

Implementasi Algoritma *Backpropagation* Dalam Memprediksi Jumlah Penduduk Usia Produktif Pada Kota Pematangsiantar

Mhd Ridho Azhar, Sumarno, Indra Gunawan

Teknik Informatika, STIKOM Tunas Bangsa, Pematangsiantar, Indonesia

Email: ¹ridho.azhar1997@gmail.com, ²sumarno@amiktunasbangsa.ac.id, ³indra@amiktunasbangsa.ac.id

Submitted: 15/02/2021; Accepted: 26/02/2021; Published: 27/02/2021

Abstrak—Usia produktif sendiri merupakan penduduk pada kelompok usia antara 15 hingga 64 tahun, baik bekerja, sekolah, dan mengurus rumah tangga, dalam hal ini individu yang berada pada ruang lingkup usia produktif adalah orang-orang yang masih dapat bekerja dengan baik untuk menghasilkan suatu produk dan jasa. Penelitian ini menggunakan Jaringan Syaraf Tiruan (JST) dengan metode *backpropagation*. Algoritma *backpropagation* merupakan salah satu metode yang ada pada jaringan syaraf tiruan sebagai algoritma prediksi, estimasi, klasifikasi, dan pengenalan pola. Data penelitian merupakan data sekunder yang bersumber dari Badan Pusat Statistik (BPS) tahun 2013 sampai dengan 2015. Data dibagi menjadi 2 bagian yakni data pelatihan dan pengujian. Ada 5 model arsitektur yang digunakan pada penelitian ini. 2-20-1, 2-21-1, 2-22-1, 2-23-1, 2-24-1. Dari ke 5 model arsitektur yang digunakan di peroleh 1 model terbaik yaitu 2-24-1 dengan tingkat akurasi 80%, MSE 0.00085177 dan epoch 100. Sehingga model ini bagus untuk memprediksi jumlah penduduk usia produktif pada kota pematangsiantar dimasa yang akan datang.

Kata Kunci: Jumlah Penduduk Usia Produktif; JST; Prediksi; *Backpropagation*; Matlab

Abstract—The productive age itself is a population in the age group between 15 to 64 years, whether they work, go to school, and take care of the household, in this case individuals who are in the scope of productive age are people who can still work well to produce a product and services. This study uses an Artificial Neural Network (ANN) with the *backpropagation* method. The *backpropagation* algorithm is one of the existing methods of neural networks as a prediction, estimation, classification, and pattern recognition algorithm. The research data is secondary data sourced from the Central Statistics Agency (BPS) from 2013 to 2015. The data is divided into 2 parts, namely training and testing data. There are 5 architectural models used in this study. 2-20-1, 2-21-1, 2-22-1, 2-23-1, 2-24-1. Of the 5 architectural models used, the best 1 model is obtained, namely 2-24-1 with an accuracy level of 80%, MSE 0.00085177 and epoch 100. So this model is good for predicting the number of productive age population in the city of Pematangsiantar in the future.

Keywords: Total Population Productive Age; ANN; Prediction; *Backpropagation*; Matlab

1. PENDAHULUAN

Pematang Siantar merupakan salah satu wilayah yang ada di Provinsi Sumatera Utara dengan jumlah penduduk usia produktif yang berdasarkan data yang di peroleh melalui *website* Badan Pusat Statistik (BPS) kota Pematangsiantar pada tahun 2015 mencapai 178.316 jiwa berbanding dengan jumlah penduduk keseluruhan yang berjumlah 247.411 jiwa. Penduduk dengan usia produktif sendiri merupakan penduduk pada kelompok usia antara 15 hingga 64 tahun, baik bekerja, sekolah, dan mengurus rumah tangga. Dalam hal ini individu yang berada pada ruang lingkup usia produktif adalah orang-orang yang masih dapat bekerja dengan baik untuk menghasilkan suatu produk dan jasa. Namun dalam penerapannya, penduduk usia produktif yang seharusnya berada dalam ruang lingkup kerja di kota Pematangsiantar, masih banyak yang belum memenuhi kriteria yang ada berdasarkan data yang ada. Penduduk usia produktif sendiri mempunyai kriteria yaitu, masih sanggup dan energik untuk bekerja, masih bisa berkarya, bekerja keras dan pekerja cerdas, memiliki pandangan untuk kedepan, dan mandiri. Namun, berdasarkan data yang diperoleh terdapat satu variable data yang tidak sesuai atau berbanding terbalik dengan kriteria yang sudah disebutkan, yaitu adanya pengangguran terbuka dalam data yang tercantum penduduk usia produktif di kota Pematangsiantar.

Pengangguran terbuka adalah angkatan kerja yang sama sekali tidak mempunyai pekerjaan. Pengangguran ini terjadi karena angkatan kerja tersebut belum mendapat pekerjaan padahal telah berusaha secara maksimal atau dikarenakan faktor malas mencari pekerjaan atau malas bekerja. Sehingga, pada penelitian ini penulis tertarik untuk memprediksi jumlah penduduk usia produktif yang di dalam nya terdapat beberapa variabel data yaitu, bekerja, pengangguran terbuka, sekolah, mengurus rumah tangga, dan lainnya di luar variable yang di tentukan dengan menerapkan algoritma pembelajaran *backpropagation*, agar nantinya hasil dari penelitian ini dapat di pergunakan oleh pemerintah daerah sebagai acuan atau gambaran untuk mengambil kebijakan dalam membuka dan menyetarakan peluang pekerjaan bagi penduduk usia produktif yang ada di kotamadya Pematangsiantar.

Algoritma pembelajaran *backpropagation* merupakan salah satu metode yang ada pada jaringan syaraf tiruan sebagai algoritma prediksi, estimasi, klasifikasi, dan pengenalan pola. Dalam penelitian ini, penulis menerapkan algoritma *backpropagation* sebagai metode untuk memprediksi jumlah penduduk usia produktif, karena pada penerapannya, *backpropagation* mampu memperkecil tingkat error nya nilai pada suatu data [1], [2].

2. METODOLOGI PENELITIAN

2.1 Algoritma *Backpropagation*

“Algoritma *backpropagation* merupakan salah satu model jaringan syaraf tiruan umpan mundur dengan menggunakan pelatihan terbimbing yang disusun berdasar pada algoritma kesalahan perambatan balik”[3], “sering digunakan dalam menyelesaikan masalah-masalah yang rumit disebabkan karena algoritma ini memiliki dasar matematis yang kuat dan dilatih dengan menggunakan metode belajar terkontrol”[4]. “*Backpropagation* adalah pelatihan jenis terkontrol (*supervised*) yang bermakna pelatihannya menggunakan pola penyesuaian bobot untuk mencapai nilai kesalahan yang minimum antara keluaran hasil prediksi dengan keluaran yang nyata [5]”.

2.2 Software Matlab

“*Matlab* merupakan *software* tepat sebagai alat komputasi yang melibatkan penggunaan matriks dan vektor, juga mendukung berbagai algoritma untuk jaringan syaraf tiruan diantaranya adalah algoritma *backpropagation* dengan menyediakan fungsi-fungsi khusus untuk menyelesaikan model jaringan syaraf tiruan”[4].

3. HASIL DAN PEMBAHASAN

Pada implementasi algoritma *backpropagation* untuk melakukan proses prediksi jumlah penduduk usia produktif pada kota Pematangsiantar, diperlukan data sebagai berikut:

1) Data Input

Pada penelitian ini, penulis menggunakan data input jumlah penduduk usia produktif yang diperoleh melalui website Badan Pusat Statistik (BPS) Pematangsiantar. Data input yang digunakan berupa data matriks tahun yang dimulai dari 2013 hingga 2015. Data input jumlah penduduk usia produktif yang digunakan dapat dilihat pada tabel 1. berikut.

Tabel 1. Data Jumlah Penduduk Usia Produktif

No	Uraian	2013	2014	2015
1	Bekerja	100.958	91.802	110.785
2	Pengangguran Terbuka	7.145	9.373	11.593
3	Sekolah	26.321	30.762	22.008
4	Mengurus Rumah Tangga	27.233	32.007	24.580
5	Lainnya	7.248	12.106	9.350

(Sumber : Website Badan Pusat Statistik Pematangsiantar)

2) Transformasi Data

Transformasi data dilakukan dengan melakukan normalisasi data yang bertujuan agar nantinya nilai data awal dapat sesuai dengan apa yang dibutuhkan JST. Sebelum melakukan normalisasi, data yang ada akan dikelompokkan menjadi 2 jenis data, yaitu sebagai berikut.

a) Data Pelatihan (*Training Data*)

Proses ini dilakukan untuk melakukan *training data* terhadap data latih. Pada penelitian ini penulis melakukan 3 pembagian data input terhadap 5 data awal yang digunakan, dengan penerapan sebagai berikut.

- a) Data awal penduduk usia produktif : 2013-2015
- b) Data Pelatihan (*Training Data*) : 2013-2014
- c) Target Data : 2014

Data pelatihan (*Training Data*) yang sudah dikelompokkan dapat dilihat pada tabel 2. berikut.

Tabel 2. Data Pelatihan (*Training Data*)

Uraian	2013	2014	Target
Bekerja	100.958	91.802	91.802
Pengangguran Terbuka	7.145	9.373	9.373
Sekolah	26.321	30.762	30.762
Mengurus Rumah Tangga	27.233	32.007	32.007
Lainnya	7.248	12.106	12.106

Data sudah dikelompokkan menjadi data pelatihan (*training data*), untuk mendapatkan hasil sesuai dengan fungsi aktivasi yang digunakan yaitu sigmoid biner, dimana pola input data diskalakan pada rentang 0-1 dilakukan konversi data melalui normalisasi data dengan persamaan yang sudah ditentukan. Normalisasi data yang sudah dikonversi dapat dilihat pada tabel 3 berikut.

Tabel 3. Normalisasi Training Data

Uraian	2013	2014	Target
Bekerja	0,9000	0,8219	0,8219
Pengangguran Terbuka	0,1000	0,1190	0,1190
Sekolah	0,2635	0,3014	0,3014
Mengurus Rumah Tangga	0,2713	0,3120	0,3120
Lainnya	0,1009	0,1423	0,1423

b) Data Pengujian (*Testing Data*)

Data pengujian dilakukan untuk melakukan uji coba sistem terhadap data uji. Sesuai pada pembagian data pelatihan, tidak jauh berbeda pada data pengujian dengan pembagian 3 data input pada 5 data awal yang digunakan, dengan penerapan sebagai berikut.

- Data awal penduduk usia produktif : 2013-2015
- Data input pengujian (*Testing Data*) : 2014-2015
- Target data : 2015

Data pengujian (*Testing Data*) yang sudah dikelompokkan dapat dilihat pada tabel 4. berikut.

Tabel 4. Data Pengujian (Testing Data)

Uraian	2014	2015	Target
Bekerja	91.802	110.785	110.785
Pengangguran Terbuka	9.373	11.593	11.593
Sekolah	30.762	22.008	22.008
Mengurus Rumah Tangga	32.007	24.580	24.580
Lainnya	12.106	9.350	9.350

Seperti halnya pada data pelatihan (*training data*), setelah data dikelompokkan menjadi data pengujian (*testing data*) dilakukan konversi data melalui normalisasi data sesuai dengan persamaan yang sudah ditentukan. Normalisasi data yang sudah dikonversi dapat dilihat pada tabel 5. berikut.

Tabel 5. Normalisasi Testing Data

Uraian	2014	2015	Target
Bekerja	0,7503	0,9000	0,9000
Pengangguran Terbuka	0,1002	0,1177	0,1177
Sekolah	0,2689	0,1998	0,1998
Mengurus Rumah Tangga	0,2787	0,2201	0,2201
Lainnya	0,1217	0,1000	0,1000

3.2 Pemilihan Arsitektur Terbaik

Data *input* yang sudah dinormalisasi akan diolah dengan menggunakan *software Matlab R2011b* dengan menyisipkan lapisan tersembunyi (*hidden layer*). Pada penelitian ini terdapat lima model arsitektur yaitu 2-20-1, 2-21-1, 2-22-1, 2-23-1, 2-24-1, dan 2-25-1. Dari kelima model tersebut dapat diperoleh satu model arsitektur terbaik dibandingkan dengan model arsitektur lainnya, yaitu arsitektur 2-24-1. Model arsitektur dapat dilihat pada tabel 6 di bawah ini

Tabel 6. Arsitektur *Backpropagation*

No.	Arsitektur	Training			Testing	
		Epoch	Waktu	MSE	MSE	Akurasi
1	2 - 20 - 1	3109	0:56	0.00063869	0.37254282	100
2	2 - 21 - 1	3795	1:01	0.00099198	0.87153455	60
3	2 - 22 - 1	583	0:14	0.00100378	0.28598602	20
4	2 - 23 - 1	91	0:01	0.00100852	0.55804045	40
5	2 - 24 - 1	3221	0:24	0.00085177	0.17159564	80

Berdasarkan gambar diatas, dapat dilihat bahwa penulis memulai penelitian ini dengan menentukan masalah yang akan diteliti dan tujuan penelitian ini dibuat. Setelah masalah ditentukan, maka penulis mengambil data dari Badan Pusat Statistik Pematangiantar dan melakukan validasi data. Apabila data sudah valid maka data akan diproses dengan melakukan transformasi data melalui normalisasi data pelatihan dan data pengujian pada *software Microsoft Excel*. Setelah itu menentukan pola, dengan pola yang sudah diperoleh.

3.3 Pelatihan dan pengujian data dengan matlab.

Dalam melakukan pelatihan dan pengujian hasil dari pengolahan data untuk memprediksi jumlah pengangguran dikota Pematangsiantar, maka pengolahan data tersebut juga akan diujikan ke dalam sistem komputerisasi. Dalam melakukan pengujian data, penulis menggunakan *Software Matlab R2011*.

Langkah pertama yang harus dilakukan dalam menerapkan algoritma *Backpropagation* adalah membagi data yang akan diuji menjadi dua bagian, dimana bagian pertama adalah untuk data pelatihan dan bagaian kedua adalah untuk data pengujian. Data pelatihan jumlah pengangguran dikota Pematangsiantar dapat di lihat pada tabel 7. berikut.

Tabel 7. Data Training Sebelum Dinormalisasikan(2013-2014) / Target 2014

Uraian	2013	2014	Target
Bekerja	100,958	91,802	91,802
Pengangguran Terbuka	7,145	9,373	9,373
Sekolah	26,321	30,762	30,762
Mengurus Rumah Tangga	27,233	32,007	32,007
Lainnya	7,248	12,106	12,106

Data *Training* yang telah dinormalisasi dapat dilihat pada table 8 berikut :

Tabel 8. Data Training Sesudah Dinormalisasikan(2013-2014) / Target 2014

Uraian	2013	2014	Target
Bekerja	0.9000	0.8219	0.8219
Pengangguran Terbuka	0.1000	0.1190	0.1190
Sekolah	0.2635	0.3014	0.3014
Mengurus Rumah Tangga	0.2713	0.3120	0.3120
Lainnya	0.1009	0.1423	0.1423

Data pengujian (*Testing*) sebelum dinormalisasi dapat dilihat pada tabel 9.

Tabel 9. Data Training Sebelum Dinormalisasikan(2014-2015) / Target 2016

Uraian	2014	2015	Target
Bekerja	91,802	110,785	110,785
Pengangguran Terbuka	9,373	11,593	11,593
Sekolah	30,762	22,008	22,008
Mengurus Rumah Tangga	32,007	24,580	24,580
Lainnya	12,106	9,350	9,350

Data *Testing* yang telah dinormalisasi dapat dilihat pada tabel 10

Tabel 10. Data Training Sesudah Dinormalisasikan(2014-2015) / Target 2016

Uraian	2014	2015	Target	Uraian	2014
Bekerja	0.7503	0.9000	0.9000	Bekerja	0.7503
Pengangguran Terbuka	0.1002	0.1177	0.1177	Pengangguran Terbuka	0.1002
Sekolah	0.2689	0.1998	0.1998	Sekolah	0.2689
Mengurus Rumah Tangga	0.2787	0.2201	0.2201	Mengurus Rumah Tangga	0.2787
Lainnya	0.1217	0.1000	0.1000	Lainnya	0.1217

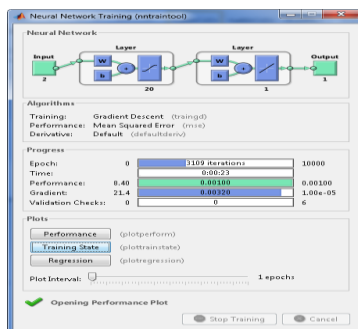
3.4 Pengujian Data

Pada penelitian ini menggunakan 5 (lima) model arsitektur pelatihan dan pengujian data yakni 2-20-1, 2-21-1, 2-22-1, 2-23-1, 2-24-1. Pelatihan pengujian ke 5 (lima) arsitektur dapat dilihat sebagai berikut :

a) Pengujian Data dengan Model Arsitektur 2-20-1

Pada tahap ini pelatihan dan pengujian data dengan algoritma *Backpropagation* model arsitektur 2-20-1. Maksudnya pelatihan dan pengujian data menggunakan *layer* masukan sebanyak 2 *neuron*, 20 *layer* tersembunyi dan 1 *neuron* keluaran. Gambar 1 dapat dijelaskan bahwa pelatihan dan pengujian dengan model arsitektur 2-20-1 menghasilkan *epoch* sebesar 3109 dalam waktu 23 detik. Hasil akurasi dari pengujian menggunakan model arsitektur 2-20-1 dapat dilihat pada tabel 4.5 terdapat 5 (lima) jenis kegiatan yang dipakai dalam pengujian. Nilai target diperoleh dari tabel data normalisasi tahun 2015, nilai *output* diperoleh dari hasil pelatihan menggunakan *Software Matlab R2011b* dengan rumus $[a, Pf, Af, e, Perf] = sim(net, PP, [], [], TT)$, nilai *error* diperoleh dari $error^2$ (^ = pangkat), jumlah SSE

adalah total keseluruhan nilai SSE. Nilai MSE diperoleh dari $SSE/5$ (nilai SSE dibagi nilai data) dan hasil 1 (benar) diperoleh dari rumus $=if(error \leq 0,01; 1; 0)$ untuk data pengujian. Nilai akurasi (%) diperoleh dari jumlah benar / $5 * 100$ ($5 =$ jumlah data).



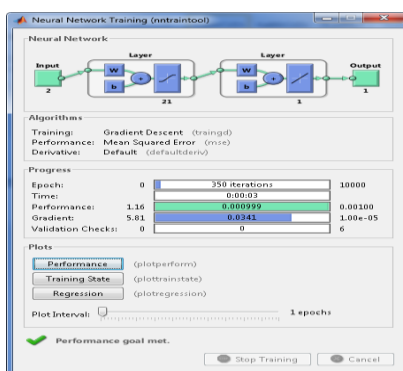
Gambar 1. Pelatihan menggunakan Model Arsitektur 2-20-1

Tabel 11. Hasil Akurasi Data Testing Dengan Model Arsitektur 2-20-1

No	Pola	Target	Output	Error	SSE	Hasil
1	Pola 1	0.90000	1.82170	-0.92170	0.84953089	1
2	Pola 2	0.11769	0.17170	-0.05401	0.00291706	1
3	Pola 3	0.19983	0.76610	-0.56627	0.32066011	1
4	Pola 4	0.22012	0.78320	-0.56308	0.31706322	1
5	Pola 5	0.10000	0.27480	-0.17480	0.03055504	1
					1.49017128	100
					MSE	0.37254282

b) Pengujian Data dengan Model Arsitektur 2-21-1

Pada tahap ini pelatihan dan pengujian data dengan algoritma *Backpropagation* model arsitektur 2-21-1. Maksudnya pelatihan dan pengujian data menggunakan *layer* masukan sebanyak 2 *neuron*, 21 *layer* tersembunyi dan 1 *neuron* keluaran. Dapat dilihat pada gambar berikut:



Gambar 2. Pelatihan menggunakan Model Arsitektur 2-21-1

Gambar 2 dapat dijelaskan bahwa pelatihan dan pengujian dengan model arsitektur 2-21-1 menghasilkan *epoch* sebesar 350 dalam waktu 3 detik. Hasil akurasi dari pengujian menggunakan model arsitektur 2-21-1 dapat dilihat pada tabel 4.6 terdapat 5 (lima) jenis kegiatan yang dipakai dalam pengujian. Nilai target diperoleh dari tabel data normalisasi tahun 2015, nilai *output* diperoleh dari hasil pelatihan menggunakan *Software Matlab R2011b* dengan rumus $[a, Pf, Af, e, Perf] = sim(net, PP, [], [], TT)$, nilai *error* diperoleh dari $error^2$ ($\wedge =$ pangkat), jumlah SSE adalah total keseluruhan nilai SSE. Nilai MSE diperoleh dari $SSE/5$ (nilai SSE dibagi nilai data) dan hasil 1 (benar) diperoleh dari rumus $=if(error \leq 0,01; 1; 0)$ untuk data pengujian. Nilai akurasi (%) diperoleh dari jumlah benar / $5 * 100$ ($5 =$ jumlah data).

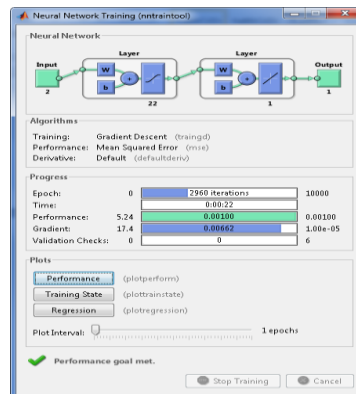
Tabel 12. Hasil Akurasi Data Testing dengan Model Arsitektur 2-21-1

No	Pola	Target	Output	Error	SSE	Hasil
1	Pola 1	0.90000	2.69980	-1.79980	3.23928004	1
2	Pola 2	0.11769	0.15550	-0.03781	0.00142959	1
3	Pola 3	0.19983	-0.14570	0.34553	0.11939196	0
4	Pola 4	0.22012	-0.13490	0.35502	0.12603660	0

5	Pola 5	0.10000	0.14920	-0.04920	0.00242064	1
					3.48613818	60
				MSE	0.87153455	

c) Pengujian Data dengan Model Arsitektur 2-22-1

Pada tahap ini pelatihan dan pengujian data dengan algoritma *Backpropagation* menggunakan model arsitektur 2-22-1. Maksudnya pelatihan dan pengujian data menggunakan *layer* masukan sebanyak 2 *neuron*, 22 *layer* tersembunyi dan 1 *neuron* keluaran. Pengujian jaringan syaraf tiruan dengan model arsitektur 2-22-1 dapat dilihat pada gambar 3. dibawah ini.



Gambar 3. Pelatihan Menggunakan Model Arsitektur 2-22-1

Gambar 3. dapat dijelaskan bahwa pelatihan dan pengujian dengan model arsitektur 2-22-1 menghasilkan *epoch* sebesar 2960 dalam waktu 22 detik. Hasil akurasi dari pengujian menggunakan model arsitektur 2-22-1 dapat dilihat pada tabel 4.7. terdapat 5 (lima) jenis kegiatan yang dipakai dalam pengujian. Nilai target diperoleh dari tabel data normalisasi tahun 2015, nilai *output* diperoleh dari hasil pelatihan menggunakan *Software* Matlab R2011b dengan rumus $[a, Pf, Af, e, Perf] = sim(net, PP, [], [], TT)$, nilai *error* diperoleh dari : $target - output$, nilai SSE diperoleh dari : $error^2$ (^ = pangkat), jumlah SSE adalah total keseluruhan nilai SSE. Nilai MSE diperoleh dari : $jumlah\ SSE / 5$ (banyak jenis kegiatan) dan hasil 1 (benar) diperoleh dari rumus $= IF(error \leq 0.01; 1; 0)$ untuk data pengujian. Nilai akurasi (%) diperoleh dari : $jumlah\ benar / 5 * 100$.

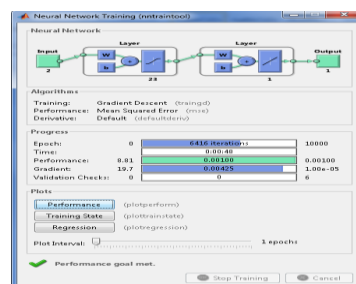
Tabel 13. Hasil Akurasi Data Testing Dengan Model Arsitektur 2-22-1

No	Pola	Target	Output	Error	SSE	Hasil
1	Pola 1	0.90000	-0.03500	0.93500	0.87422500	0
2	Pola 2	0.11769	0.14660	-0.02891	0.00083578	1
3	Pola 3	0.19983	-0.26830	0.46813	0.21914703	0
4	Pola 4	0.22012	-0.00290	0.22302	0.04973628	0
5	Pola 5	0.10000	-0.04710	0.14710	0.02163841	0
					1.14394409	20
				MSE	0.28598602	

d) Pengujian Data dengan Model Arsitektur 2-23-1

Pada tahap ini pelatihan dan pengujian data dengan algoritma *Backpropagation* menggunakan model arsitektur 2-23-1. Maksudnya pelatihan dan pengujian data menggunakan *layer* masukan sebanyak 2 *neuron*, 23 *layer* tersembunyi dan 1 *neuron* keluaran.

Pengujian jaringan syaraf tiruan dengan model arsitektur 2-23-1 dapat dilihat pada gambar 4. dibawah ini.



Gambar 4. Pelatihan Menggunakan Model Arsitektur 2-23-1

Gambar 4. dapat dijelaskan bahwa pelatihan dan pengujian dengan model arsitektur 2-23-1 menghasilkan *epoch* sebesar 6416 dalam waktu 48 detik.

Hasil akurasi dari pengujian menggunakan model arsitektur 2-23-1 dapat dilihat pada tabel 4. terdapat 5 (lima) jenis kegiatan yang dipakai dalam pengujian. Nilai target diperoleh dari tabel data normalisasi tahun 2015, nilai *output* diperoleh dari hasil pelatihan menggunakan *Software* Matlab R2011b dengan rumus $[a,Pf,Af,e,Perf]=sim(net,PP,[],[],TT)$, nilai *error* diperoleh dari : target – *output*, nilai SSE diperoleh dari : $error^2$ (^ = pangkat), jumlah SSE adalah total keseluruhan nilai SSE. Nilai MSE diperoleh dari : jumlah SSE/5 (banyak jenis kegiatan) dan hasil 1 (benar) diperoleh dari rumus $=IF(error \leq 0.01; 1; 0)$ untuk data pengujian. Nilai akurasi (%) diperoleh dari : jumlah benar/5*100.

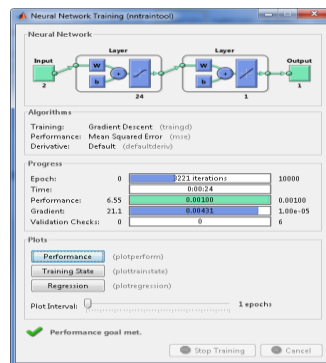
Tabel 14. Hasil Akurasi Data *Testing* dengan Model Arsitektur 2-23-1

No	Pola	Target	Output	Error	SSE	Hasil
1	Pola 1	0.90000	2.38890	-1.48890	2.21682321	1
2	Pola 2	0.11769	0.07850	0.03919	0.00153587	0
3	Pola 3	0.19983	0.11510	0.08473	0.00717941	0
4	Pola 4	0.22012	0.30150	-0.08138	0.00662330	1
5	Pola 5	0.10000	-0.20840	0.30840	0.09511056	0
					2.23216179	40
MSE					0.55804045	

e) Pengujian Data dengan Model Arsitektur 2-24-1

Pada tahap ini pelatihan dan pengujian data dengan algoritma *Backpropagation* menggunakan model arsitektur 2-24-1. Maksudnya pelatihan dan pengujian data menggunakan *layer* masukan sebanyak 2 *neuron*, 24 *layer* tersembunyi dan 1 *neuron* keluaran.

Pengujian jaringan syaraf tiruan dengan model arsitektur 2-24-1 dapat dilihat pada gambar 5. dibawah ini.



Gambar 5. Pelatihan Menggunakan Model Arsitektur 2-24-1

Gambar 5. dapat dijelaskan bahwa pelatihan dan pengujian dengan model arsitektur 2-24-1 menghasilkan *epoch* sebesar 3221 dalam waktu 24 detik.

Hasil akurasi dari pengujian menggunakan model arsitektur 2-24-1 dapat dilihat pada tabel 4.9. terdapat 5 (lima) jenis kegiatan yang dipakai dalam pengujian. Nilai target diperoleh dari tabel data normalisasi tahun 2015, nilai *output* diperoleh dari hasil pelatihan menggunakan *Software* Matlab R2011b dengan rumus $[a,Pf,Af,e,Perf]=sim(net,PP,[],[],TT)$, nilai *error* diperoleh dari : target – *output*, nilai SSE diperoleh dari : $error^2$ (^ = pangkat), jumlah SSE adalah total keseluruhan nilai SSE. Nilai MSE diperoleh dari : jumlah SSE/5 (banyak jenis kegiatan) dan hasil 1 (benar) diperoleh dari rumus $=IF(error \leq 0.01; 1; 0)$ untuk data pengujian. Nilai akurasi (%) diperoleh dari : jumlah benar/5*100.

Tabel 15. Hasil Akurasi Data *Testing* dengan Model Arsitektur 3-45-1

No	Pola	Target	Output	Error	SSE	Hasil
1	Pola 1	0.90000	1.26690	-0.36690	0.13461561	1
2	Pola 2	0.11769	0.07420	0.04349	0.00189139	0
3	Pola 3	0.19983	0.74070	-0.54087	0.29253882	1
4	Pola 4	0.22012	0.72740	-0.50728	0.25733672	1
5	Pola 5	0.10000	0.11700	-0.01700	0.00028900	1
					0.68638255	80
MSE					0.17159564	

- f) Kesimpulan Pelatihan dan Pengujian Jaringan Saraf Tiruan
Kesimpulan dari pelatihan dan pegujian data dapat dilihat pada tabel 16. yang menggunakan 5 model arsitektur.

Tabel 16. Kesimpulan Pelatihan dan Pengujian Jaringan

NO	Arsitektur	Algoritma Backpropagation				
		Training			Testing	
		Epoch	Waktu	MSE	MSE	Akurasi
1	2 - 20 - 1	3109	0:56	0.00063869	0.37254282	100
2	2 - 21 - 1	3795	1:01	0.00099198	0.87153455	60
3	2 - 22 - 1	583	0:14	0.00100378	0.28598602	20
4	2 - 23 - 1	91	0:01	0.00100852	0.55804045	40
5	2 - 24 - 1	100	0:04	0.00085177	0.17159564	80

Berdasarkan tabel 16. dapat dijelaskan bahwa model arsitektur pelatihan 2-24-1 menggunakan *layer* masukan sebanyak 2 *neuron*, *layer* tersembunyi sebanyak 24 *neuron* dan *layer* keluaran sebanyak 1 *neuron* dengan *epoch* sebesar 100 iterasi, waktu 00:04, MSE pelatihan sebesar 0.00085177 dan MSE pengujian sebesar 0.17159564 dengan tingkat akurasi sebesar 80%.

- g) Hasil Estimasi Menggunakan Algoritma *Backpropagation*
Dalam melakukan proses estimasi untuk beberapa tahun berikutnya, penelitian ini menggunakan data pengujian dengan model arsitektur terbaik yakni 2-24-1.
- h) Hasil Prediksi Tahun 2016 Menggunakan Model Arsitektur 2-24-1
Untuk mendapatkan hasil prediksi, penulis menggunakan *Software* Matlab R2011b dengan data yang diolah merupakan data pengujian (*testing*) dan menggunakan model arsitektur terbaik yakni 2-24-1. Hasil prediksi tahun 2016 dapat dilihat pada tabel 17.

Tabel 17. Prediksi Tahun 2016

No	Data Real	Target	Target Prediksi	Prediksi
1	110,785	0.9000	0.9000	110785
2	11,593	0.1177	0.0742	6079
3	22,008	0.1998	0.2199	24553
4	24,580	0.2201	0.1966	21598
5	9,350	0.1000	0.1463	15221

Berdasarkan tabel 17. dapat dijelaskan bahwa data real diperoleh dari tahun 2015, data target diperoleh dari data testing tahun 2015 yang telah dinormalisasi, target prediksi diperoleh menggunakan *Software* Matlab R2011b dengan *Source Code* sebagai berikut :

```
>> net=newff(minmax(PP),[45,1],{'tansig','tansig'},'traingd');
>> net.LW{1,1};
>> net.b{1};
>> net.LW{2,1};
>> net.b{2};
>> net.trainParam.epochs=100000;
>> net.trainParam.goal = 0.001;
>> net.trainParam.Lr = 0.01;
>> net.trainParam.show = 1000;
>> net=train(net,PP,TT)
[a,Pf,Af,e,Perf]=sim(net,PP,[],[],TT)
```

- i) Hasil Prediksi Tahun 2017 Menggunakan Model Arsitektur 2-24-1
Untuk mendapatkan hasil prediksi, penulis menggunakan *Software* Matlab R2011b dengan data yang diolah merupakan data pengujian (*testing*) dan menggunakan model arsitektur terbaik yakni 2-24-1. Hasil prediksi tahun 2017 dapat dilihat pada tabel 18.

Tabel 18. Prediksi Tahun 2017

No	Data Real	Target	Target Prediksi	Prediksi
1	110785	0.88000	0.88000	108167
2	6079	0.09769	0.05420	98712
3	24553	0.17983	0.20010	13101
4	21598	0.20012	0.17650	10013
5	15221	0.08000	0.12620	3429

Berdasarkan tabel 18. dapat dijelaskan bahwa data real diperoleh dari tahun 2016, data target diperoleh dari data testing tahun 2016 yang telah dinormalisasi, target prediksi diperoleh menggunakan *Software* Matlab R2011b dengan *Source Code* sebagai berikut :

```
>> net=newff(minmax(PP),[45,1],{'tansig','tansig'},'traingd');
>> net.LW{1,1};
>> net.b{1};
>> net.LW{2,1};
>> net.b{2};
>> net.trainParam.epochs=100000;
>> net.trainParam.goal = 0.001;
>> net.trainParam.Lr = 0.01;
>> net.trainParam.show = 1000;
>> net=train(net,PP,TT) [a,Pf,Af,e,Perf]=sim(net,PP,[],[],TT)
```

j) Hasil Prediksi Tahun 2018 Menggunakan Model Arsitektur 2-24-1

Untuk mendapatkan hasil prediksi, penulis menggunakan *Software* Matlab R2011b dengan data yang diolah merupakan data pengujian (*testing*) dan menggunakan model arsitektur terbaik yakni 2-24-1. Hasil prediksi tahun 2018 dapat dilihat pada 19.

Tabel 19. Prediksi Tahun 2018

No	Data Real	Target	Target Prediksi	Prediksi
1	110903	0.9000	0.89960	110852
2	12754	0.1509	0.13330	13101
3	23879	0.2358	0.18590	19811
4	21863	0.2205	0.25360	28447
5	8853	0.1212	0.15440	15793

Berdasarkan tabel 19. dapat dijelaskan bahwa data real diperoleh dari tahun 2017, data target diperoleh dari data testing tahun 2017 yang telah dinormalisasi, target prediksi diperoleh menggunakan *Software* Matlab R2011b dengan *Source Code* sebagai berikut :

```
>> net=newff(minmax(PP),[45,1],{'tansig','tansig'},'traingd');
>> net.LW{1,1};
>> net.b{1};
>> net.LW{2,1};
>> net.b{2};
>> net.trainParam.epochs=100000;
>> net.trainParam.goal = 0.001;
>> net.trainParam.Lr = 0.01;
>> net.trainParam.show = 1000;
>> net=train(net,PP,TT)
[a,Pf,Af,e,Perf]=sim(net,PP,[],[],TT)
```

k) Hasil Prediksi Tahun 2019 Menggunakan Model Arsitektur 2-24-1

Untuk mendapatkan hasil prediksi, penulis menggunakan *Software* Matlab R2011b dengan data yang diolah merupakan data pengujian (*testing*) dan menggunakan model arsitektur terbaik yakni 2-24-1. Hasil prediksi tahun 2019 dapat dilihat pada tabel 20.

Tabel 20. Prediksi Tahun 2019

No	Data Real	Target	Target Prediksi	Prediksi
1	110852	0.8996	0.89790	110595
2	13101	0.1333	0.12750	16461
3	19811	0.1859	0.24050	30269
4	28447	0.2536	0.24980	31405
5	15793	0.1544	0.1101	14335

Berdasarkan tabel 20. dapat dijelaskan bahwa data real diperoleh dari tahun 2019, data target diperoleh dari data testing tahun 2019 yang telah dinormalisasi, target prediksi diperoleh menggunakan *Software* Matlab R2011b dengan *Source Code* sebagai berikut :

```
>> net=newff(minmax(PP),[45,1],{'tansig','tansig'},'traingd');
>> net.LW{1,1};
>> net.b{1};
>> net.LW{2,1};
```

```
>> net.b{2};
>> net.trainParam.epochs=100000;
>> net.trainParam.goal = 0.001;
>> net.trainParam.Lr = 0.01;
>> net.trainParam.show = 1000;
>> net=train(net,PP,TT)
[a,Pf,Af,e,Perf]=sim(net,PP,[],[],TT)
```

1) Hasil Prediksi Tahun 2020 Menggunakan Model Arsitektur 2-24-1

Untuk mendapatkan hasil prediksi, penulis menggunakan *Software* Matlab R2011b dengan data yang diolah merupakan data pengujian (*testing*) dan menggunakan model arsitektur terbaik yakni 2-24-1. Hasil prediksi tahun 2020 dapat dilihat pada tabel 21.

Tabel 21. Prediksi Tahun 2020

No	Data Real	Target	Target Prediksi	Prediksi
1	110595	0.8979	0.9055	111257
2	16461	0.12750	0.1531	20725
3	30269	0.2405	0.1890	25044
4	31405	0.2498	0.2387	31024
5	14335	0.1101	0.1486	20183

Berdasarkan tabel 21. dapat dijelaskan bahwa data real diperoleh dari tahun 2019, data target diperoleh dari data testing tahun 2019 yang telah dinormalisasi, target prediksi diperoleh menggunakan *Software* Matlab R2011b dengan *Source Code* sebagai berikut :

```
>> net=newff(minmax(PP),[45,1],{'tansig','tansig','traingd'});
>> net.LW{1,1};
>> net.b{1};
>> net.LW{2,1};
>> net.b{2};
>> net.trainParam.epochs=100000;
>> net.trainParam.goal = 0.001;
>> net.trainParam.Lr = 0.01;
>> net.trainParam.show = 1000;
>> net=train(net,PP,TT)
[a,Pf,Af,e,Perf]=sim(net,PP,[],[],TT)
```

3.5 Pembahasan

Berdasarkan arsitektur jaringan syaraf tiruan tersebut didapatkan hasil prediksi jumlah penduduk usia produktif hingga 2020 di Pematangsiantar mengalami kenaikan yang cukup stabil dibandingkan hasil prediksi jumlah penduduk usia produktif tahun sebelumnya. Sebagai perbandingan hasil prediksi jumlah penduduk usia produktif, maka dapat dilihat pada tabel 22. untuk data jumlah penduduk usia produktif 3 tahun kebelakang yaitu tahun 2013 hingga 2015 dan data prediksi jumlah penduduk usia produktif 5 tahun kedepan yaitu tahun 2015 hingga 2020.

Tabel 22. Perbandingan Data Jumlah Penduduk usia produktif

Uraian	2013	2014	2015	2016	2017	2018	2019	2020
Bekerja	100,958	91,802	110,785	110785	110903	110852	110595	111257
Pengangguran Terbuka	7,145	9,373	11,593	6079	12754	13101	16461	20725
Sekolah	26,321	30,762	22,008	24553	23879	19811	30269	25044
Mengurus Rumah Tangga	27,233	32,007	24,580	21598	21863	28447	31405	31024
Lainnya	7,248	12,106	9,350	15221	8853	15793	14335	20183

Berdasarkan tabel 22. Jumlah penduduk usia produktif mengalami peningkatan setiap tahunnya. Namun dalam hal ini, hasil prediksi yang diperoleh menggunakan metode *Backpropagation* dengan model arsitektur terbaik yakni 2-24-1 dengan tingkat akurasi sebesar 80% merupakan suatu hal yang masih merupakan prediksi yang dapat digunakan sebagai acuan atau tidak. Dengan hasil yang diperoleh dapat digunakan sebagai acuan di Pematangsiantar.

4. KESIMPULAN

Hasil prediksi jumlah penduduk usia produktif tahun 2015 hingga tahun 2020 mengalami kenaikan dan penurunan dibanding jumlah penduduk usia produktif 3 tahun sebelumnya yakni tahun 2013 hingga tahun 2015. Setelah dilakukan pelatihan dan pengujian menggunakan 5 model arsitektur yakni 2-20-1, 2-21-1, 2-22-1, 2-23-1, dan 2-24-1, maka didapat 1 arsitektur terbaik dengan model 2-24-1 yang memiliki tingkat akurasi tertinggi yaitu 80% dan dengan learning rate yaitu 0.01, jumlah epoch sebanyak 3221 iterasi dengan waktu 24 detik dan nilai *Mean Squared Error* (MSE) terendah dibanding dengan model arsitektur lain sebesar 0.17159564.

REFERENCES

- [1] D. Syahfitri, A. P. Windarto, M. Fauzan, and Solikhun, "Peningkatan Nilai Akurasi Prediksi Algoritma Backpropagation (Kasus : Jumlah Pengunjung Tamu pada Hotel berbintang di Sumatera Utara)," *J. Inf. Syst. Res.*, vol. 2, no. 1, pp. 90–101, 2020.
- [2] I. C. Saragih, D. Hartama, and A. Wanto, "Prediksi Perkembangan Jumlah Pelanggan Listrik Menurut Pelanggan Area Menggunakan Algoritma Backpropagation," *Build. Informatics, Technol. science*, vol. 2, no. 1, pp. 48–54, 2020.
- [3] T. Ujjianto, B. Winardi, and Karnoto, "Proyeksi Kebutuhan Energi Listrik APJ Pekalongan Tahun 2014-2018 Menggunakan Jaringan Saraf Tiruan Metode Backpropagation Dengan Software MATLAB R2014A," *TRANSIENT*, p. 638, 2015.
- [4] R. L. Gema and D. Kartika, "Algoritma Propagasi Balik Dalam Pencarian Pola Training Terbaik Untuk Menentukan Prediksi Produksi Usaha Songket Silungkang Dengan Menggunakan MATLAB," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komput.*, p. 59, 2018.
- [5] Y. D. Lestari, "Jaringan Syaraf Tiruan Untuk Prediksi Penjualan jamur Menggunakan Algoritma Backpropagation," *J. ISD*, p. 41, 2017.