mengetahui aliran prosedur dari sistem aplikasi yang akan dibangun. *Flowchart* sistem yang dirancang oleh penulis dapat dilihat pada Gambar 4.1 berikut ini:



Gambar 4.1 Flowchart Proses Sistem Peramalan Kenaikan permukaan air secara umum

Berdasarkan *flowchart* pada Gambar 4.1, terdapat empat proses utama dalam sistem yakni, pelatihan data dengan Jaringan Syaraf Tiruan menggunakan Algoritma *Backpropagation* dengan metode pelatihan *Levenberg-Marquardt*. Pengujian dengan menggunakan data target dan bobot hasil dari pelatihan dengan menggunakan Algoritma *Backpropagation* metode pengujian *Levenberg-Marquardt*, peramalan sistem dengan menggunakan bobot hasil pelatihan, serta validasi yang merupakan hasil perbandingan antara data prediksi dengan data sesungguhnya.

Berikut penjelasan *flowchart* yang terdapat pada Gambar 4.1:

1. Pertama *user* harus memilih menu pelatihan. Pada menu pelatihan ini akan dilakukan proses pelatihan terhadap data-data yang telah diinputkan oleh *user* sebelumnya pada *panel input* data. Jika *user* memilih menu pelatihan maka *user* menentukan parameter jaringan syaraf tiruan, tetapi apabila *user* tidak ingin melakukan pelatihan data maka *user* dapat memilih menu pengujian data.
2. Pada menu pengujian ini akan dilakukan pengujian terhadap data yang sebelumnya belum pernah dilatih atau data yang baru diinputkan oleh *user*. Proses uji ini dilakukan dengan mengambil data uji yang telah diinputkan oleh *user*, kemudian mengambil bobot-bobot hasil pelatihan. Selanjutnya menguji bobot dengan jaringan syaraf tiruan. Apabila *user* tidak ingin melakukan pengujian data, maka *user* dapat memilih menu prediksi.
3. Pada menu prediksi ini *user* dapat meramalkan besar kenaikan permukaan air satu jam kedepan. Pada menu prediksi ini sistem

akan mengambil data yang akan diprediksi yang telah diinputkan oleh *user*, serta bobot hasil pelatihan, yang nantinya akan dilakukan prediksi dengan menggunakan jaringan syaraf tiruan.

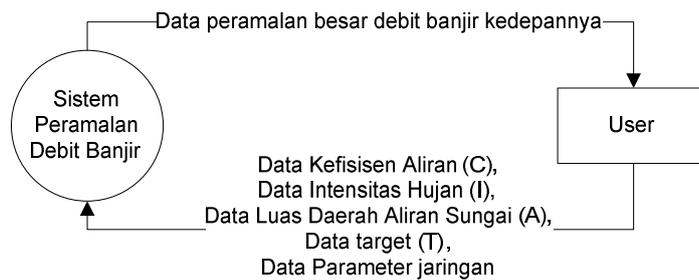
4. Apabila *user* sudah melakukan prediksi dan ingin mengetahui seberapa tingkat keakuratan hasil dari prediksi tersebut. Maka *user* dapat memilih menu validasi. Pada menu validasi ini akan dilakukan perbandingan antara hasil prediksi jaringan syaraf tiruan dengan data sesungguhnya yang disajikan dalam bentuk grafik. Apabila *user* sudah melakukan validasi maka *user* dapat keluar dari sistem.

4.2. Perancangan Sistem

Setelah dilakukan analisis sistem, dapat diketahui apa saja yang menjadi komponen masukan, keluaran dan *interface* program aplikasi yang dibangun sehingga sesuai dengan rencana. Perancangan sistem terdiri dari perancangan *Data Flow Diagram* (DFD) dan perancangan antarmuka (*user interface*).

4.2.1. Perancangan Data Flow Diagram(DFD)

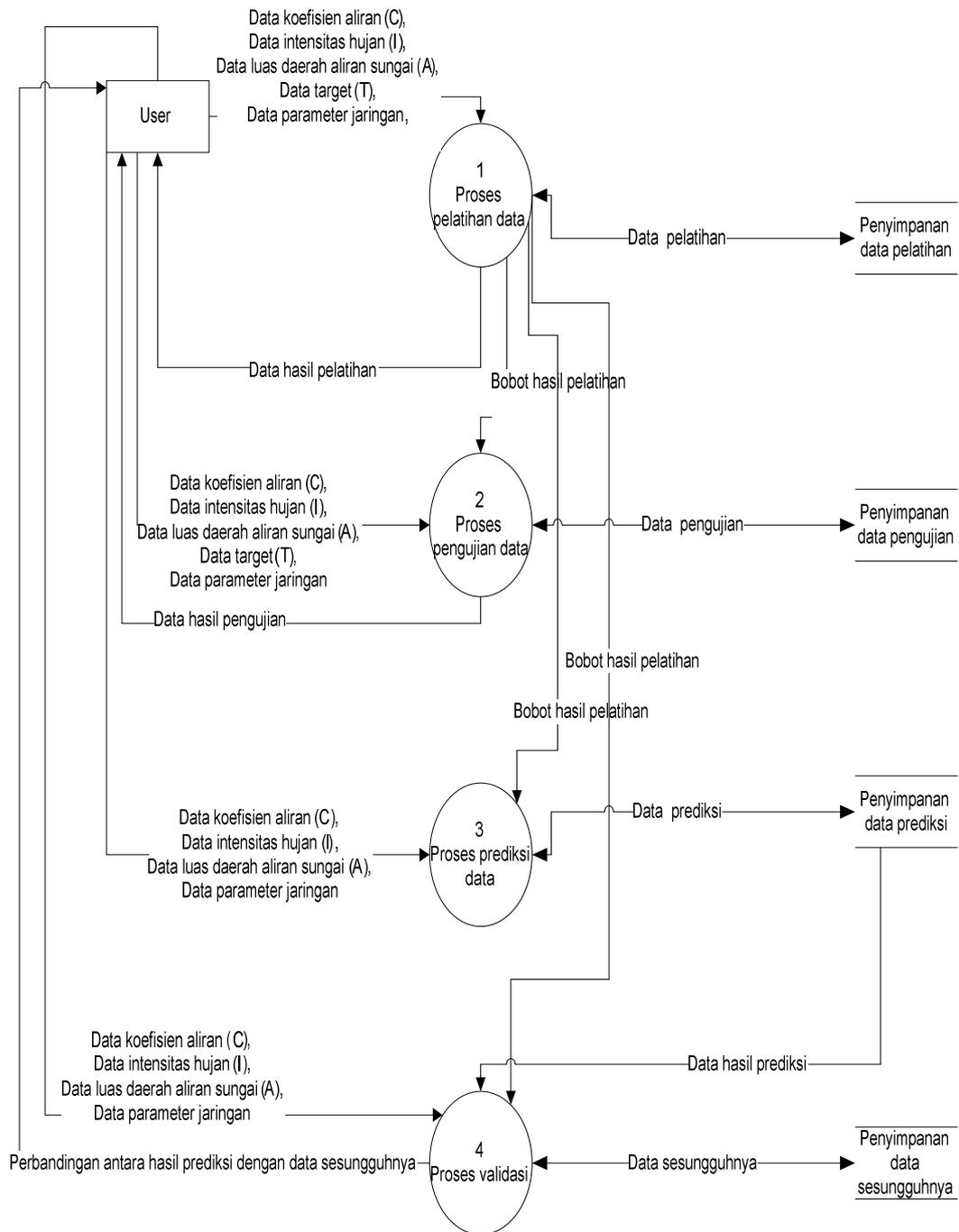
Untuk lebih memahami tahapan dalam pembuatan sistem, pada bagian ini akan dijelaskan alur proses dari sistem secara grafis dengan menggunakan pendekatan pemodelan *Data Flow Diagram* (DFD) dengan notasi DeMarco dan Yourdan. Berikut desain dari masing-masing proses:



Gambar 4.2. Diagram Konteks

Diagram Konteks menggambarkan sistem secara umum. Pada sistem ini terlihat bahwa hanya terdapat pengguna aplikasi (*user*) yang dapat menggunakan sistem. *User* akan memberikan sejumlah data yang dibutuhkan oleh sistem seperti koefisien aliran, intensitas hujan, luas daerah aliran sungai, *learning rate* (laju pembelajaran) , momentum, maksimum epoch, *goal* (kinerja tujuan), gradien minimum, *maximum fail* (maksimum kegagalan), jumlah epoch yang akan ditunjukkan kemajuannya.

Sedangkan yang diberikan oleh sistem kepada user yaitu data peramalan besar kenaikan permukaan air kedepannya berdasarkan perhitungan yang terdapat pada metode *Backpropagation*.



Gambar 4.3. DFD Level 1

DFD Level 1 merupakan penjelasan lebih lanjut dari Diagram Konteks. Di dalam sistem yang akan dibangun terdapat empat proses utama yang berjalan yaitu proses pelatihan data, proses pengujian

data, proses peramalan data dan proses validasi. Penjelasan masing-masing proses yaitu sebagai berikut:

1. Proses Pelatihan Data

Pada proses ini, *user* memasukkan data ke dalam sistem yang berupa data koefisien aliran, data intensitas hujan, data luas daerah aliran sungai, data target, dan data parameter jaringan. Pada proses ini dilakukan pelatihan data dengan *Backpropagation* menggunakan metode *Levenberg-Marquardt*. Hasil dari pelatihan akan berupa bobot yang nantinya digunakan untuk pengujian dan peramalan data. Data pelatihan ini akan disimpan dalam *database* dengan menggunakan MySQL .

2. Proses Pengujian Data

Pada proses pengujian ini, *user* akan memasukkan kembali data koefisien aliran, data intensitas hujan, data luas daerah aliran sungai, data target dan data parameter jaringan. Disini akan dilakukan pengujian apakah data yang dihasilkan sama atau mendekati data target (data sesungguhnya) dengan menggunakan bobot hasil pelatihan sebelumnya. Pengujian disini digunakan untuk membuktikan apakah pelatihan data yang kita lakukan sudah benar atau tidak. Data pengujian ini juga nantinya akan disimpan dalam *database*.

3. Proses Peramalan Data

Pada peramalan data ini, *user* kembali menginputkan data koefisien aliran, data intensitas hujan, data luas daerah aliran sungai, dan data parameter jaringan. Hasil dari peramalan ini sendiri akan berbentuk kenaikan permukaan air (Q) kedepannya. Data dari peramalan juga akan disimpan dalam *database* dengan menggunakan MySQL.

4. Proses Validasi

Pada proses validasi ini *user* kembali menginputkan data koefisien aliran, data intensitas hujan, data luas daerah aliran sungai, dan data parameter jaringan. Hasil dari data prediksi yang telah dilakukan pada proses prediksi akan dibandingkan dengan data-data yang telah diinputkan pada proses validasi untuk mengetahui seberapa akurat hasil prediksi yang dilakukan. Hasil dari validasi ini disajikan dalam bentuk grafik.

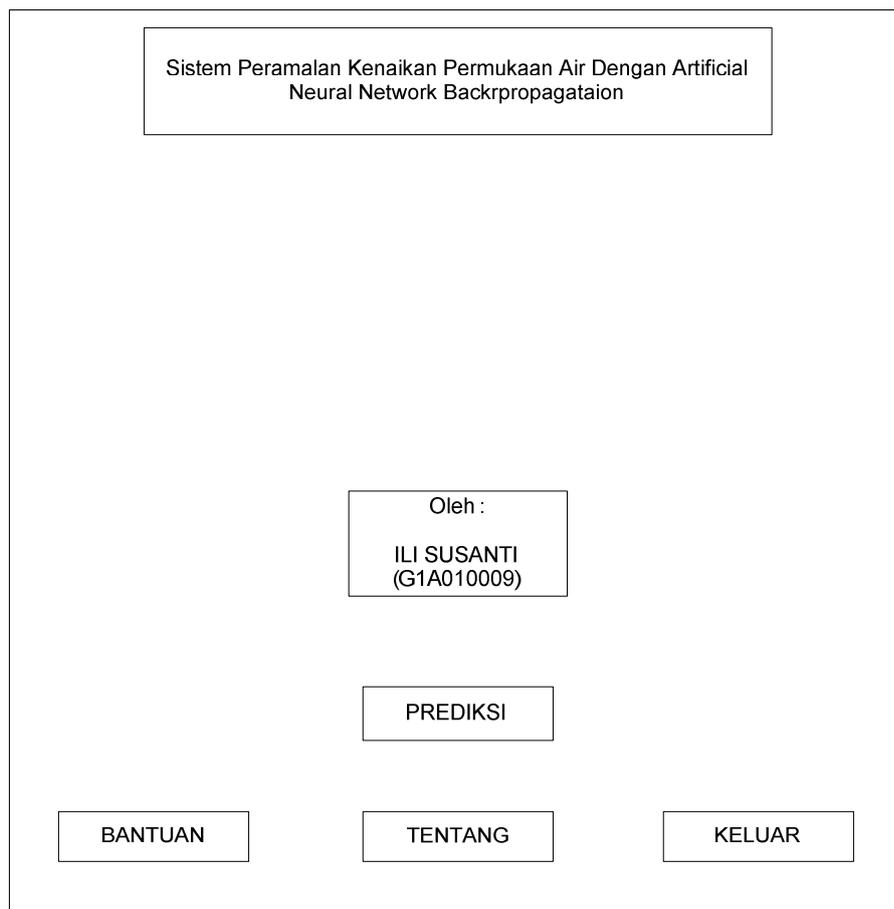
4.2.2. Perancangan Antarmuka (*User Interface*)

Tahap perancangan *user interface* menyediakan tampilan antarmuka untuk interaksi antar pengguna dengan sistem yang akan dibangun. Perancangan antarmuka ini diharapkan agar pengguna atau *user* dapat berinteraksi dengan mudah dalam penggunaan sistem tersebut. Perancangan antarmuka dalam program aplikasi ini terdiri dari lima rancangan antarmuka. Lima rancangan antarmuka yaitu *form* Menu Utama, *form* Prediksi, *form* bantuan, *form* tentang dan *form* keluar.

Penulis membuat perancangan antarmuka menggunakan Microsoft Visio 2007.

1. *Form* Menu Utama

Form ini merupakan antarmuka saat pertama kali program aplikasi dijalankan. Pada *form* ini berisi semua menu-menu pilihan yang dapat digunakan oleh *user* untuk meramalkan besar kenaikan permukaan air di Desa Tanjung Jaya, Kelurahan Surabaya, Kecamatan Sungai Serut, Kota Bengkulu. Tampilan rancangan *form* Menu Utama dalam sistem yang akan dibangun adalah sebagai berikut:



Gambar 4.4 Rancangan Antarmuka *Form* Menu Utama

Pada Gambar 4.4 diatas dapat dilihat terdapat nama dari program sistem yang akan dibangun oleh penulis. Dibawah logo Universitas Bengkulu terdapat nama dari pengembang sistem. Sedangkan pada bagian bawah terdapat menu-menu pilihan yang dapat digunakan oleh *user* yaitu menu Prediksi, Bantuan, Tentang dan Keluar.

2. *Form* Menu Prediksi

Form ini merupakan antamuka saat menu prediksi dijalankan. Pada *form* ini *user* memasukkan data-data yang dibutuhkan oleh sistem. Tampilan rancangan *form* Prediksi dalam sistem yang akan dibangun adalah sebagai berikut:

The figure displays three panels of the Prediction Form interface design:

- Panel A:** Contains input fields for 'Koefisien (0-1 %)', 'Intensitas', 'Luas', and 'Target'. To the right are buttons for 'Simpan Latih', 'Simpan Uji', 'Simpan Prediksi', and 'Validasi'.
- Panel B:** Contains input fields for 'Learning Rate (0.001 - 1)', 'Epoch', 'Momentum (0.5 - 0.9)', 'Goal', 'Show Iterasi', 'Maximum Fail', and 'Gradien Minimum'.
- Panel C:** Contains buttons for 'LATIH', 'KEMBALI', 'UJI', 'KELUAR', and 'PREDIKSI'.

Gambar 4.5 Rancangan Antarmuka *Form* Prediksi

1. Menu *Input Data* (A)

Pada menu ini pengguna aplikasi (*user*) akan mengisi data koefisien aliran (C), intensitas hujan (I), luas daerah aliran sungai (A) dan data target (T). Koefisien daerah aliran sungai yang diinputkan oleh user untuk Desa Tanjung Jaya itu sendiri sebesar 0.35 yang diperoleh dari hasil perhitungan C_{das} . Luas Daerah Aliran Sungai di Desa Tanjung Jaya yang merupakan DAS Sungai Serut sebesar 953.97 ha atau 9.5397km^2 . Setelah menginputkan data koefisien, intensitas, luas dan target, *user* akan memilih apakah data yang telah dimasukkan kedalam data latih (simpan latih), data uji (simpan uji), data ramal (simpan prediksi) dan data validasi (validasi). Dimana data yang dimasukkan oleh *user* tersebut akan secara otomatis tersimpan kedalam *database* sesuai dengan menu pilihan yang ada, yakni latih, uji, prediksi, dan bandingkan.

2. Menu Parameter Jaringan (B)

Pada menu parameter jaringan, *user* akan menginputkan jumlah *learning rate*, maksimum *epoch*, *goal* yang akan dicapai, jumlah *momentum*, dan *show iterasi*. *Maximum fail*, Gradien Minimum.

Learning rate adalah laju pembelajaran. Semakin besar nilai *learning rate* akan berimplikasi pada semakin

besarnya langkah pembelajaran. Jika *learning rate* diset terlalu besar, maka algoritma akan menjadi tidak stabil. Sebaliknya, jika *learning rate* diset terlalu kecil, maka algoritma akan konvergen dalam jangka waktu yang sangat lama. Pada sistem sudah diberi penjelasan bahwa *learning rate* yang dipakai berkisar antara 0.001 – 1. Pada sistem ini *learning rate* yang dipakai sebesar 0.1

Maksimum *epoch* adalah jumlah epoh maksimum yang boleh dilakukan selama proses pelatihan. *Iterasi* akan dihentikan apabila nilai epoh melebihi maksimum epoh. Pada sistem ini maksimum epoh yang dipakai sebesar 76.

Momentum digunakan untuk mempercepat proses pembelajaran pada saat *training* dilakukan, sehingga jaringan dapat mencapai konvergensi dalam waktu yang cepat. Tetapi jika momentum diset terlalu besar, maka jaringan akan mengalami ketidakstabilan. Oleh sebab itu pada sistem ini diberi penjelasan penggunaan momentum berkisar antara 0.5 – 0.9. Pada sistem ini momentum yang dipakai sebesar 0.8

Goal atau kinerja tujuan adalah target nilai fungsi kinerja. *Iterasi* akan dihentikan apabila nilai fungsi kinerja kurang dari atau sama dengan kinerja tujuan.

Pada sistem ini kinerja tujuan yang digunakan adalah 10^{-10} atau $1e-10$.

Show iterasi atau jumlah epoch yang akan ditunjukkan kemajuannya. Menunjukkan berapa jumlah epoch berselang yang akan ditunjukkan kemajuannya. Pada sistem ini jumlah epoch yang akan ditunjukkan kemajuannya sebesar 20.

Maximum fail atau maksimum kegagalan diperlukan apabila pada algoritma disertai dengan validitas (optional). Maksimum kegagalan ketidakvalitan terbesar yang diperbolehkan. Apabila gradien pada *iterasi* ke- k lebih besar daripada gradien iterasi ke- $(k-1)$, maka kegagalannya akan bertambah 1. Iterasi akan dihentikan apabila jumlah kegagalan lebih dari maksimum kegagalan. Pada sistem ini maksimum kegagalan yang digunakan sebesar 6.

Gradien minimum adalah akar dari jumlah kuadrat semua gradien (bobot input, bobot lapisan, bobot bias) terkecil yang diperbolehkan. Iterasi akan dihentikan apabila nilai akar jumlah kuadrat semua gradien ini kurang dari gradien minimum. Pada sistem ini gradien yang digunakan adalah $1e-10$ atau 10^{-10} .

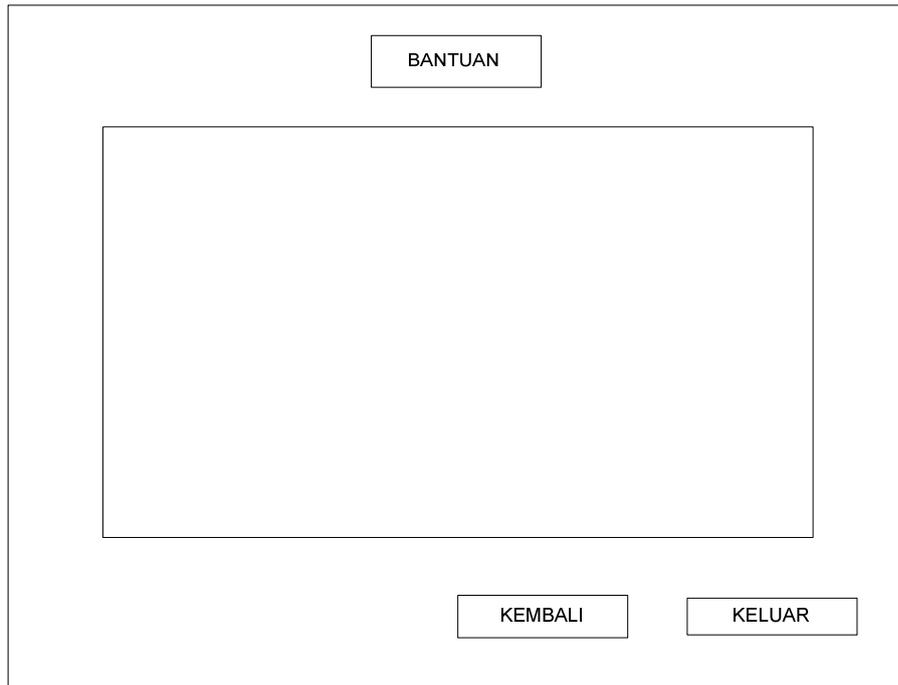
Penggunaan parameter jaringan disini bertujuan untuk mempercepat proses *training* agar dapat mencapai konvergensi.

3. Menu Pilihan (C)

Pada menu pilihan ini, *user* akan memilih apakah akan melakukan Latih, Uji, dan Validasi. Apabila *user* memilih *button* latih maka akan dilakukan proses pelatihan berdasarkan data yang telah diinputkan oleh *user* sebelumnya. Jika *user* memilih *button* uji maka akan dilakukan pengujian terhadap data-data yang telah diinputkan oleh *user* dengan menggunakan bobot hasil pelatihan. Dan jika *user* memilih *button* validasi, maka akan dilakukan proses validasi yang tujuannya untuk mengetahui seberapa akurat hasil prediksi yang dihasilkan oleh jaringan syaraf tiruan.

3. *Form* Menu Bantuan

Form ini merupakan antarmuka saat menu Bantuan dijalankan. Pada menu ini berisi tentang tata cara menggunakan sistem (berupa petunjuk) agar *user* dapat menggunakan sistem dengan baik dan benar. Tampilan rancangan *form* Bantuan dalam sistem yang akan dibangun adalah sebagai berikut:



Gambar 4.6. Rancangan Antarmuka *Form* Bantuan

4. *Form* Menu Tentang

Form ini merupakan antarmuka saat menu Tentang dijalankan. Pada *form* ini hanya terdapat penjelasan mengenai penjabaran sistem dan pengembang sistem. Tampilan rancangan *form* Tentang dalam sistem yang akan dibangun adalah sebagai berikut:

The image shows a rectangular window with a white background and a black border. Inside the window, there are two horizontal rectangular boxes stacked vertically. The top box contains the text "Tentang Sistem" centered. The bottom box contains the text "Tentang Pembuat Sistem" centered. Below these two boxes, there are two smaller rectangular buttons side-by-side. The left button contains the text "KEMBALI" and the right button contains the text "KELUAR".

Gambar 4.7. Rancangan Antarmuka *Form* Tentang

5. *Form* Menu Keluar

Form ini merupakan antarmuka saat menu Keluar dijalankan. Pada *form* ini terdapat pilihan apakah *user* ingin keluar dari aplikasi atau tidak. Jika *user* memilih Ya, maka *user* akan keluar dari sistem, tetapi jika *user* memilih tidak maka *user* akan tetap berada didalam sistem. Tampilan rancangan *form* Keluar dalam sistem yang akan dibangun adalah sebagai berikut

The image shows a rectangular window with a white background and a black border. Inside the window, there is a horizontal rectangular box containing the text "Apakah Anda Ingin Keluar Dari Sistem?". Below this box, there are two smaller rectangular buttons side-by-side. The left button contains the text "Ya" and the right button contains the text "Tidak".

Gambar 4.8. Rancangan Antarmuka *Form* Keluar

4.3. Struktur *Basis Data*

Basis data merupakan komponen yang penting pada sistem informasi. Karena di dalam basis data inilah data-data akan disimpan.

Adapun Basis Data sistem ini adalah sebagai berikut:

Tabel 4.1. Data Latih

Field	Type	Panjang	Keterangan
No*	Int	10	Identitas data training
Koefisien_la	Double		Koefisien aliran training
Hujan_la	Double		Intensitas hujan training
Luas_la	Double		Luas daerah aliran sungai training
Debit_la	Double		Kenaikan permukaan air training
Target_la	Double		Data target training

Tabel 4.2. Data Uji

Field	Type	Panjang	Keterangan
No*	Int	10	Identitas data testing
Koefisien_uj	Double		Koefisien aliran testing
Hujan_uj	Double		Intensitas hujan testing
Luas_uj	Double		Luas daerah aliran sungai testing
Debit_uj	Double		Kenaikan permukaan air testing
Target_uj	Double		Data target testing

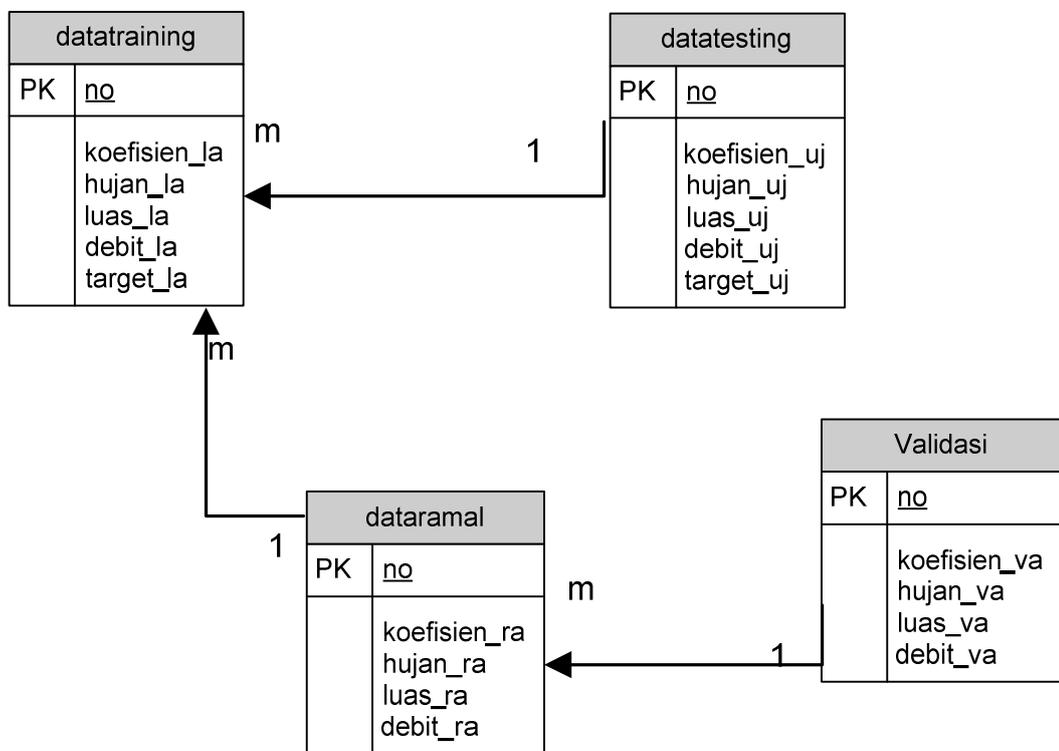
Tabel 4.3. Data Ramal

Field	Type	Panjang	Keterangan
No*	Int	10	Identitas data ramal
Koefisien_ra	Double		Koefisien aliran ramal
Hujan_ra	Double		Intensitas hujan ramal
Luas_ra	Double		Luas daerah aliran sungai ramal
Debit_ra	Double		Kenaikan permukaan air ramal

Tabel 4.4. Data Validasi

Field	Type	Panjang	Keterangan
No*	Int	10	Identitas data ramal
Koefisien_va	Double		Koefisien aliran ramal
Hujan_va	Double		Intensitas hujan ramal
Luas_va	Double		Luas daerah aliran sungai ramal
Debit_va	Double		Kenaikan permukaan air ramal

Berikut ini adalah relasi dari Tabel-Tabel yang dibuat:



BAB V

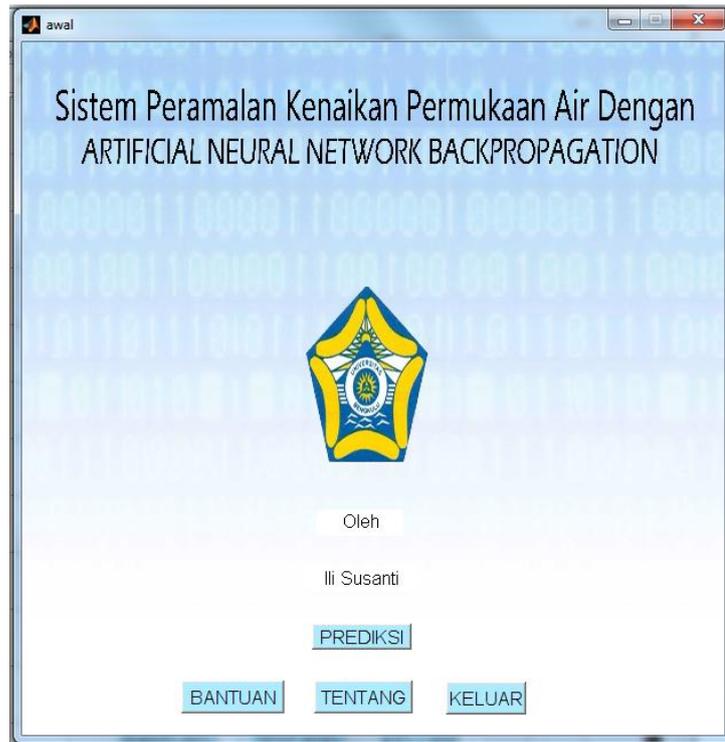
HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai hasil dan pembahasan mengenai program yang telah dibuat berdasarkan dengan analisis dan perancangan sistem pada bab sebelumnya. Pembahasan terdiri dari implementasi sistem dan uji coba program.

5.1 Implementasi Sistem

Pada bab analisis dan perancangan sistem telah dibahas tentang desain dari sistem yang akan dibuat. Desain tersebut merupakan panduan bagi penulis untuk mengembangkannya ke dalam perangkat lunak atau program yang dibuat dengan menggunakan bahasa pemrograman MatLab versi 7.7.0.741 (R2008b). Berikut ini akan dijelaskan beberapa prosedur dan fungsi program yang digunakan dalam implementasi sistem yang telah dirancang.

Antarmuka *Form* Menu Utama menampilkan sekilas informasi judul tentang program yang dibuat. Terdapat empat *button* yang merupakan menu yang dapat dipilih oleh pengguna sesuai kebutuhan, yang terdiri dari menu “Prediksi”, menu “Bantuan”, menu “Tentang”, dan menu “Keluar”. Tampilan antarmuka *Form* Menu Utama dapat dilihat pada Gambar 5.1



Gambar 5.1 Antarmuka *Form* Menu Utama

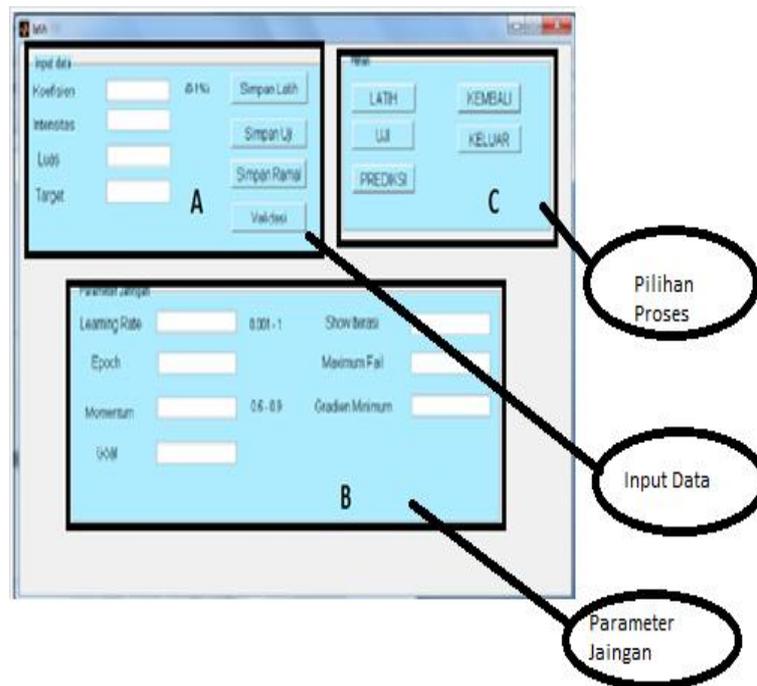
5.2 Implementasi *Form* Menu Prediksi

Tampilan antarmuka *Form* Menu Prediksi dapat dilihat pada Gambar 5.2.



Gambar 5.2. Antarmuka *Form* Menu Prediksi

Form Menu Prediksi berfungsi sebagai *form input* sejumlah data yang dibutuhkan untuk proses *training* (pelatihan), *testing* (pengujian), *forecasting* (peramalan), serta data validasi. Dalam *form* ini terbagi 3 bagian yakni : (A) *Input Data*, (B) Parameter Jaringan dan (C) Menu Pilihan Proses. Gambar 5.3 berikut ini menunjukkan pembagian atribut dan cara pengoperasian *Form* Menu Prediksi.



Gambar 5.3. Atribut-Atribut Pada *Form* Menu Prediksi

A. *Edit Text* “*Input Data*”

Atribut ini merupakan kumpulan dari *edit text* yaitu koefisien aliran, intensitas curah hujan, luas daerah aliran sungai, dan data target (data yang sesungguhnya). Koefisien aliran di Desa Tanjung Jaya ini bernilai 0.35. Nilai intensitas curah hujan di Desa Tanjung Jaya diambil berdasarkan data dari BMKG dari bulan November 2013

sampai dengan bulan Februari 2014. Luas daerah aliran sungai di Desa Tanjung Jaya ini yang merupakan Daerah Aliran Sungai Serut sebesar 953.97 ha atau 9.5397km^2 . Data target atau data sesungguhnya merupakan data intensitas hujan yang terjadi di Desa Tanjung Jaya pada bulan November 2013 sampai dengan bulan Februari 2014. *Button* simpan latih pada sistem ini berfungsi untuk menyimpan data-data yang telah diinputkan oleh *user* kedalam *database* MySQL. Adapun potongan program pada *button* simpan latih adalah sebagai berikut:

```
c= get(handles.koef, 'String');
i= get(handles.in, 'String');
a= get(handles.lu, 'String');
t= get(handles.tar, 'String');
c2=str2num(c);
i2=str2num(i);
a2=str2num(a);
t2=str2num(t);
t3=t2*a2*c2
q=c2*i2*a2
debit(c2, i2, a2, q, t3);
```

B. *Edit text* “Parameter Jaringan” (B)

Atribut ini merupakan kumpulan dari *edit text learning rate*, *epoch*, *momentum*, *goal*, *show iterasi*, *maksimum fail*, dan *gradien minimum*. *Learning rate* yang diinputkan pada sistem ini sebesar 0.1, *epoch* sebesar 200, *momentum* sebesar 0.8, *goal* sebesar $1e-10$, *show iterasi* sebesar 20, *maksimum fail* sebesar 6, dan *gradien minimum* sebesar $1e-10$. Adapun potongan program dari *learning rate* adalah sebagai berikut:

```
learning=str2double(get(hObject,'String'))
handles.learning=learning;
guidata(hObject,handles)
```

C. Button Menu “Pilihan” (C)

Atribut *button menu* pada sistem ini terdiri dari Latih, Uji, Prediksi, Keluar dan Kembali. Apabila *user* memilih *button* Latih, maka sistem akan menampilkan proses *training*. Adapun potongan program dari *button* Latih adalah sebagai berikut:

```
conn = database('curah_hujan','root','');
setdbprefs('DataReturnFormat','numeric');
D = exec(conn,'select debit from datatraining');
D = fetch(D)
D=D.Data;
[m,n]=size(D);
p = D';
conn = database('curah_hujan','root','');
setdbprefs('DataReturnFormat','numeric');
E = exec(conn,'select target from datatraining');
E = fetch(E)
E=E.Data;
[m,n]=size(E);
t = E';
min_p = min(min(p));
max_p = max(max(p));
pn = (0.8*(p-min_p)/(max_p-min_p))+0.1
min_t = min(min(t));
max_t = max(max(t));
tn=(0.8*(t-min_t)/(max_t-min_t))+0.1
net=newff(minmax(p),[10 5 1], {'tansig' 'logsig'
'purelin'}, 'trainlm');
net=init(net);
net.IW{1,1}
net.b{1,1}
net.LW{2,1}
net.b{2,1}
net.LW{3,2}
net.b{3,1}
BobotAwal_Input = net.IW{1,1}
BobotAwal_Bias_Input = net.b{1,1}
BobotAwal_Lapisan1 = net.LW{2,1}
BobotAwal_Bias_Lapisan1 = net.b{2,1}
BobotAwal_Lapisan2 = net.LW{3,2}
BobotAwal_Bias_Lapisan2 = net.b{3,1}
net.trainParam.epochs=handles.epoch
net.trainParam.goal=handles.goal;
net.trainParam.max_fail=handles.fail;
net.trainParam.min_grad=handles.gradien;
net.trainParam.mu=0.001;
```

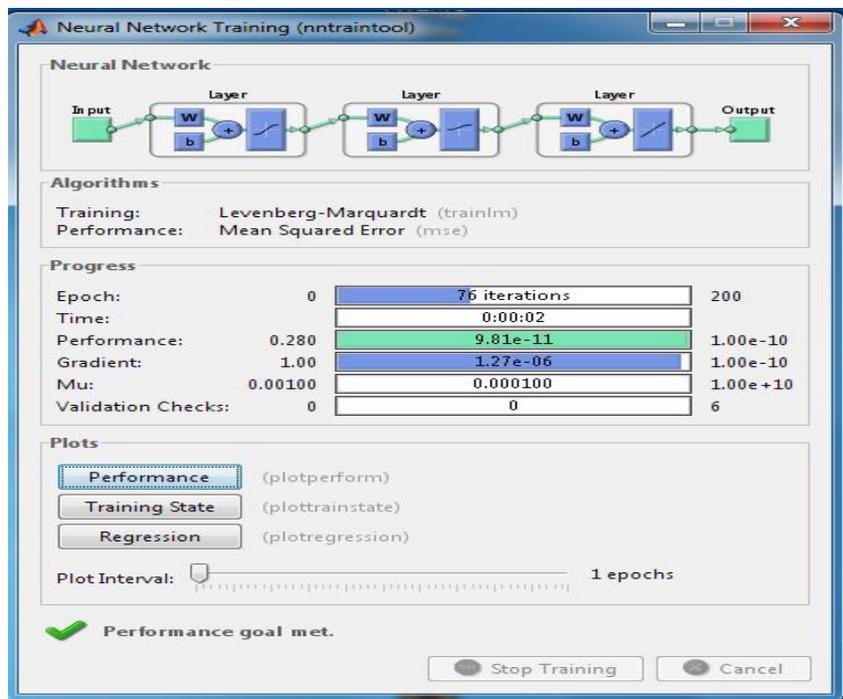
```

net.trainParam.mu_max=1e+10;
net.trainParam.show=handles.iterasi;
net.trainParam.mc=handles.momentum;
net.trainParam.lr=handles.learning;
net=train(net,pn,tn);pause
BobotAkhir_Input = net.IW{1,1}
BobotAkhir_Bias_Input = net.b{1,1}
BobotAkhir_Lapisan1 = net.LW{2,1}
BobotAkhir_Bias_Lapisan1 = net.b{2,1}
BobotAkhir_Lapisan2 = net.LW{3,2}
BobotAkhir_Bias_Lapisan2 = net.b{3,1}
an = sim(net,pn);
ta=round(((tn-0.1)*(max_t-min_t))/0.8)+min_t)
o=round(((an-0.1)*(max_t-min_t))/0.8)+min_t)
H = [1:size(p,2)]' ta' o' (ta'-o)'];
sprintf('%2d %9.2f %7.2f %5.2f\n',H')
[bx,a,rx] = postreg(o,ta)
Pause
plot([1:size(p,2)]',ta,'bo',[1:size(p,2)]',o','r*')
;
title('Hasil Pengujian dengan Data Pelatihan:
Target(o), Output(*)');
xlabel('Data ke-'); ylabel('Target/Output');
Pause

```

Hasil dari proses training jaringan dapat dilihat pada Gambar

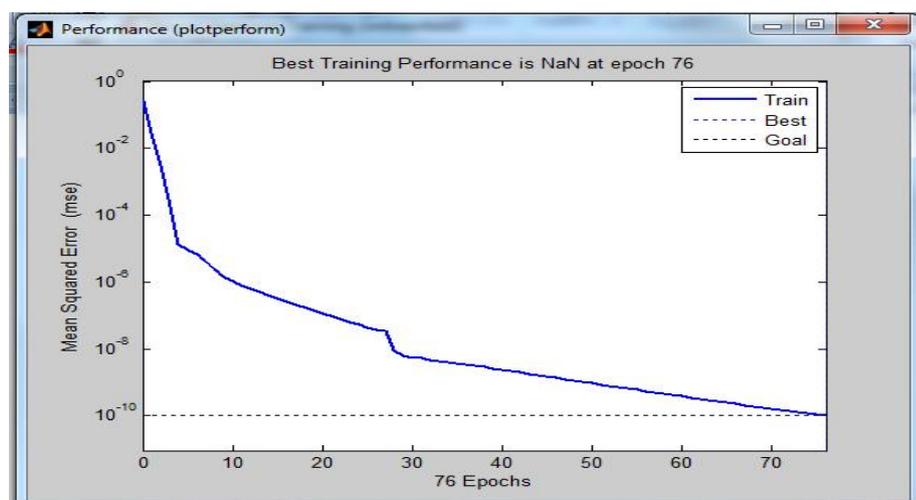
5.4. dibawah ini:



Gambar 5.4. Tampilan Proses Training

Pada Gambar 5.4. dapat dilihat bahwa *goal* $1e-10$ ditemukan hanya dalam 76 kali *iterasi*. Pada percobaan ini, jenis pelatihan yang digunakan adalah Levenberg-Marquardt. Performance, yang menyatakan metode untuk menghentikan proses jika sudah mendekati real misalnya *Mean Squared Error* (MSE). *Epoch*, menyatakan jumlah perulangan pembelajaran. Pada sistem ini epoch 76 *iterasi*, berarti proses pembelajaran berhenti setelah perulangan sebanyak 76 kali. *Time*, menyatakan waktu yang ditempuh oleh Matlab dalam melakukan pembelajaran. *Performance*, menyatakan kualitas hasil pembelajaran, makin mendekati nol, kualitasnya makin baik. *Gradient*, merupakan kemiringan antara satu *iterasi* dengan *iterasi* berikutnya. Proses pembelajaran akan berhenti biasanya jika kemiringan sudah tidak berubah. *Validation check* bermaksud untuk mengecek apakah proses pembelajaran mengarah ke arah yang tepat atau malah menyimpang. Di *Region "Plot"* tampak tombol untuk melihat grafik hasil pembelajaran. Berikut ini adalah tampilan dari *region plot*:

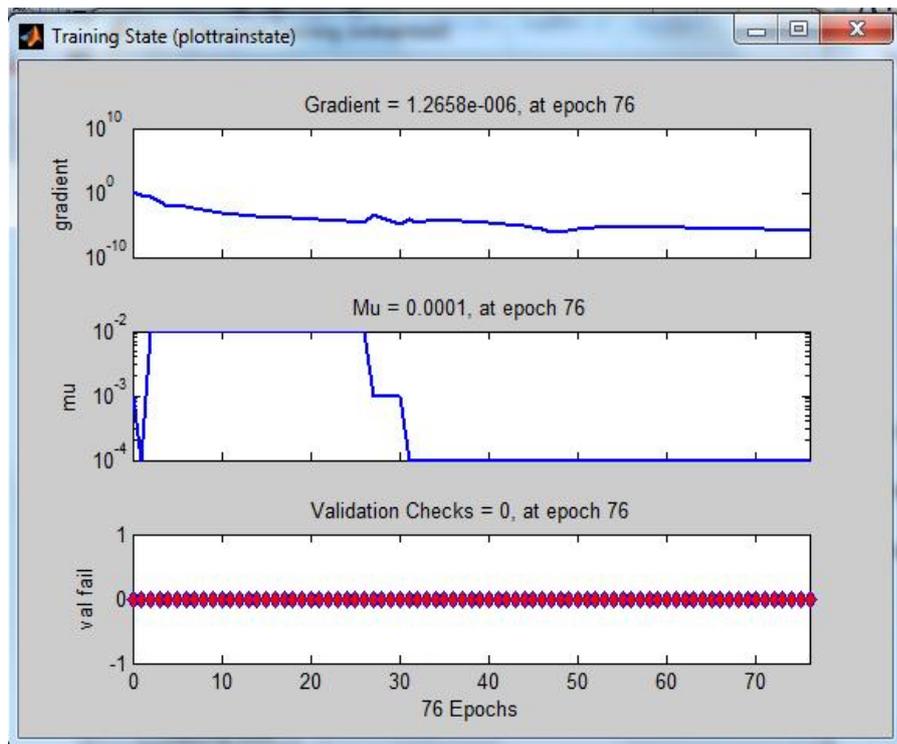
a. Tampilan *plot performance*



Gambar 5.5. Tampilan *Plot Performance*

Pada Gambar 5.5. merupakan grafik hubungan antara epoch dengan MSE (*Mean Squarred Error*). Terlihat bahwa nilai MSE turun yang artinya *training* berhasil dengan ditemukannya *goal*.

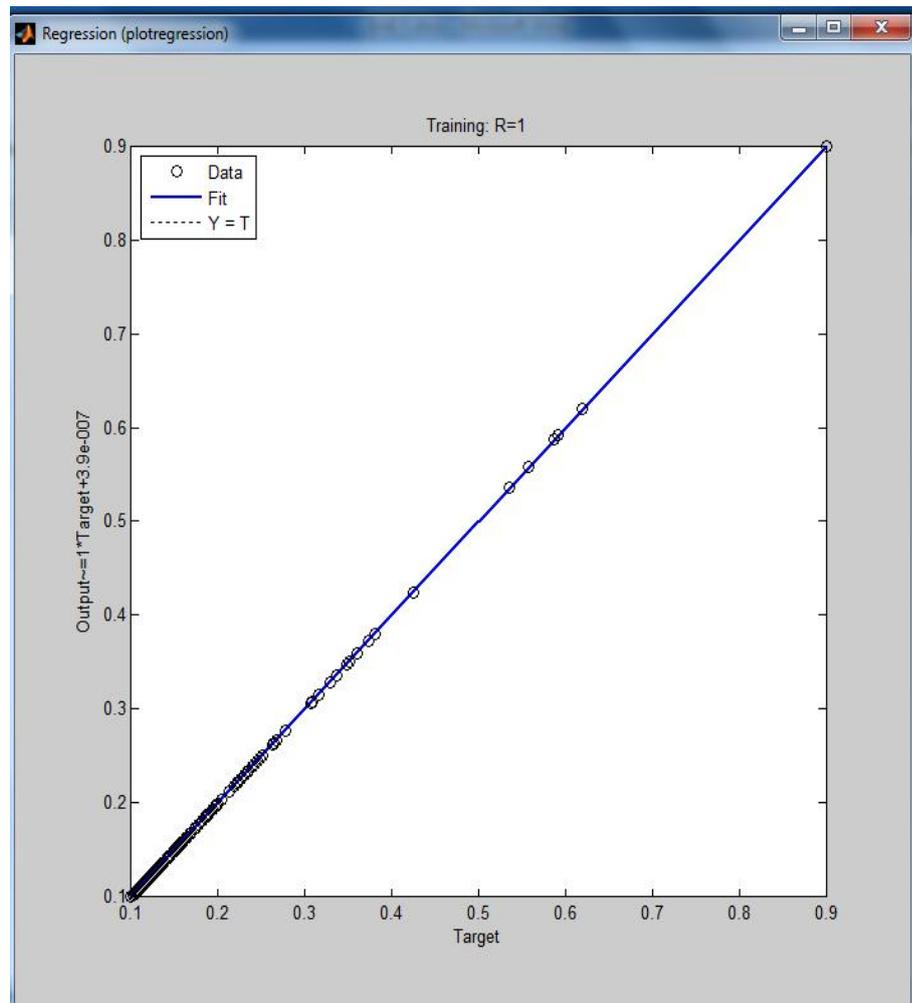
b. Tampilan *plot training state*



Gambar 5.6. Tampilan *Plot Training State*

Pada Gambar 5.6 tampilan dari *training state*, terlihat ada grafik hubungan antara *epoch* dengan gradient, grafik hubungan antara *epoch* dengan *mu*, dan grafik hubungan antara *epoch* dengan *val fail*. Nilai gradien pada proses *training* ini sebesar $1.2658e-006$, nilai *mu* sebesar 0.0001, dan *validation checks* sebesar 0.

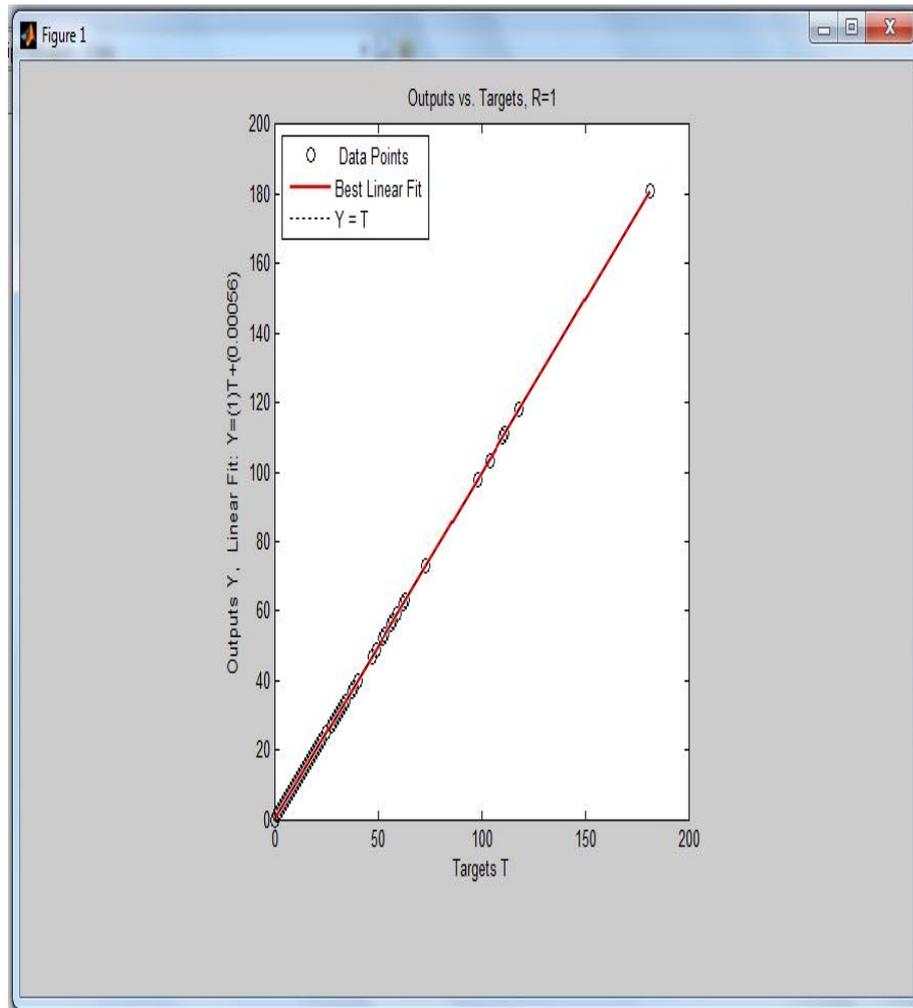
c. Tampilan *plot regression*



Gambar 5.7. Tampilan *Plot Regression*

Gambar 5.7 merupakan *plot regression* pada proses *preprocessing*, pada gambar target dengan *output* nilainya berkisar antara 0.1 – 0.9, hal ini terjadi karena *output* dengan target dilakukan proses normalisasi data sehingga data *output* dan target berada pada diantara *range* (0.1-0.9). Pada *plot regresi* ini mempunyai korelasi sebesar 1. yang artinya menunjukkan korelasi yang baik antara data *output* dengan target karena bernilai 1.

Output jaringan dan target pada data latih dianalisis dengan *regresi linier* dapat dilihat pada Gambar 5.8. dibawah ini:

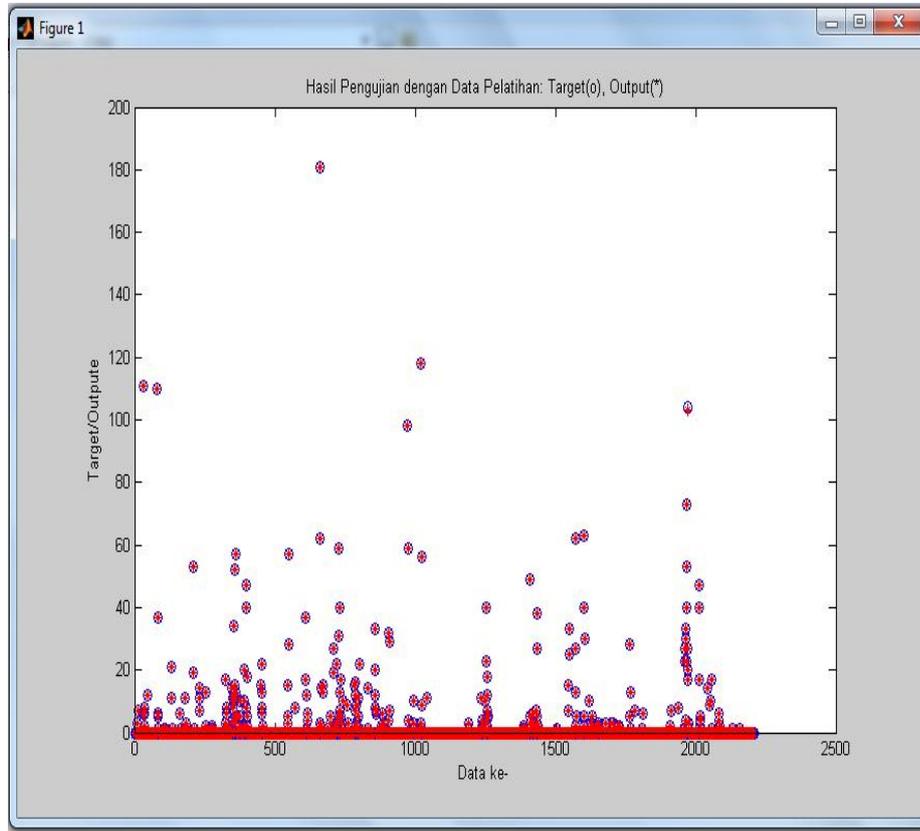


Gambar 5.8. Hubungan antara target dengan *output* jaringan, untuk data pelatihan

Pada Gambar 5.8. dapat dilihat bahwa *output* dengan target memiliki korelasi (r) sebesar 1. Hal ini menunjukkan hasil yang baik untuk kecocokkan *output* jaringan terhadap target. Hal ini dikarenakan grafiknya berbentuk linear dengan posisi data point, *output* sama dengan target berada pada *best linear fit*. *Output* jaringan untuk data pelatihan mempunyai persamaan regresi $Y = (1)T + (0.00056)$. Untuk mengetahui persamaan regresi *output* suatu jaringan dapat menggunakan fungsi *postreg* pada

Matlab. Apabila *output* jaringan tepat sama dengan targetnya, maka koefisien korelasi ini akan bernilai 1.

Berikut adalah hasil pengujian data pelatihan:



Gambar 5.9. Perbandingan antara target dengan output jaringan, untuk data pelatihan.

Pada Gambar 5.9. merupakan hasil pengujian terhadap data yang ikut dilatih pada proses pelatihan. Pada Gambar diatas data target (o) yang berbentuk lingkaran, dan data *output* (*) yang berbentuk bintang, sebagian berada di posisi yang sama persis. Hasil yang paling baik adalah jika semua data *output* dan target berada pada posisi yang sama persis.

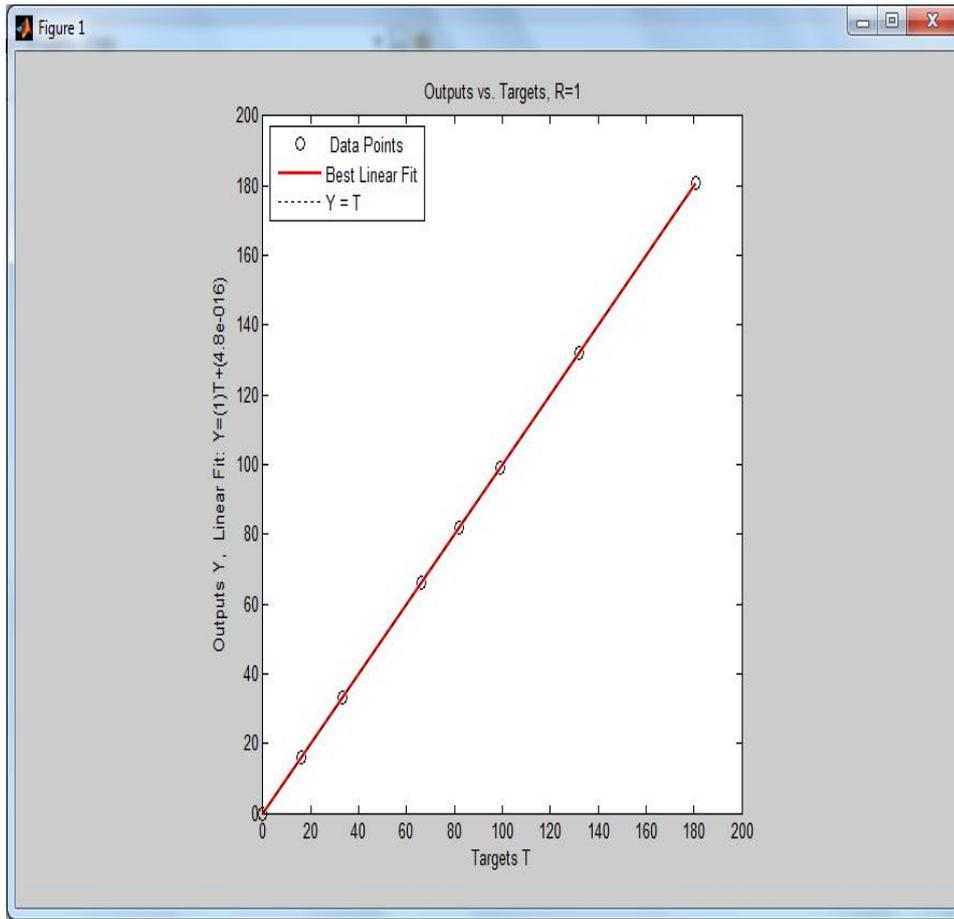
Apabila *user* memilih *button* Uji maka sistem akan melakukan proses pengujian. Adapun potongan program dari *button* Uji adalah sebagai berikut:

```

conn = database('curah_hujan','root','')
setdbprefs('DataReturnFormat','numeric');
A = exec(conn,'select debit from
datatesting');
A = fetch(A)
A = A.Data;
[m,n]=size(A);
Q = A';
setdbprefs('DataReturnFormat','numeric');
B = exec(conn,'select target from
datatesting');
B = fetch(B)
B = B.Data;
[m,n]=size(B);
TQ = B';
min_Q = min(min(Q));
max_Q = max(max(Q));
Qn = ((0.8*(Q-min_Q)/(max_Q-min_Q))+0.1)
min_TQ = min(min(TQ));
max_TQ = max(max(TQ));
Tn = ((0.8*(TQ-min_TQ)/(max_TQ-min_TQ))+0.1)
bn = sim(net,Qn)
Qa=round((((Tn-0.1)*(max_t-
min_t))/0.8)+min_t);
b=round((((bn-0.1)*(max_t-
min_t))/0.8)+min_t);
L = [(1:size(Q,2))' Qa' b' (Qa'-b')];
sprintf('%2d %11.2f %9.2f %7.2f\n',L')
[m2,b1,r2] = postreg(b,Qa)
Pause
k = [1:size(Q,2)]';
plot(k,Qa,'bo',k,b,'r*');
title('Hasil Pengujian dengan Data Uji
Target (o), Output(*)');
xlabel('Data ke-'); ylabel('Target/Output');
text(k+0.2*ones(length(k),1),Qa,int2str(k));

```

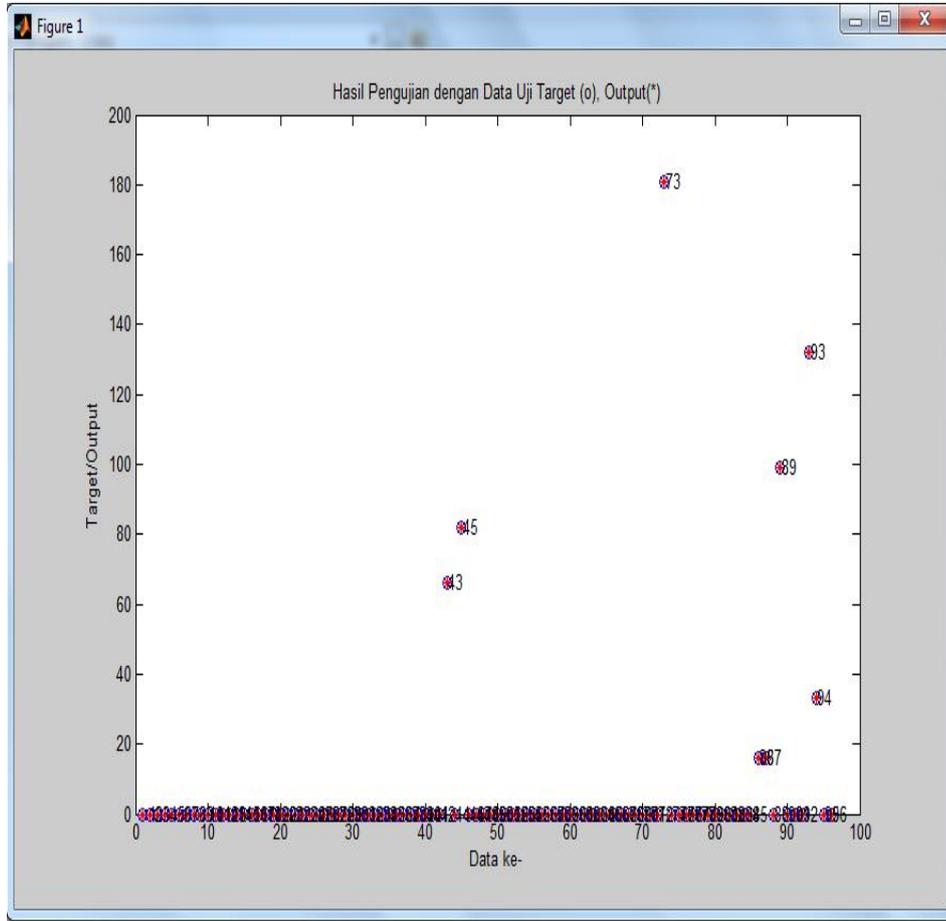
Pada Gambar 5.10. dibawah ini adalah tampilan *output* jaringan untuk data uji terhadap data yang belum pernah dilatih sebelumnya. Hal ini dilakukan untuk melakukan pengujian sejauh mana kemampuan jaringan yang telah dilatih sebelumnya mengenli pola-pola data yang bersifat *outlier*.



Gambar 5.10. Hubungan antara target dengan *output* jaringan, untuk data pengujian

Pada Gambar 5.10. merupakan hubungan antara *output* dengan target. *Output* dengan target ini mempunyai koefisien korelasi bernilai 1. Hal ini menunjukkan hasil yang baik untuk kecocokkan *output* jaringan dengan target. Hal ini dikarenakan grafik yang ada berbentuk linear dengan posisi data point, *output* sama dengan target berada pada posisi yang sama dengan *best fit linear*. Persamaan regresinya $Y = (1)T + (4.8e-0.16)$. Persamaan regresi ini didapat dengan menggunakan fungsi *postreg* pada Matlab.

Perbandingan antara data target dengan *output* jaringan, untuk data pengujian (data yang tidak dilatih sebelumnya) dapat dilihat pada Gambar 5.11. dibawah ini:



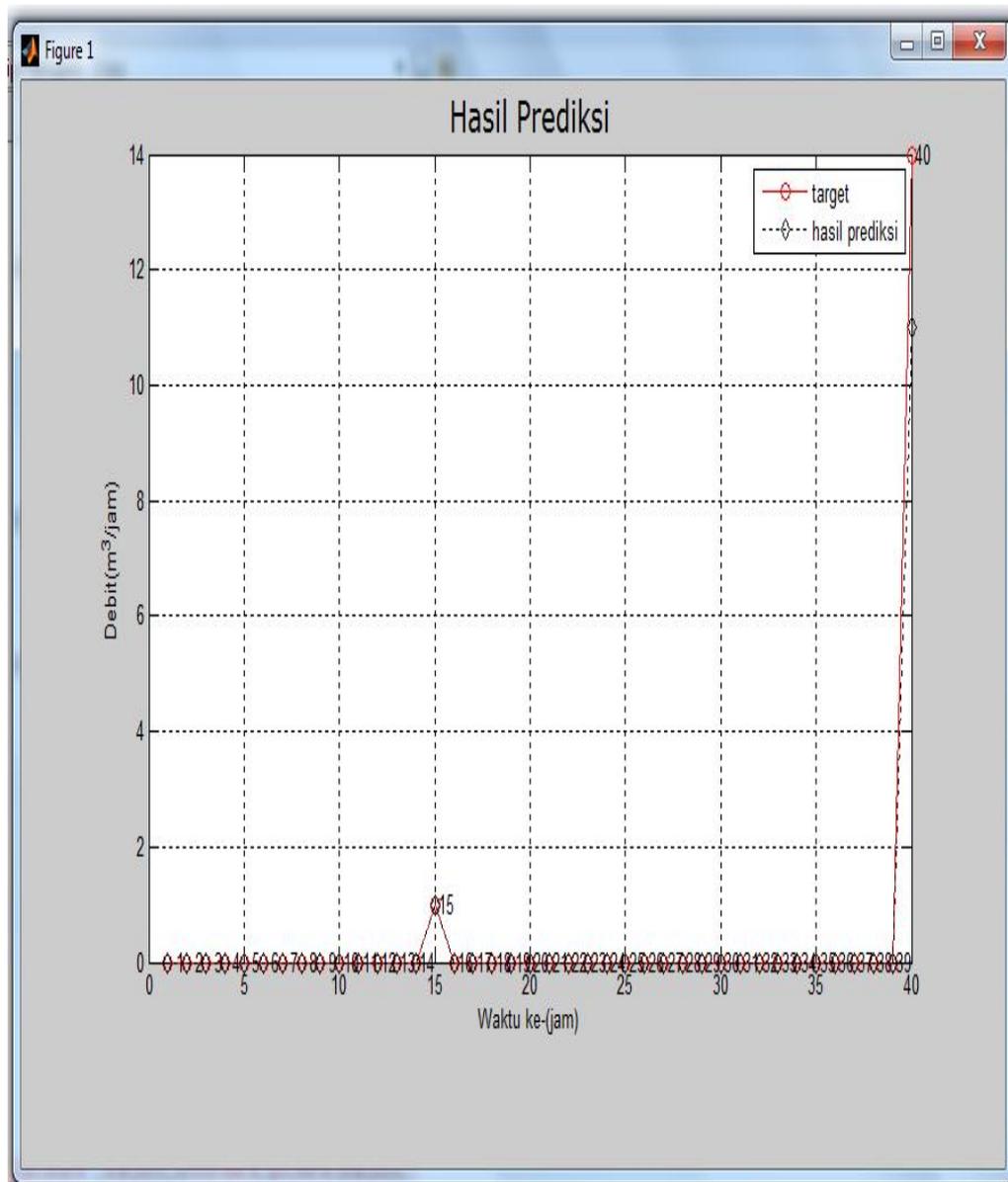
Gambar 5.11. Perbandingan antara target dengan *output* jaringan, untuk data pengujian

Pada Gambar 5.11. *output* jaringan (*) dan target (o) berada di posisi yang sama. Hasil yang terbaik terjadi apabila posisi (o) dan (*) betul-betul berada pada posisi yang sama.

Selanjutnya, apabila *user* memilih *button* Prediksi, maka akan dilakukan proses peramalan terhadap data yang telah diinputkan. Perlu diingat bahwa

data untuk proses *train*, *test*, dan *forecast* harus data yang riil tidak boleh fiktif.

Berikut ini adalah gambar dari grafik peramalan kenaikan permukaan air pada tanggal 7 Februari 2014.



Gambar 5.12. Hasil Prediksi Kenaikan permukaan air pada Tanggal 7 Februari

Berdasarkan grafik hasil prediksi pada Gambar 5.12. garis berwarna merah dengan lingkaran menunjukkan data target(data sesungguhnya), garis

putus-putus dengan berlian menunjukkan data *output*(data hasil prediksi). Perbandingan antara *output* dengan target dapat dilihat pada Tabel 5.1. dibawah ini.

Tabel 5.1. Perbandingan antara Data Target dengan Data Hasil Prediksi

Jam ke-	Data target (data sesungguhnya)	Data output (hasil prediksi)
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0
13	0	0
14	0	0
15	1	1
16	0	0
17	0	0
18	0	0
19	0	0
20	0	0
21	0	0
22	0	0
23	0	0
24	0	0
25	0	0
26	0	0
27	0	0
28	0	0
29	0	0
30	0	0

31	0	0
32	0	0
33	0	0
34	0	0
35	0	0
36	0	0
37	0	0
38	0	0
39	0	0
40	14	11

Berdasarkan Tabel 5.1 terlihat beberapa data hasil prediksi berbeda dengan data target, perbedaan antara target dengan *output* (hasil prediksi) ini disebut dengan *error*, yang akan nantinya akan dihitung dengan menggunakan MAPE (*Mean Absolute Percentage Error*).

Jika *user* memilih *button* Kembali, maka *user* akan kembali ke menu utama. Adapun potongan program dari *button back* adalah sebagai berikut:

```
figure(awal);
```

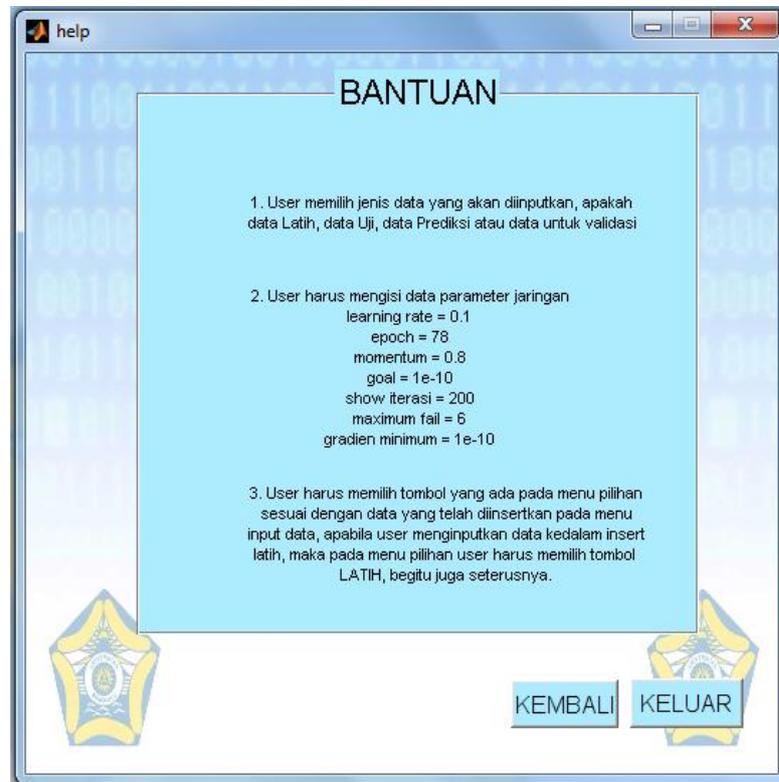
Jika *user* memilih *button* Keluar, maka *user* akan keluar dari menu Prediksi. Adapun potongan program dari *button* keluar adalah sebagai berikut:

```
close;
```

Apabila *user* keluar dari menu prediksi, maka *user* akan kembali ke *Form* Awal, Pada *Form* Awal ini terdapat *Form* Bantuan, *Form* Tentang dan *Form* Keluar.

5.3 Implementasi *Form* Bantuan

Tampilan antarmuka *Form* Menu Bantuan dapat dilihat pada Gambar 5.13.



Gambar 5.13. *Form* Menu Bantuan

Pada antarmuka *form menu* bantuan ini berisi tentang tata cara penggunaan sistem peramalan kenaikan permukaan air, pada antarmuka *form* ini terdapat dua *button* yakni *button* kembali dan *button* keluar. Dimana apabila *user* memilih *button* kembali, maka *user* akan kembali ke *form* menu utama, tetapi jika *user* memilih *button* keluar, maka *user* akan keluar dari *form menu* bantuan.

5.4 Implementasi *Form* Menu Tentang

Tampilan antarmuka *form menu* tentang dapat dilihat pada Gambar 5.14



Gambar 5.14. *Form Menu Tentang*

Pada antarmuka *form menu* tentang ini berisi tentang sistem dan pembuat sistem. Pada *form* ini juga terdapat dua *button*, yakni *button* kembali dan *button* keluar. Jika *user* memilih *button* kembali maka *user* akan kembali ke *form* menu utama, tetapi jika *user* memilih *button* keluar, maka *user* akan keluar dari antarmuka *form menu* tentang.

5.5 Implementasi *Form Menu Keluar*

Tampilan antarmuka *form* menu keluar dapat dilihat pada Gambar 5.15.



Gambar 5.15. *Form Menu Keluar*

Pada Gambar diatas apabila *user* memilih *form menu* keluar, maka akan muncul kotak pesan dialog yang berupa pilihan. Apabila *user* memilih ya maka *user* akan keluar dari sistem peramalan kenaikan permukaan air, tetapi apabila *user* memilih tidak maka *user* akan kembali ke *form* menu utama.

5.6 Pengujian Sistem

Setelah dilakukan pengujian sistem dengan data kenaikan permukaan air pada tanggal 7 Februari 2014, didapat nilai *output* yang tidak sama dengan target, oleh sebab itulah diperoleh *error*. Untuk itu, dilakukan perhitungan persentase *error* yang dimiliki oleh sistem. Adapun rumus yang digunakan untuk melakukan persentase *error* dengan menggunakan MAPE seperti dibawah ini:

$$MAPE(\%) = \frac{1}{N} \sum_{i=1}^N |P_A^i - P_F^i / P_A^i| \times 100\% \dots\dots\dots(5.1)$$

Tabel 5.2 di bawah ini menunjukkan persentase *error* di setiap peramalan yang dilakukan dengan kenaikan permukaan air

Tabel 5.2. Perbandingan Data Target dan Data Hasil Prediksi tiap jam

Tanggal	Jam	Kenaikan permukaan air (m ³ /jam)		Error peramalan JST (%)
		Target	Peramalan JST	
06-02-2014	07-08	0	0	0
06-02-2014	08-09	0	0	0
06-02-2014	09-10	0	0	0
06-02-2014	10-11	0	0	0
06-02-2014	11-12	0	0	0
06-02-2014	12-13	0	0	0
06-02-2014	13-14	0	0	0
06-02-2014	14-15	0	0	0

06-02-2014	15-16	0	0	0
06-02-2014	16-17	0	0	0
06-02-2014	17-18	0	0	0
06-02-2014	18-19	0	0	0
06-02-2014	19-20	0	0	0
06-02-2014	20-21	0	0	0
06-02-2014	21-22	1	1	0
06-02-2014	22-23	0	0	0
06-02-2014	23-24	0	0	0
06-02-2014	00-01	0	0	0
06-02-2014	01-02	0	0	0
06-02-2014	02-03	0	0	0
06-02-2014	03-04	0	0	0
06-02-2014	04-05	0	0	0
06-02-2014	05-06	0	0	0
06-02-2014	06-07	0	0	0
07-02-2014	07-08	0	0	0
07-02-2014	08-09	0	0	0
07-02-2014	09-10	0	0	0
07-02-2014	10-11	0	0	0
07-02-2014	12-13	0	0	0
07-02-2014	13-14	0	0	0
07-02-2014	14-15	0	0	0
07-02-2014	15-16	0	0	0
07-02-2014	16-17	0	0	0
07-02-2014	17-18	0	0	0
07-02-2014	18-19	0	0	0
07-02-2014	19-20	0	0	0
07-02-2014	20-21	0	0	0
07-02-2014	21-22	0	0	0
07-20-2014	22-23	14	11	21.42
MAPE				0.5355
Error Minimum				0
Error Maksimum				21.42

Berdasarkan Tabel diatas dapat dikatakan hasil prediksi JST mendekati dengan data aktual. *Error* yang paling besar yakni 21.42%, dan *error* yang paling kecil adalah 0%. Selanjutnya untuk mengetahui seberapa *valid* hasil dari peramalan JST dapat menggunakan rumus tingkat akurasi yang telah dijelaskan sebelumnya pada persamaan (3.2), yakni:

$$\text{Tingkat akurasi}(\%) = \frac{\Sigma \text{data uji benar}}{\Sigma \text{jumlah total data uji}} \times 100\%$$

$$\text{Tingkat akurasi}(\%) = \frac{39}{40} \times 100\%$$

$$\text{Tingkat akurasi}(\%) = 0.975 \times 100\%$$

$$\text{Tingkat akurasi}(\%) = 97.5\%$$

Jadi, berdasarkan pengujian yang dilakukan terhadap sistem peramalan kenaikan permukaan air dengan *artificial neural network backpropagation* didapat bahwa sistem peramalan kenaikan permukaan air mempunyai MAPE sebesar 0.5355% dengan tingkat akurasi sebesar 97.5%.

5.7 Pengujian Parameter Jaringan

Berikut adalah hasil uji coba dari pengujian parameter jaringan. Hal ini dilakukan untuk memperoleh hasil dari parameter jaringan yang optimal. Hasil dari pengujian parameter jaringan dapat dilihat pada Tabel 5.3

Tabel 5.3. Pengujian Parameter Jaringan

No	Learning rate	Epoch	Momentum	Goal	Show iterasi	Maximum fail	Gradien minimum	Time	Ket
1	0.001	76	0.8	1e-10	20	6	1e-10	0:00:04	Baik
2	0.09	76	0.8	1e-10	20	6	1e-10	0:00:03	Baik
3	0.08	76	0.8	1e-10	20	6	1e-10	0:00:02	Baik
4	0.07	76	0.8	1e-10	20	6	1e-10	0:00:02	Baik
5	0.06	76	0.8	1e-10	20	6	1e-10	0:00:02	Baik
6	0.05	76	0.8	1e-10	20	6	1e-10	0:00:02	Baik
7	0.04	76	0.8	1e-10	20	6	1e-10	0:00:02	Baik
8	0.03	76	0.8	1e-10	20	6	1e-10	0:00:02	Baik
9	0.02	76	0.8	1e-10	20	6	1e-10	0:00:02	Baik
10	0.01	76	0.8	1e-10	20	6	1e-10	0:00:02	Baik
11	0.9	76	0.8	1e-10	20	6	1e-10	0:00:02	Baik
12	0.8	76	0.8	1e-10	20	6	1e-10	0:00:02	Baik
13	0.7	76	0.8	1e-10	20	6	1e-10	0:00:02	Baik
14	0.6	76	0.8	1e-10	20	6	1e-10	0:00:02	Baik
15	0.5	76	0.8	1e-10	20	6	1e-10	0:00:02	Baik
16	0.4	76	0.8	1e-10	20	6	1e-10	0:00:02	Baik
17	0.3	76	0.8	1e-10	20	6	1e-10	0:00:02	Baik

18	0.2	76	0.8	1e-10	20	6	1e-10	0:00:02	Baik
19	0.1	76	0.8	1e-10	20	6	1e-10	0:00:02	Baik
20	1	76	0.8	1e-10	20	6	1e-10	0:00:02	Baik
21	0.1	200	0.8	1e-10	20	6	1e-10	0:00:02	Baik
22	0.1	50	0.8	1e-10	20	6	1e-10	0:00:02	Kurang baik
23	0.1	76	0.8	1e-10	20	6	1e-10	0:00:02	Sangat baik
24	0.1	76	0.5	1e-10	20	6	1e-10	0:00:02	Baik
25	0.1	76	0.6	1e-10	20	6	1e-10	0:00:02	Baik
26	0.1	76	0.7	1e-10	20	6	1e-10	0:00:02	Baik
26	0.1	76	0.8	1e-10	20	6	1e-10	0:00:02	Sangat baik
27	0.1	76	0.9	1e-10	20	6	1e-10	0:00:02	Baik
28	0.1	1	0.8	1e-1	20	6	1e-10	0:00:00	Kurang baik
29	0.1	2	0.8	1e-2	20	6	1e-10	0:00:02	Kurang baik
30	0.1	3	0.8	1e-3	20	6	1e-10	0:00:00	Kurang baik
31	0.1	4	0.8	1e-4	20	6	1e-10	0:00:00	Kurang baik
32	0.1	5	0.8	1e-5	20	6	1e-10	0:00:00	Baik
33	0.1	11	0.8	1e-6	20	6	1e-10	0:00:00	Baik
34	0.1	21	0.8	1e-7	20	6	1e-10	0:00:00	Baik
35	0.1	28	0.8	1e-8	20	6	1e-10	0:00:01	Baik
36	0.1	50	0.8	1e-9	20	6	1e-10	0:00:01	Baik
37	0.1	76	0.8	1e-10	20	6	1e-10	0:00:02	Sangat baik
38	0.1	76	0.8	1e-10	5	6	1e-10	0:00:02	Baik
39	0.1	76	0.8	1e-10	10	6	1e-10	0:00:02	Baik
40	0.1	76	0.8	1e-10	15	6	1e-10	0:00:02	Baik
41	0.1	76	0.8	1e-10	20	6	1e-10	0:00:02	Sangat baik
42	0.1	76	0.8	1e-10	20	1	1e-10	0:00:02	Baik
43	0.1	76	0.8	1e-10	20	2	1e-10	0:00:02	Baik
44	0.1	76	0.8	1e-10	20	3	1e-10	0:00:02	Baik
45	0.1	76	0.8	1e-10	20	4	1e-10	0:00:02	Baik
46	0.1	76	0.8	1e-10	20	5	1e-10	0:00:02	Baik
47	0.1	76	0.8	1e-10	20	6	1e-10	0:00:02	Sangat baik
48	0.1	3	0.8	1e-10	20	6	1e-1	0:00:00	Kurang baik
49	0.1	4	0.8	1e-10	20	6	1e-2	0:00:00	Kurang baik
50	0.1	10	0.8	1e-10	20	6	1e-3	0:00:00	Baik
51	0.1	19	0.8	1e-10	20	6	1e-4	0:00:00	Baik
52	0.1	43	0.8	1e-10	20	6	1e-5	0:00:02	Baik
53	0.1	47	0.8	1e-10	20	6	1e-6	0:00:01	Baik
54	0.1	50	0.8	1e-10	20	6	1e-7	0:00:02	Baik
55	0.1	53	0.8	1e-10	20	6	1e-8	0:00:02	Baik
56	0.1	60	0.8	1e-10	20	6	1e-9	0:00:02	Baik
57	0.1	76	0.8	1e-10	20	6	1e-10	0:00:02	Sangat baik

Berdasarkan hasil pengujian parameter jaringan pada Tabel 5.2. didapat bahwa jaringan akan optimal dengan nilai-nilai parameter sebagai berikut:

Learning rate : 0.1
Epoch : 76
Momentum : 0.8
Goal : 1e-10
Show iterasi : 20
Maximum fail : 6
Gradien minimum : 1e-10

Serta jaringan tidak akan optimal dengan nilai-nilai parameter sebagai berikut:

a. *Learning rate* : 0.1
Epoch : 50
Momentum : 0.8
Goal : 1e-10
Show iterasi : 20
Maximum fail : 6
Gradien minimum : 1e-10

b. *Learning rate* : 0.1
Epoch : 50
Momentum : 0.8
Goal : 1e-1
Show iterasi : 20
Maximum fail : 6
Gradien minimum : 1e-10

- c. *Learning rate* : 0.1
 - Epoch* : 50
 - Momentum* : 0.8
 - Goal* : 1e-2
 - Show iterasi* : 20
 - Maximum fail* : 6
 - Gradien minimum* : 1e-10
- d. *Learning rate* : 0.1
 - Epoch* : 50
 - Momentum* : 0.8
 - Goal* : 1e-3
 - Show iterasi* : 20
 - Maximum fail* : 6
 - Gradien minimum* : 1e-10
- e. *Learning rate* : 0.1
 - Epoch* : 50
 - Momentum* : 0.8
 - Goal* : 1e-4
 - Show iterasi* : 20
 - Maximum fail* : 6
 - Gradien minimum* : 1e-10
- f. *Learning rate* : 0.1
 - Epoch* : 50
 - Momentum* : 0.8

Goal : 1e-10
Show iterasi : 20
Maximum fail : 6
Gradien minimum : 1e-1
g. *Learning rate* : 0.1
Epoch : 50
Momentum : 0.8
Goal : 1e-10
Show iterasi : 20
Maximum fail : 6
Gradien minimum : 1e-2

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil dan analisa dari bab sebelumnya maka penulis dapat mengambil kesimpulan sebagai berikut:

1. Sistem Peramalan Kenaikan Permukaan Air dengan *Artificial Neural Network Backpropagation* dibangun dengan menggunakan metode Pemrograman Terstruktur dan menggunakan *Data Flow Diagram* (DFD) serta menggunakan model pengembangan sistem *sekuensial linier* yang bersifat sistematis dan berurutan.
2. Sistem Peramalan Kenaikan Permukaan Air dengan *Artificial Neural Network Backpropagation* yang dibangun mampu melakukan peramalan terhadap kenaikan permukaan air, hal ini dibuktikan dengan tingkat akurasi yang dihasilkan pada tanggal 7 Februari 2014 sebesar 97.5% dengan MAPE sebesar 0.5355%.
3. Jaringan akan optimal dengan parameter jaringan:

<i>Learning rate</i>	: 0.1
<i>Epoch</i>	: 76
<i>Momentum</i>	: 0.8
<i>Goal</i>	: 1e-10
<i>Show iterasi</i>	: 20
<i>Maximum fail</i>	: 6
<i>Gradien minimum</i>	: 1e-10

6.2 Saran

Saran yang penulis ajukan untuk memperbaiki dan pengembangan lebih lanjut mengenai sistem ini adalah sebagai berikut:

1. Pengembangan Sistem Informasi Peramalan Kenaikan Permukaan Air dengan *Artificial Neural Network Backpropagation* ini perlu memperhatikan pembagian data untuk data latih, data uji, dan data ramal, agar tidak terjadi *overheading* dalam pelatihan Jaringan Syaraf Tiruan nantinya.
2. Mengembangkan dengan metode pembelajaran yang lebih sempurna, dimana data *feedback* dan *error* peramalan dijadikan salah satu variabel pembelajaran selanjutnya.
3. Penentuan nilai parameter jaringan agar mendapatkan hasil yang optimal

DAFTAR PUSTAKA

- Amaliah, B. d. (2011). Penentuan Jenis Produk Kosmetik Pilihan Berdasarkan Faktor Usia Dan Warna Kulit Menggunakan Jaringan Syaraf Tiruan. <http://eprints.undip.ac.id/252801/ML2F/002572.pdf>.
- Irawan, A. (2012). Perbandingan Metode Jaringan Syaraf Tiruan Backpropagation Dan Learning Vector Quantitation Pada Pengenalan Wajah . <http://komputa.if.unikom.ac.id> .
- Jaya, I. d. (2007). Identifikasi Kawanan Ikan Lemuru Dari Data Hidroakustik Dengan Metode Jaringan Syaraf Tiruan. <http://repository.unhas.ac.id> .
- Jek Siang, J. (2005). *Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan MATLAB*. Yogyakarta: ANDI.
- Jogiyanto. (2005). *Analisis dan Desain Sistem Informasi : Pendekatan Terstruktur Teori dan Praktik Aplikasi Bisnis*. Yogyakarta: ANDI.
- Ketahun, B. (2013). *Laporan Hasil Monitoring Curah Hujan di Provinsi Bengkulu*. Bengkulu: BMKG Propinsi Bengkulu.
- Kusumadewi, S. (2004). *Artificial Intelligent (Teknik dan Aplikasinya)*. Jakarta: Graha Ilmu.
- Nasution, M. I. (2011). Pengertian Curah Hujan. <http://repository.usu.ac.id/bitstream/123456789/19244/4/Chapter%2011.pdf>.
- Novikaginanto. (2012, November 14). Dipetik November 14, 2012, dari <http://novikaginanto.wordpress.com>.
- Pratiwi, S. (2011). *Sistem Informasi Penerimaan Keuangan Universitas Bengkulu*. Bengkulu.
- Pressman, R. S. (2002). *Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku Satu)*. Yogyakarta: ANDI.
- Salman, A. G. (2011). Implementasi Jaringan Syaraf Tiruan Recurrent Dengan Metode Pembelajaran Gradient Descent Adaptive Learning Rate Untuk Pendugaan Curah Hujan. <http://journal.uui.ac.id/index.php/Snati/article/viewFile/2222/2167> .
- Setyawan, H. D. (2008). Peramalan Ketinggian Muka Air Dengan Jaringan Syaraf Tiruan Perambatan-Balik. [Eprints.Undip.ac.id/25353/2/ML2F003506.pdf](http://eprints.Undip.ac.id/25353/2/ML2F003506.pdf) .

- Soemarno. (2007). Penelitian.
<http://images.soemarno.multiply.com/attachment/0/RvmpdAoKCtgAA/BUBWx01/PENELITIAN.ppt?nmid=59416790> .
- Syarif, A. (2009). Perancangan Sistem Informasi Berbasis WEB dengan Menggunakan PHP & MYSQL di Program Studi Ilmu Keperawatan Universitas Sumatera Utara.
<http://repository.usu.ac.id/bitstream/123456789/7819/1/09E02119.pdf>
- Trisnawan, B. (2013). Jaringan Syaraf Tiruan. <http://bayu-tfst10.web.unair.ac.id> .
- Widodo, P. P. (2013). *Penerapan Data Mining Dengan Matlab*. Bandung: Rekayasa Sains.
- Wijayaning, R. d. (2009). *Modul Pratikum Rekayasa Perangkat Lunak*. Yogyakarta: Universitas Islam Indonesia.
- Windarto, J. d. (2008). Model Prediksi Tinggi Muka Air Sungai Kali Garang Semarang Dengan Jaringan Syaraf Tiruan . *Portalgaruda.org* .

LAMPIRAN

LAMPIRAN 1

Perhitungan Manual Data Pelatihan

Berikut adalah data *training* yang disimpan dalam *database*. Sampel yang diambil sebanyak 24 data latih (sebelum dilakukan normalisasi)

No	Koefisien	Intensitas (mm/jam)	Luas (km ²)	Debit (m ³ /jam)
1	0.35	0	9.5397	0
2	0.35	0	9.5397	0
3	0.35	0	9.5397	0
4	0.35	0	9.5397	0
5	0.35	0	9.5397	0
6	0.35	0	9.5397	0
7	0.35	0	9.5397	0
8	0.35	0	9.5397	0
9	0.35	0.1	9.5397	0.3338895
10	0.35	0.8	9.5397	2.671116
11	0.35	0	9.5397	0
12	0.35	0.1	9.5397	0.3338895
13	0.35	2.0	9.5397	6.67779
14	0.35	2.1	9.5397	7.0116795
15	0.35	0.1	9.5397	0.3338895
16	0.35	0	9.5397	0
17	0.35	0	9.5397	0
18	0.35	0	9.5397	0
19	0.35	0	9.5397	0
20	0.35	0	9.5397	0
21	0.35	0	9.5397	0
22	0.35	0	9.5397	0
23	0.35	0	9.5397	0
24	0.35	0	9.5397	0

Sebelum dilakukan *preprocessing*, maka data *input* dan target dilakukan *transpose* terlebih dahulu (baris menjadi kolom, kolom menjadi baris)

$$p_n = (0.8 * (p - \min_p) / (\max_p - \min_p)) + 0.1$$

$$p_1 = 0$$

$$p_n = (0.8 * (0 - 0) / (180.9681 - 0)) + 0.1$$

$$p_n = 0.1$$

Berikut adalah hasil *preprocessing* data input yang disimpan pada matriks pn

pn=

Kolom 1 Sampai 16

0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1015	0.1118
0.1000	0.1015	0.1295	0.1310	0.1015	0.1000				

Kolom 17 sampai 24

0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
--------	--------	--------	--------	--------	--------	--------	--------

LAMPIRAN 2

Perhitungan Manual Data Uji

Berikut adalah data *testing* yang disimpan di dalam database (sebelum dilakukan normalisasi)

No	Koefisien	Intensitas (mm/jam)	Luas (km ²)	Debit (m ³ /jam)
1	0.35	0	9.5397	0
2	0.35	0	9.5397	0
3	0.35	0	9.5397	0
4	0.35	0	9.5397	0
5	0.35	0	9.5397	0
6	0.35	0	9.5397	0
7	0.35	0	9.5397	0
8	0.35	0	9.5397	0
9	0.35	0	9.5397	0
10	0.35	0	9.5397	0
11	0.35	0	9.5397	0
12	0.35	0	9.5397	0
13	0.35	0	9.5397	0
14	0.35	0	9.5397	0
15	0.35	0	9.5397	0
16	0.35	0	9.5397	0
17	0.35	0	9.5397	0
18	0.35	0	9.5397	0
19	0.35	0	9.5397	0
20	0.35	0	9.5397	0
21	0.35	0	9.5397	0
22	0.35	0	9.5397	0
23	0.35	0	9.5397	0
24	0.35	0	9.5397	0
25	0.35	0	9.5397	0
26	0.35	0	9.5397	0
27	0.35	0	9.5397	0
28	0.35	0	9.5397	0
29	0.35	0	9.5397	0
30	0.35	0	9.5397	0
31	0.35	0	9.5397	0
32	0.35	0	9.5397	0
33	0.35	0	9.5397	0
34	0.35	0	9.5397	0
35	0.35	0	9.5397	0
36	0.35	0	9.5397	0
37	0.35	0	9.5397	0
38	0.35	0	9.5397	0
39	0.35	0	9.5397	0
40	0.35	0	9.5397	0

41	0.35	0	9.5397	0
42	0.35	0	9.5397	0
43	0.35	0.4	9.5397	1.3355579999999998
44	0.35	0	9.5397	0
45	0.35	0.5	9.5397	1.6694475
46	0.35	0	9.5397	0
47	0.35	0	9.5397	0
48	0.35	0	9.5397	0
49	0.35	0	9.5397	0
50	0.35	0	9.5397	0
51	0.35	0	9.5397	0
52	0.35	0	9.5397	0
53	0.35	0	9.5397	0
54	0.35	0	9.5397	0
55	0.35	0	9.5397	0
56	0.35	0	9.5397	0
57	0.35	0	9.5397	0
58	0.35	0	9.5397	0
59	0.35	0	9.5397	0
60	0.35	0	9.5397	0
61	0.35	0	9.5397	0
62	0.35	0	9.5397	0
63	0.35	0	9.5397	0
64	0.35	0	9.5397	0
65	0.35	0	9.5397	0
66	0.35	0	9.5397	0
67	0.35	0	9.5397	0
68	0.35	0	9.5397	0
69	0.35	0	9.5397	0
70	0.35	0	9.5397	0
71	0.35	0	9.5397	0
72	0.35	0	9.5397	0
73	0.35	1.1	9.5397	3.6727845
74	0.35	0	9.5397	0
75	0.35	0	9.5397	0
76	0.35	0	9.5397	0
77	0.35	0	9.5397	0
78	0.35	0	9.5397	0
79	0.35	0	9.5397	0
80	0.35	0	9.5397	0
81	0.35	0	9.5397	0
82	0.35	0	9.5397	0
83	0.35	0	9.5397	0
84	0.35	0	9.5397	0
85	0.35	0	9.5397	0
86	0.35	0.1	9.5397	0.33388949999999995
87	0.35	0.1	9.5397	0.33388949999999995

88	0.35	0	9.5397	0
89	0.35	0.6	9.5397	2.0033369999999997
90	0.35	0	9.5397	0
91	0.35	0	9.5397	0
92	0.35	0	9.5397	0
93	0.35	0.8	9.5397	2.6711159999999996
94	0.35	0.2	9.5397	0.6677789999999999
95	0.35	0	9.5397	0
96	0.35	0	9.5397	0

minimum Q (min_Q) = 0

Nilai maksimum Q (max_Q) = 3.6728

Sebelum dilakukan *preprocessing*, maka data *input* dan target dilakukan *transpose* terlebih dahulu (baris menjadi kolom, kolom menjadi baris)

$$Q_n = (0.8 * (Q - min_Q) / (max_Q - min_Q)) + 0.1$$

$$Q_1 = 0$$

$$Q_n = (0.8 * (0 - 0) / (180.9681 - 0)) + 0.1$$

$$Q_n = 0.1$$

Berikut adalah hasil *preprocessing* data input yang disimpan pada matriks Q_n

$Q_n =$

Kolom 1 Sampai 16

0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000
0.1000 0.1000 0.1000 0.1000 0.1000 0.1000

Kolom 17 Sampai 32

0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000
0.1000 0.1000 0.1000 0.1000 0.1000 0.1000

Kolom 33 Sampai 48

0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000
0.3909 0.1000 0.4636 0.1000 0.1000 0.1000

Kolom 49 Sampai 64

0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000
0.1000 0.1000 0.1000 0.1000 0.1000 0.1000

Kolom 65 Sampai 80

```

0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.9000 0.1000
0.1000 0.1000 0.1000 0.1000 0.1000 0.1000

```

Kolom 81 Sampai 96

```

0.1000 0.1000 0.1000 0.1000 0.1000 0.1727 0.1727 0.1000 0.5364 0.1000
0.1000 0.1000 0.6818 0.2455 0.1000 0.1000

```

$$T_n = (0.8 * (t - \min_t) / (\max_t - \min_t)) + 0.1$$

$$\min_T = 0$$

$$\max_T = 3.6728$$

$$T_1 = 0$$

$$T_n = (0.8 * (0 - 0) / (180.9681 - 0)) + 0.1$$

$$T_n = 0.1$$

Berikut adalah hasil preprocessing data input yang disimpan pada matriks tn

Tn =

Kolom 1 Sampai 16

```

0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000
0.1000 0.1000 0.1000 0.1000 0.1000 0.1000

```

Kolom 17 Sampai 32

```

0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000
0.1000 0.1000 0.1000 0.1000 0.1000 0.1000

```

Kolom 33 Sampai 48

```

0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000
0.3909 0.1000 0.4636 0.1000 0.1000 0.1000

```

Kolom 49 Sampai 64

```

0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000
0.1000 0.1000 0.1000 0.1000 0.1000 0.1000

```

Kolom 65 Sampai 80

```

0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.9000 0.1000
0.1000 0.1000 0.1000 0.1000 0.1000 0.1000

```

Kolom 81 Sampai 96

0.1000 0.1000 0.1000 0.1000 0.1000 0.1727 0.1727 0.1000 0.5364 0.1000
0.1000 0.1000 0.6818 0.2455 0.1000 0.1000

LAMPIRAN 3

Perhitungan Manual Data Ramal

Berikut adalah data *forecasting* yang disimpan di dalam *database* (sebelum dilakukan normalisasi)

No	Koefisien	Intensitas (mm/jam)	Luas (km ²)	Debit (m ³ /jam)
1	0.35	0	9.5397	0
2	0.35	0	9.5397	0
3	0.35	0	9.5397	0
4	0.35	0	9.5397	0
5	0.35	0	9.5397	0
6	0.35	0	9.5397	0
7	0.35	0	9.5397	0
8	0.35	7	9.5397	23.372265
9	0.35	2.5	9.5397	8.3472375
10	0.35	0	9.5397	0
11	0.35	0.3	9.5397	1.0016684999999999
12	0.35	0.2	9.5397	0.6677789999999999
13	0.35	0	9.5397	0
14	0.35	0	9.5397	0
15	0.35	4.3	9.5397	14.357248499999999
16	0.35	0	9.5397	0
17	0.35	0.1	9.5397	0.33388949999999995
18	0.35	0	9.5397	0
19	0.35	0	9.5397	0
20	0.35	0	9.5397	0
21	0.35	0	9.5397	0
22	0.35	0	9.5397	0
23	0.35	0	9.5397	0
24	0.35	0	9.5397	0
25	0.35	0	9.5397	0
26	0.35	0	9.5397	0
27	0.35	0	9.5397	0
28	0.35	0	9.5397	0
29	0.35	0	9.5397	0
30	0.35	0	9.5397	0
31	0.35	0	9.5397	0
32	0.35	0	9.5397	0
33	0.35	0	9.5397	0
34	0.35	0	9.5397	0
35	0.35	0	9.5397	0
36	0.35	0	9.5397	0
37	0.35	0	9.5397	0
38	0.35	0	9.5397	0
39	0.35	0.3	9.5397	1.0016684999999999

40	0.35	0.1	9,5397	0.33388949999999995
----	------	-----	--------	---------------------

Dari tabel diatas diketahui nilai minimum dari data yang ada di dalam tabel dataramal adalah 0, nilai maksimumnya adalah 23.3723

$$\text{Min}_u = 0$$

$$\text{Max}_u = 23.3723$$

$$U_n = (0.8 * (u - \text{min}_u) / (\text{max}_u - \text{min}_u)) + 0.1$$

$$U_n (\text{data ke-1} = 0) = (0.8 * (0 - 0) / (23.3723 - 0)) + 0.1$$

$$U_n (\text{data ke-1} = 0) = 0.1$$

Dilakukan *transpose* pada dataramal, sehingga kolom menjadi baris, baris menjadi kolom
 $u_n =$

Kolom 1 Sampai 16

0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.9000 0.3857 0.1000
0.1343 0.1229 0.1000 0.1000 0.5914 0.1000

Kolom 17 Sampai 32

0.1114 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000
0.1000 0.1000 0.1000 0.1000 0.1000 0.1000

Kolom 33 Sampai 40

0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1343 0.1114

Setelah dilakukan *preprocessing*, dilakukan *postprocessing* dengan tujuan untuk mengembalikan kebentuk debit banjir, dengan rumus:

$$\text{Min}_y = 0$$

$$\text{Max}_y = 0.9$$

$$y_s =$$

Kolom1 Sampai 16

0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000
0.1000 0.1000 0.1000 0.1000 0.1598 0.1190

Kolom17 Sampai 32

0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000
0.1000 0.1000 0.1000 0.1000 0.1000 0.1000

Kolom33 Sampai 40

0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.1000 0.9000

```
un3=round((ys-0.1).*(max_y-min_y)-(0.8*min_y)/0.8)
```

```
un3 (data ke-1) = 0.1
```

```
un3 = round((0.1-0.1).*(0.9-0) - (0.8*0) / 0.8)
```

```
un3 = round ((0)*(0) - (0) / 0.8) = 0
```

sehingga un3 setelah dilakukan denormalisasi (*postprocessing*) menjadi:

un3 =

Kolom 1 Sampai 27

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0

Kolom 28 Sampai 40

0 0 0 0 0 0 0 0 0 0 0 0 0 11

LAMPIRAN 4

Pengujian Arsitektur Jaringan

Learning rate : 0.1

Epoch : 76

Momentum : 0.8

Goal : $1e-10$

Show iterasi : 20

Maximum fail : 6

Gradien minimum : $1e-10$

latih

input data

Koefisien (0-1 %)

Intensitas

Luas

Target

Simpan Latih

Simpan Uji

Simpan Ramal

Validasi

Pilihan

LATIH

KEMBALI

UJI

KELUAR

PREDIKSI

Parameter Jaringan

Learning Rate 0.1 0.001 - 1

Epoch 76

Momentum 0.8 0.5 - 0.9

Goal 1e-10

Show Iterasi 20

Maximum Fail 6

Gradien Minimum 1e-10

Neural Network Training (nntraintool)

Neural Network

Algorithms

Training: Levenberg-Marquardt (trainlm)
 Performance: Mean Squared Error (mse)

Progress

Epoch:	0	76 iterations	76
Time:		0:00:03	
Performance:	0.280	9.82e-11	1.00e-10
Gradient:	1.00	1.27e-06	1.00e-10
Mu:	0.00100	0.000100	1.00e+10
Validation Checks:	0	0	6

Plots

Performance (plotperform)
 Training State (plottrainstate)
 Regression (plotregression)

Plot Interval: 1 epochs

Performance goal met.

