LAMPIRAN

Lampiran A Data Awal Penelitian

Tabel A.1 Energi Terjual PLN Menurut Jenis Tarif di Provinsi Banten (GWh) 2011 s.d. 2020

Tahun	Rt	Bisnis	Publik	Industri
2011	3681	1931	404.02	11471
2012	4050	1755	410.52	12353
2013	3640	2299	419.08	12920
2014	3892	2243	438.00	12569
2015	4370	2147	440.65	11645
2016	4543	2344	469.10	12811
2017	4599	2521	500.63	13623
2018	4825	2736	543.67	14803
2019	5231	2930	587.46	14601
2020	5794	2890	556.62	13027

Tabel A.2 Jumlah Pelanggan PLN Menurut Klasifikasi Tarif di Provinsi Banten (Ribu) 2011 s.d. 2020

Tahun	Rt	Bisnis	Publik	Industri
2011	1659	74	39	5.2
2012	1795	84	42	5.3
2013	1992	93	45	5.6
2014	2178	101	47	5.7
2015	2399	111	52	6
2016	2548	132	56	6.1
2017	2712	152	62	6.3
2018	2894	174	62	6.3
2019	3078	175	76	6.7
2020	3224	191	76	6.8

Tabel A.3 Jumlah Penduduk dan PDRB Provinsi Banten 2011 s.d. 2020

Tahun	PDRB	Penduduk
2011	290545838.9	11005518
2012	310385592.4	11248947
2013	331099105.5	11452491
2014	349584667.8	11704877
2015	369209288.1	11955243
2016	389543932.3	12203148
2017	412639618.3	12448160
2018	436581428.1	12689736
2019	459828497.4	12927316
2020	445672956.7	11904562

Lampiran B Listing Code

Lampiran *listing code* metode jaringan syaraf tiruan untuk mendapatkan nilai prakiraan kebutuhan energi listrik Provinsi Banten di MATLAB.

B1. *Training Code* Model 1

```
% Membaca data dari excel
filename = 'data_fix.xlsx';
sheet = 1;
xlRange = 'O5:X14';

Data = xlsread(filename, sheet, xlRange);
data_latih = Data(:,1:6)';
target_latih = Data(:,7:10)';
[m,n] = size(data_latih);

% Pembuatan neural networks feedforward
net = newff(minmax(data_latih), [6 4], {'tansig', 'purelin'},
'trainlm');

net.performFcn = 'mse';
net.trainParam.goal = 0.01;
net.trainParam.show = 25;
net.trainParam.epochs = 1000;
net.trainParam.mc = 0.1;
net.trainParam.lr = 0.001;

% Proses training
[net_keluaran,tr,Y,E] = train(net,data_latih,target_latih);

% Hasil setelah pelatihan
bobot_hidden = net_keluaran.IW{1,1};
bobot_keluaran = net_keluaran.LW{2,1};
bias_hiddan = net_keluaran.b{1,1};
bias_keluaran = net_keluaran.b{2,1};
jumlah_iterasi = tr.num_epochs;
nilai_keluaran = Y;
nilai_error = E;
error_MSE = (1/n)*sum(nilai_error.^2);

save('C:\Users\Lenovo\Documents\fix\pelatihan_6.mat')

% Hasil pelatihan
hasil_latih = sim(net_keluaran,data_latih);
% Performansi hasil pelatihan
target_latih_asli = target_latih;

figure,
plotregression(target_latih_asli,hasil_latih,'Regression')
figure,
plotperform(tr)
```

Testing Code

```
%memanggil bobot dan bias hasil pelatihan
```

```
load('C:\Users\Lenovo\Documents\fix\pelatihan_6.mat')

% Membaca data dari excel
filename = 'data_fix.xlsx';
sheet = 1;
xlRange = 'O21:T30';

Data = xlsread(filename, sheet, xlRange);
data_uji = Data(:,1:6)';
[m,n] = size(data_uji);

% Hasil PREDIKSI
hasil_uji = sim(net_keluaran,data_uji);
```

B2. *Training Code* Model 2

```
% Membaca data dari excel
filename = 'data_fix.xlsx';
sheet = 1;
xlRange = 'O5:X14';

Data = xlsread(filename, sheet, xlRange);
data_latih = Data(:,1:6)';
target_latih = Data(:,7:10)';
[m,n] = size(data_latih);

% Pembuatan neural networks feedforward backpropagation
net = newff(minmax(data_latih), [12 4], {'tansig', 'purelin'},
'trainlm');

net.performFcn = 'mse';
net.trainParam.goal = 0.01;
net.trainParam.show = 25;
net.trainParam.epochs = 1000;
net.trainParam.mc = 0.1;
net.trainParam.lr = 0.001;

% Proses training
[net_keluaran,tr,Y,E] = train(net,data_latih,target_latih);

% Hasil setelah pelatihan
bobot_hidden = net_keluaran.IW{1,1};
bobot_keluaran = net_keluaran.LW{2,1};
bias_hiddan = net_keluaran.b{1,1};
bias_keluaran = net_keluaran.b{2,1};
jumlah_iterasi = tr.num_epochs;
nilai_keluaran = Y;
nilai_error = E;
error_MSE = (1/n)*sum(nilai_error.^2);

save('C:\Users\Lenovo\Documents\fix\pelatihan_12.mat')

% Hasil pelatihan
hasil_latih = sim(net_keluaran,data_latih);
% Performansi hasil pelatihan
target_latih_asli = target_latih;
```

```
figure,
plotregression(target_latih_asli,hasil_latih,'Regression')
figure,
plotperform(tr)
```

Testing Code

```
%memanggil bobot dan bias hasil pelatihan
load('C:\Users\Lenovo\Documents\fix\pelatihan_12.mat')

% Membaca data dari excel
filename = 'data_fix.xlsx';
sheet = 1;
xlRange = 'O21:T30';

Data = xlsread(filename, sheet, xlRange);
data_uji = Data(:,1:6)';
[m,n] = size(data_uji);

% Hasil PREDIKSI
hasil_uji = sim(net_keluaran,data_uji);
```

B3. *Training Code* Model 3

```
% Membaca data dari excel
filename = 'data_fix.xlsx';
sheet = 1;
xlRange = 'O5:X14';

Data = xlsread(filename, sheet, xlRange);
data_latih = Data(:,1:6)';
target_latih = Data(:,7:10)';
[m,n] = size(data_latih);

% Pembuatan neural networks feedforward backpropagation
net = newff(minmax(data_latih), [18 4], {'tansig', 'purelin'},
'trainlm');

net.performFcn = 'mse';
net.trainParam.goal = 0.01;
net.trainParam.show = 25;
net.trainParam.epochs = 1000;
net.trainParam.mc = 0.1;
net.trainParam.lr = 0.001;

% Proses training
[net_keluaran,tr,Y,E] = train(net,data_latih,target_latih);

% Hasil setelah pelatihan
bobot_hidden = net_keluaran.IW{1,1};
bobot_keluaran = net_keluaran.LW{2,1};
bias_hiddan = net_keluaran.b{1,1};
bias_keluaran = net_keluaran.b{2,1};
jumlah_iterasi = tr.num_epochs;
nilai_keluaran = Y;
nilai_error = E;
```

```
error_MSE = (1/n)*sum(nilai_error.^2);

save('C:\Users\Lenovo\Documents\fix\pelatihan_18.mat')

% Hasil pelatihan
hasil_latih = sim(net_keluaran,data_latih);
% Performansi hasil pelatihan
target_latih_asli = target_latih;

figure,
plotregression(target_latih_asli,hasil_latih,'Regression')
figure,
plotperform(tr)
```

Testing Code

```
%memanggil bobot dan bias hasil pelatihan
load('C:\Users\Lenovo\Documents\fix\pelatihan_18.mat')

% Membaca data dari excel
filename = 'data_fix.xlsx';
sheet = 1;
xlRange = 'O21:T30';

Data = xlsread(filename, sheet, xlRange);
data_uji = Data(:,1:6)';
[m,n] = size(data_uji);

% Hasil PREDIKSI
hasil_uji = sim(net_keluaran,data_uji);
```

B4. *Training Code* Model 4

```
% Membaca data dari excel
filename = 'data_fix.xlsx';
sheet = 1;
xlRange = 'O5:X14';

Data = xlsread(filename, sheet, xlRange);
data_latih = Data(:,1:6)';
target_latih = Data(:,7:10)';
[m,n] = size(data_latih);

% Pembuatan neural networks feedforward backpropagation
net = newff(minmax(data_latih), [24 4], {'tansig', 'purelin'},
'trainlm');

net.performFcn = 'mse';
net.trainParam.goal = 0.01;
net.trainParam.show = 25;
net.trainParam.epochs = 1000;
net.trainParam.mc = 0.1;
net.trainParam.lr = 0.001;

% Proses training
```

```
[net_keluaran,tr,Y,E] = train(net,data_latih,target_latih);

% Hasil setelah pelatihan
bobot_hidden = net_keluaran.IW{1,1};
bobot_keluaran = net_keluaran.LW{2,1};
bias_hiddan = net_keluaran.b{1,1};
bias_keluaran = net_keluaran.b{2,1};
jumlah_iterasi = tr.num_epochs;
nilai_keluaran = Y;
nilai_error = E;
error_MSE = (1/n)*sum(nilai_error.^2);

save('C:\Users\Lenovo\Documents\fix\pelatihan_24.mat')

% Hasil pelatihan
hasil_latih = sim(net_keluaran,data_latih);
% Performansi hasil pelatihan
target_latih_asli = target_latih;

figure,
plotregression(target_latih_asli,hasil_latih,'Regression')
figure,
plotperform(tr)
```

Testing Code

```
%memanggil bobot dan bias hasil pelatiha
load('C:\Users\Lenovo\Documents\fix\pelatihan_24.mat')

% Membaca data dari excel
filename = 'data_fix.xlsx';
sheet = 1;
xlRange = 'O21:T30';

Data = xlsread(filename, sheet, xlRange);
data_uji = Data(:,1:6)';
[m,n] = size(data_uji);

% Hasil PREDIKSI
hasil_uji = sim(net_keluaran,data_uji);
```

B5. *Training Code* Model 5

```
% Membaca data dari excel
filename = 'data_fix.xlsx';
sheet = 1;
xlRange = 'O5:X14';

Data = xlsread(filename, sheet, xlRange);
data_latih = Data(:,1:6)';
target_latih = Data(:,7:10)';
[m,n] = size(data_latih);

% Pembuatan neural networks feedforward backpropagation
net = newff(minmax(data_latih), [30 4], {'tansig', 'purelin'},
'trainlm');
```

```
net.performFcn = 'mse';
net.trainParam.goal = 0.01;
net.trainParam.show = 25;
net.trainParam.epochs = 1000;
net.trainParam.mc = 0.1;
net.trainParam.lr = 0.001;

% Proses training
[net_keluaran,tr,Y,E] = train(net,data_latih,target_latih);

% Hasil setelah pelatihan
bobot_hidden = net_keluaran.IW{1,1};
bobot_keluaran = net_keluaran.LW{2,1};
bias_hiddan = net_keluaran.b{1,1};
bias_keluaran = net_keluaran.b{2,1};
jumlah_iterasi = tr.num_epochs;
nilai_keluaran = Y;
nilai_error = E;
error_MSE = (1/n)*sum(nilai_error.^2);

save('C:\Users\Lenovo\Documents\fix\pelatihan_30.mat')

% Hasil pelatihan
hasil_latih = sim(net_keluaran,data_latih);
% Performansi hasil pelatihan
target_latih_asli = target_latih;

figure,
plotregression(target_latih_asli,hasil_latih,'Regression')
figure,
plotperform(tr)
```

Testing Code

```
%memanggil bobot dan bias hasil pelatiha
load('C:\Users\Lenovo\Documents\fix\pelatihan_30.mat')

% Membaca data dari excel
filename = 'data_fix.xlsx';
sheet = 1;
xlRange = 'O21:T30';

Data = xlsread(filename, sheet, xlRange);
data_uji = Data(:,1:6)';
[m,n] = size(data_uji);

% Hasil PREDIKSI
hasil_uji = sim(net_keluaran,data_uji);
```