

**PENDEKATAN *NEURAL NETWORK* DALAM PERAMALAN
JUMLAH PENDUDUK KOTA SEMARANG DENGAN
MENGUNAKAN METODE *BACKPROPAGATION***

SKRIPSI

Diajukan untuk Memenuhi Sebagai Syarat Guna

Memperoleh Gelar Sarjana Strata Satu (S1)

dalam Ilmu Matematika



Oleh: **AGUSTIN ABSARI WAHYU KUNTARINI**

NIM: 1808046013

**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG**

2023

PERNYATAAN KEASLIAN

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Agustin Absari Wahyu Kuntarini

NIM : 1808046013

Program Studi : Matematika

Menyatakan bahwa skripsi yang berjudul:

Pendekatan *Neural Network* dalam Peramalan Jumlah Penduduk Kota Semarang dengan Menggunakan Metode *Backpropagation*

Secara keseluruhan adalah hasil penelitian/karya saya sendiri, kecuali bagian tertentu yang dirujuk sumbernya.

Semarang, 14 Maret 2023

Pembuat



Agustin Absari Wahyu Kuntarini

NIM. 1808046013

PENGESAHAN



KEMENTERIAN AGAMA REPUBLIK INDONESIA
UNIVERSITAS ISLAM NEGERI WALISONGO
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Prof. Dr. Hamka Ngaliyan Semarang 50185
Telepon. 024-7601295, Fax. 024-7615387, www.walisongo.ac.id

PENGESAHAN

Naskah skripsi berikut ini:

Judul : Pendekatan *Neural Network* dalam Peramalan Jumlah Penduduk Kota Semarang dengan Menggunakan Metode *Backpropagation*

Penulis : Agustin Absari Wahyu Kuntarini

NIM : 1808046013

Jurusan : Matematika

Telah diujikan dalam sidang *munaqosyah* oleh Dewan Penguji Fakultas Sains dan Teknologi UIN Walisongo Semarang dan dapat diterima sebagai salah satu syarat memperoleh gelar sarjana dalam Ilmu Matematika.

Semarang, 14 Maret 2023

DEWAN PENGUJI

Ketua Sidang

Ariska Kurnia Rachmawati, M.Sc.

NIP.198908112019032019

Penguji Utama I

Budi Cahyono, S.Pd., M.Si

NIP.198012152009121805

Dosen Pembimbing I

Eya Khoirun Nisa, M.Si.

NIP.198701022019032010

Sekretaris Sidang

Siti Maslihah, M.Si.

NIP.197706112011012004

Penguji Utama II

Nur Khasanah, M.Si.

NIP.199111212019032017

Dosen Pembimbing II

Ariska Kurnia Rachmawati, M.Sc.

NIP.198908112019032019



NOTA DINAS

Semarang, 14 Maret 2023

Yth. Ketua Program Studi Matematika

Fakultas Sains dan Teknologi

UIN Walisongo Semarang

Assalamu'alaikum Warahmatullahi Wabarakatuh

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan, dan koreksi naskah skripsi dengan:

Judul : Pendekatan *Neural Network* dalam Peramalan Jumlah Penduduk Kota Semarang dengan Menggunakan Metode *Backpropagation*

Nama : Agustin Absari Wahyu Kuntarini

NIM : 1808046013

Program Studi: Matematika

Saya menandakan bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo Semarang untuk diajukan dalam Sidang Munaqosyah.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Pembimbing 1



Eva Khoirun Nisa, M.Si

NIP. 19870102 201903 2 010

NOTA DINAS

Semarang, 14 Maret 2023

Yth. Ketua Program Studi Matematika

Fakultas Sains dan Teknologi

UIN Walisongo Semarang

Assalamu'alaikum Warahmatullahi Wabarakatuh

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan, dan koreksi naskah skripsi dengan:

Judul : Pendekatan *Neural Network* dalam Peramalan Jumlah Penduduk Kota Semarang dengan Menggunakan Metode *Backpropagation*

Nama : Agustin Absari Wahyu Kuntarini

NIM : 1808046013

Program Studi: Matematika

Saya menandakan bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo Semarang untuk diajukan dalam Sidang Munaqosyah.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Pembimbing II,



Ariska Kurnia Rachmawati, M.Sc.

NIP. 19890811 201903 2 019

KATA PENGANTAR

Segala puji dan syukur bagi Allah *Subhanahu wa ta'ala* atas segala karunia dan rahmat-Nya yang telah diberikan. Sholawat serta salam senantiasa turunkan kepada Nabi Muhammad *Shalallahu 'alaihi wa salam*. Syukur alhamdulillah atas kehadiran Allah *Subhanahu wa ta'ala* atas limpahan nikmat-Nya, sehingga penulis dapat menyelesaikan Skripsi dengan judul "Pendekatan *Neural Network* dalam Peramalan Jumlah Penduduk Kota Semarang dengan Menggunakan Metode *Backpropagation*".

Skripsi ini merupakan syarat untuk menyelesaikan Sarjana Strata Satu (S1) di Fakultas Sains dan Teknologi, Universitas Islam Negeri Walisongo Semarang. Selama masa penyelesaian Skripsi ini, penulis mendapatkan dukungan dari berbagai pihak baik secara spiritual maupun moral, oleh karena itu penulis ingin berterima kasih kepada:

1. Dr. H. Ismail, M.Ag selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Walisongo Semarang.
2. Emy Siswanah, M.Sc selaku Ketua Program Studi Matematika.

3. Ahmad Aunur Rohman, M.Pd selaku Sekretaris program Studi Matematika.
4. Eva Khoirun Nisa, M.Si. selaku Dosen Pembimbing I yang telah meluangkan waktu dan pikirannya untuk memberikan masukan, bimbingan, dan pengarahan demi perbaikan skripsi ini.
5. Ariska Kurnia Rachmawati, M.Sc. selaku Dosen Pembimbing II yang telah meluangkan waktu dan pikirannya untuk memberikan masukan, bimbingan, dan pengarahan demi perbaikan skripsi ini.
6. Teristimewa dan terutama penulis sampaikan ucapan terima kasih kepada kedua orang tua penulis. Bapak Mustari dan Ibu Wahyuni yang senantiasa memberi semangat, perhatian, harapan, kasih sayang, dan doa tulus. Semoga apa yang telah mereka berikan menjadi ibadah dan cahaya penerang kehidupan di dunia dan di akhirat.
7. Seluruh teman-teman Matematika 2018 yang selalu memberikan semangat dan dukungan.
8. Kakak Antoni dan Putri juga adik Abdul Wahhab yang selalu menghibur hati penulis.

Dalam Skripsi ini penulis menyadari bahwa masih banyak kekurangan dan kelemahan, sehingga kritik dan saran yang membangun sangat diharapkan penulis demi

kesempurnaan Skripsi ini. Semoga skripsi ini dapat bermanfaat.

Semarang, 29 Desember 2022

Penulis

A handwritten signature in black ink, appearing to read 'Agustin' with a stylized flourish at the end.

Agustin Absari Wahyu Kuntarini

NIM. 1808046013

ABSTRAK

Kota Semarang merupakan salah satu daerah di Jawa Tengah yang tercatat memiliki luas wilayah 373,7 km². Sebagai salah satu kota metropolitan, Kota Semarang memiliki kepadatan penduduk yang cukup padat. Selama tahun 2010-2021 Kota Semarang memiliki jumlah penduduk yang berfluktuasi. Jumlah penduduk bersifat dinamis, tidak pernah tetap, tetapi selalu bertambah ataupun berkurang. Perlu dilakukan peramalan data jumlah penduduk di Kota Semarang agar perencanaan pembangunan di Kota Semarang dapat dirancang dengan lebih baik serta dapat mengatur fluktuasi jumlah penduduk di masa yang mendatang. Pada penelitian ini dilakukan analisa hasil peramalan jumlah penduduk Kota Semarang menggunakan pendekatan *Neural Network* dengan metode *Backpropagation*. Setelah dilakukan pelatihan dan pengujian didapatkan model arsitektur terbaik dengan 2 *neuron* pada lapisan *input*, 2 *neuron* pada lapisan tersembunyi, dan 1 *neuron* pada lapisan *output*. Berdasarkan hasil dari model arsitektur terbaik, didapatkan nilai MSE sebesar $9,39749 \times 10^{-6}$ dan nilai rata-rata MAPE sebesar 0,884552461% . Hasil evaluasi dengan nilai MAPE tersebut sangat akurat karena nilainya $< 10\%$. Pada penelitian ini didapatkan hasil peramalan jumlah penduduk Kota Semarang

tahun 2022-2025 berturut- turut adalah 1.863.121 jiwa, 1.878.634 jiwa, 1.891.865 jiwa, dan 1.902.947 jiwa.

Kata Kunci: Jumlah Penduduk, Peramalan, *Backpropagation Neural Network*.

DAFTAR ISI

PERNYATAAN KEASLIAN	i
PENGESAHAN	ii
NOTA DINAS	iii
NOTA DINAS	iv
KATA PENGANTAR.....	v
ABSTRAK.....	viii
DAFTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN.....	xvi
BAB I PENDAHULUAN	1
A. Latar Belakang.....	1
B. Rumusan Masalah	7
C. Batasan Masalah	8
D. Tujuan Penelitian.....	8
E. Manfaat Penelitian	9
BAB II LANDASAN PUSTAKA	10
A. Kajian Pustaka	10
B. Kajian Penelitian yang Relevan.....	36
BAB III METODE PENELITIAN	41
A. Jenis Penelitian	41

B.	Ruang Lingkup.....	41
C.	Pengumpulan Data.....	41
D.	Tahapan Penelitian.....	42
BAB IV HASIL DAN PEMBAHASAN.....		46
A.	Pengumpulan Data.....	46
B.	Pembagian Data.....	46
C.	Normalisasi Data.....	47
E.	Pengaturan Parameter Arsitektur <i>Backpropagation Neural Network</i>	49
D.	Pelatihan Jaringan.....	51
E.	Pengujian Jaringan.....	83
F.	Hasil Peramalan Jumlah Penduduk Kota Semarang Tahun 2022-2025.....	89
BAB V KESIMPULAN DAN SARAN.....		91
A.	KESIMPULAN.....	91
B.	SARAN.....	92
DAFTAR PUSTAKA.....		93
LAMPIRAN.....		98
RIWAYAT HIDUP.....		122

DAFTAR TABEL

Tabel 4. 1 <i>Input</i> dan <i>Output</i> Model Jaringan	46
Tabel 4. 2 Data Jumlah Penduduk Kota Semarang	47
Tabel 4. 3 Data Setelah Dinormalisasi	49
Tabel 4. 4 Nilai <i>Output</i> Data Pelatihan	82
Tabel 4. 5 Nilai <i>Error MSE</i> untuk 10 Model Pelatihan.....	83
Tabel 4. 6 Tabel Nilai <i>Output</i> Data Pengujian.....	84
Tabel 4. 7 Nilai <i>Error MSE</i> untuk 10 Model Pengujian.....	84
Tabel 4. 8 Hasil Nilai <i>Output</i> Data Model Arsitektur Terbaik	85
Tabel 4. 9 Hasil Peramalan Data Jumlah Penduduk.....	87
Tabel 4. 10 Hasil Nilai MAPE	88
Tabel 4. 11 Hasil Peramalan Jumlah Penduduk	89

DAFTAR GAMBAR

Gambar 3. 1 <i>Flowchart Neural Network</i> Menggunakan Metode <i>Backpropagation</i>	45
Gambar 4. 1 Model <i>Neural Network</i> Arsitektur 2-1-1.....	53
Gambar 4. 2 Hasil Pelatihan Jaringan dengan Arsitektur 2-1-1	53
Gambar 4. 3 Performa Data Pelatihan dengan Arsitektur 2-1-1	54
Gambar 4. 4 Tampilan <i>Regression Plot</i> dengan Arsitektur 2-1-1	55
Gambar 4. 5 Model <i>Neural Network</i> Arsitektur 2-2-1.....	56
Gambar 4. 6 Hasil Pelatihan Jaringan dengan Arsitektur 2-2-1	56
Gambar 4. 7 Performa Data Pelatihan dengan Arsitektur 2-2-1	57
Gambar 4. 8 Tampilan <i>Regression Plot</i> dengan Arsitektur 2-2-1	58
Gambar 4. 9 Model <i>Neural Network</i> Arsitektur 2-3-1.....	59
Gambar 4. 10 Hasil Pelatihan Jaringan dengan Arsitektur 2-3-1	59
Gambar 4. 11 Performa Data Pelatihan dengan Arsitektur 2-3-1	60
Gambar 4. 12 Tampilan <i>Regression Plot</i> dengan Arsitektur 2-3-1	61
Gambar 4. 13 Model <i>Neural Network</i> Arsitektur 2-4-1	62
Gambar 4. 14 Hasil Pelatihan Jaringan dengan Arsitektur 2-4-1	62
Gambar 4. 15 Performa Data Pelatihan dengan Arsitektur 2-4-1	63

Gambar 4. 16 Tampilan <i>Regression Plot</i> dengan Arsitektur 2-4-1	64
Gambar 4. 17 Model <i>Neural Network</i> Arsitektur 2-5-1	65
Gambar 4. 18 Hasil Pelatihan Jaringan dengan Arsitektur 2-5-1	65
Gambar 4. 19 Performa Data Pelatihan dengan Arsitektur 2-5-1	66
Gambar 4. 20 Tampilan <i>Regression Plot</i> dengan Arsitektur 2-5-1	67
Gambar 4. 21 Model <i>Neural Network</i> Arsitektur 2-6-1	68
Gambar 4. 22 Hasil Pelatihan Jaringan dengan Arsitektur 2-6-1	68
Gambar 4. 23 Performa Data Pelatihan dengan Arsitektur 2-6-1	69
Gambar 4. 24 Tampilan <i>Regression Plot</i> dengan Arsitektur 2-6-1	70
Gambar 4. 25 Model <i>Neural Network</i> Arsitektur 2-7-1	71
Gambar 4. 26 Hasil Pelatihan Jaringan dengan Arsitektur 2-7-1	71
Gambar 4. 27 Performa Data Pelatihan dengan Arsitektur 2-7-1	72
Gambar 4. 28 Tampilan <i>Regression Plot</i> dengan Arsitektur 2-7-1	72
Gambar 4. 29 Model <i>Neural Network</i> Arsitektur 2-8-1	73
Gambar 4. 30 Hasil Pelatihan Jaringan dengan Arsitektur 2-8-1	74
Gambar 4. 31 Performa Data Pelatihan dengan Arsitektur 2-8-1	75
Gambar 4. 32 Tampilan <i>Regression Plot</i> dengan Arsitektur 2-8-1	75
Gambar 4. 33 Model <i>Neural Network</i> Arsitektur 2-9-1	76

Gambar 4. 34 Hasil Pelatihan Jaringan dengan Arsitektur 2-9-1	77
Gambar 4. 35 Performa Data Pelatihan dengan Arsitektur 2-9-1	78
Gambar 4. 36 Tampilan <i>Regression Plot</i> dengan Arsitektur 2-9-1	78
Gambar 4. 37 Model <i>Neural Network</i> Arsitektur 2-10-1	79
Gambar 4. 38 Hasil Pelatihan Jaringan dengan Arsitektur 2-10-1.....	80
Gambar 4. 39 Performa Data Pelatihan dengan Arsitektur 2-10-1.....	81
Gambar 4. 40 Tampilan <i>Regression Plot</i> dengan Arsitektur 2-10-1.....	81
Gambar 4. 41 Grafik Perbandingan Hasil Peramalan dengan Data Asli.....	87

DAFTAR LAMPIRAN

Lampiran 1. Nilai <i>Error</i> pada Pelatihan Jaringan.....	98
Lampiran 2. Nilai <i>Error</i> pada Pengujian Jaringan.....	100
Lampiran 3. Syntax Program <i>Backpropagation Neural Network</i>	101
Lampiran 4. Perhitungan Manual <i>Backpropagation</i>	114

BAB I

PENDAHULUAN

A. Latar Belakang

Pada era saat ini, untuk melaksanakan perencanaan yang sukses dan efisien, peramalan adalah komponen penting. Peramalan digunakan untuk memprediksi ketidakpastian yang akan datang dan membuat keputusan yang lebih baik (Gurianto et al., 2016). Seiring dengan kemajuan teknologi, membuat semakin banyak dan berkembangnya teknik peramalan sesuai dengan kebutuhan yang diperlukan (Anitescu et al., 2019).

Salah satu teknik yang dapat diaplikasikan untuk peramalan yaitu *Neural Network* atau Jaringan Syaraf Tiruan. *Neural Network* telah menjadi topik yang sangat menarik karena kemampuannya untuk memecahkan masalah yang sulit, terutama dalam bidang pemrosesan gambar dan pengenalan objek, pengenalan suara, diagnosis medis, dan lain-lain (Anitescu et al., 2019). *Neural Network* merupakan metode yang banyak digunakan untuk pengklasifikasian, pengelompokkan, regresi, dan

melakukan peramalan data *time series* (Warsito, 2009). *Neural Network* merupakan model data yang mampu merepresentasikan hubungan *Input-Output* yang rumit karena dapat menangani berbagai masalah dan relatif mudah digunakan (Sudarsono, 2016).

Neural Network dapat digunakan untuk mendeteksi pola dan tren yang terlalu kompleks untuk diketahui oleh teknik komputer lainnya (Awodele & Jegede, 2009). Metode *Neural Network* ini mempunyai beberapa kelebihan, antara lain proses yang akurat, cepat, serta meminimalisasi kesalahan (Purwanto, 2016). Terdapat beberapa metode dalam *Neural Network* yang biasanya digunakan untuk peramalan diantaranya Adaline, Madaline, dan *Backpropagation* (Siang, 2009).

Metode Adaline memiliki beberapa unit *input* dan memiliki sebuah unit *output*. Metode Madaline merupakan modifikasi metode Adaline yang ditambah 1 lapisan tersembunyi diantara lapisan *input* dan *output* (Siang, 2009). Metode Adaline dan Madaline masih terlalu sederhana untuk digunakan dalam menyelesaikan permasalahan yang kompleks. Metode *Backpropagation* merupakan metode *Neural Network* yang memiliki unit *input*, unit *output*, dan juga

beberapa lapisan tersembunyi. Metode *Backpropagation* terbukti sudah banyak melakukan penyelesaian dalam permasalahan yang kompleks dan memiliki tingkat akurasi yang tinggi (Puspitaningrum, 2006).

Algoritma yang sering digunakan dalam mengatasi masalah yang rumit adalah pendekatan *Backpropagation* (Badrul, 2011). *Backpropagation* menyebabkan jaringan belajar secara efektif dengan menjembatani jarak antara *output* dan target yang diharapkan (Abiodun et al., 2019). *Backpropagation* mengajarkan jaringan untuk mengenali pola yang digunakan selama pelatihan dan untuk merespon dengan tepat pola *input* data yang menyerupai pola yang digunakan selama pelatihan sehingga metode *Backpropagation* memiliki tingkat akurasi yang cukup tinggi (Purwanto, 2016).

Di sektor ekonomi, teknik peramalan biasanya digunakan untuk memeriksa faktor-faktor seperti penduduk, tingkat inflasi, dan pasar keuangan (Gurianto et al., 2016). Data jumlah penduduk sangat dibutuhkan dalam perencanaan pembangunan. Pemerintah membutuhkan data jumlah penduduk untuk memperbaiki kondisi sosial ekonomi

masyarakat melalui perencanaan pembangunan. Mengingat bahwa perencanaan pembangunan baik ekonomi maupun sosial menyangkut tentang data jumlah serta karakteristik penduduk di masa depan, peramalan jumlah penduduk dianggap sebagai syarat untuk perencanaan pembangunan di berbagai bidang. Hasil peramalan digunakan untuk dua perencanaan. Yang pertama perencanaan yang bertujuan untuk menyediakan infrastuktur dasar dan yang kedua perencanaan yang bertujuan untuk mengubah tren penduduk menuju perkembangan demografi sosial ekonomi. Maka dari itu, data jumlah penduduk merupakan data penting yang harus dimiliki setiap daerah (Hardati, 2013).

Jumlah penduduk bersifat dinamis, tidak pernah statis, dan terus bertambah atau berkurang. Ekspansi penduduk akan berpengaruh terhadap struktur populasi suatu wilayah, baik secara kuantitas seperti jenis kelamin, usia angkatan kerja, dan secara kualitas seperti kesehatan, pendidikan, dan faktor lainnya. Jumlah penduduk yang semakin banyak bisa menjadi potensi dan sekaligus beban dalam pembangunan. Dengan luas wilayah yang tetap diiringi dengan jumlah penduduk yang semakin

bertambah dalam waktu bersamaan kepadatan penduduknya semakin besar. Apabila jumlah penduduk yang banyak tidak diikuti dengan peningkatan kualitas maka akan memunculkan berbagai masalah yang berkaitan dengan kebutuhan penduduk seperti terbatasnya lapangan pekerjaan, ruang terbuka hijau, dan kesejahteraan sosial (Hardati, 2013).

Kota Semarang merupakan salah satu kota metropolitan yang perencanaan tata ruangnya diprioritaskan karena pengaruhnya sangat penting bagi negara. Kota Semarang memiliki kepadatan penduduk yang cukup padat (BPS, 2019). Kota Semarang terletak di wilayah Jawa Tengah yang tercatat memiliki luas wilayah 373,7 km². Dari tahun 2010-2019 jumlah penduduk Kota Semarang selalu mengalami kenaikan. Data jumlah penduduk pada tahun 2015 sampai tahun 2019 berturut-turut adalah 1.701.114 jiwa, 1.729.083 jiwa, 1.757.686 jiwa, 1.786.114 jiwa, 1.814.110 jiwa. Tahun 2020 jumlah penduduk kota semarang mengalami penurunan yaitu dari 1.814.110 jiwa menjadi 1.653.524 jiwa (BPS, 2020). Tahun 2021 jumlah penduduknya kembali naik menjadi 1.656.564 jiwa (BPS, 2021). Dari data

tersebut bisa dilihat bahwa terjadi fluktuasi pada data jumlah penduduk di Kota Semarang. Diketahui dari data laju pertumbuhan penduduk, bahwa Kota Semarang memiliki laju pertumbuhan penduduk paling tinggi dibanding dengan kota lain di Jawa Tengah (BPS, 2020).

Pokok permasalahan utama pembangunan di Kota Semarang diantaranya adalah perlu ditingkatkannya kualitas sumber daya manusia, tata kelola pemerintahan belum diselenggarakan dengan optimal, belum optimalnya penyelesaian infrastruktur dasar dan pembangunan tata kelola ruang, perlunya peningkatan inovasi dan daya saing nilai tambah produksi pada sektor perekonomian. Penduduk merupakan subyek sekaligus obyek pembangunan, maka terdapat hubungan timbal balik antara penduduk dan pembangunan. Kebijakan kependudukan dan pembangunan yang efektif harus didasarkan pada bukti, yaitu temuan analisis data yang akurat, valid, dan relevan (Bappeda Kota Semarang, 2021). Peramalan data jumlah penduduk di Kota Semarang dilakukan dengan harapan perencanaan pembangunan di Kota Semarang dapat dirancang dengan lebih baik serta dapat mengatur

fluktuasi jumlah penduduk di masa yang mendatang, sehingga pokok permasalahan pembangunan yang terjadi di Kota Semarang tersebut bisa diatasi. Pada penelitian ini akan dilakukan peramalan jumlah penduduk dengan *Neural Network* metode *Backpropagation* karena *Neural Network* metode *Backpropagation* mempunyai nilai akurasi yang tinggi dibandingkan metode peramalan yang lain.

Berdasarkan uraian di atas penulis ingin mengetahui hasil peramalan jumlah penduduk Kota Semarang dengan menggunakan *Neural Network* metode *Backpropagation*. Untuk itu, penulis mengambil judul "**Pendekatan *Neural Network* dalam Peramalan Jumlah Penduduk Kota Semarang dengan Menggunakan Metode *Backpropagation***".

B. Rumusan Masalah

Berdasarkan latar belakang tersebut, maka dirumuskan permasalahan sebagai berikut:

1. Bagaimana model arsitektur terbaik untuk peramalan jumlah penduduk di Kota Semarang dengan pendekatan *Neural Network* menggunakan metode *Backpropagation*?

2. Bagaimana hasil peramalan jumlah penduduk di Kota Semarang dengan pendekatan *Neural Network* menggunakan metode *Backpropagation*?

C. Batasan Masalah

Adapun batasan masalah dalam penulisan penelitian ini adalah:

1. Ruang lingkup penelitian adalah jumlah penduduk wilayah Kota Semarang.
2. Data yang digunakan untuk melakukan penelitian yaitu, data jumlah penduduk perempuan, jumlah penduduk laki-laki, dan data jumlah penduduk secara keseluruhan tahun 2010-2021.

D. Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Mengetahui model arsitektur terbaik untuk peramalan jumlah penduduk di Kota Semarang dengan pendekatan *Neural Network* menggunakan metode *Backpropagation*.
2. Mengetahui hasil peramalan jumlah penduduk di Kota Semarang dengan pendekatan *Neural Network* menggunakan metode *Backpropagation*.

E. Manfaat Penelitian

Berikut beberapa manfaat yang diharapkan dari penelitian ini:

1. Bagi Penulis

Sebagai sumber belajar dan tambahan informasi peramalan jumlah penduduk.

2. Bagi Akademik

Sebagai referensi bacaan bagi yang mencari literatur di bidang peramalan.

3. Bagi Pemerintah

Sebagai bahan pembandingan hasil proyeksi jumlah penduduk Kota Semarang.

BAB II

LANDASAN PUSTAKA

A. Kajian Pustaka

1. Peramalan

Peramalan adalah teknik untuk memproyeksikan nilai di masa depan dengan mempertimbangkan data atau informasi yang relevan, baik data atau informasi terkait dari masa lalu maupun masa kini (Hutasuhut et al., 2014). Pendapat lain mengenai peramalan adalah cabang ilmu yang menggunakan informasi data dari masa lalu untuk membuat prediksi tentang kejadian masa depan yang belum terjadi (Luh et al., 2014). Berbagai disiplin ilmu, termasuk bisnis dan industri, ekonomi, pemerintahan, kedokteran, ilmu sosial, ilmu lingkungan, politik, dan keuangan semuanya menekankan pentingnya peramalan (Montgomery et al., 2015).

2. Penduduk

Penduduk adalah setiap individu yang telah berada dalam suatu daerah setidaknya selama enam bulan atau kurang dari itu dengan niat

untuk menetap (Sudarsono, 2016). Menurut Undang- Undang Republik Indonesia No. 23 Tahun 2006 pasal 1 ayat 2 menyebutkan bahwa penduduk adalah warga negara indonesia dan orang asing yang bertempat tinggal di Indonesia.

Jumlah penduduk di suatu daerah pasti sulit untuk diketahui. Hal ini berkaitan dengan tanggung jawab penduduk untuk mendaftarkan status kependudukannya di lingkungannya (Hardati, 2013). Menurut Mantra (2003) jumlah penduduk dapat ditentukan terutama dari tiga sumber: sensus penduduk, pendaftaran penduduk, dan survei demografis khusus. Kelahiran, kematian, emigrasi, perkawinan, perceraian, dan tindakan rujuk semuanya dicatat dalam berita acara pendaftaran penduduk. Sumber yang kedua adalah Sensus penduduk. Sensus penduduk adalah pengumpulan data yang dilakukan serentak di seluruh wilayah negara meliputi aspek demografis seperti umur, jenis kelamin, dan lain lain dan aspek non demografis seperti sosial, ekonomi, budaya, dan lingkungan. Sumber yang ketiga adalah survai khusus yang bersifat nasional untuk melengkapi dua sumber data sebelumnya.

Penduduk merupakan elemen kunci dalam pelaksanaan pembangunan karena berfungsi sebagai pelaksana sekaligus penerima pembangunan yang dituju. Ada empat faktor yang mempengaruhi pertumbuhan penduduk: kelahiran, kematian, dan migrasi (masuk dan keluar) (BPS, 2019).

Jumlah penduduk yang besar merupakan potensi pembangunan yang besar. Tetapi jumlah penduduk yang besar tidak menjamin bahwa pembangunan akan berhasil. Strategi pembangunan yang dilaksanakan akan terganggu oleh peningkatan jumlah penduduk yang signifikan tanpa peningkatan kesejahteraan. Jumlah penduduk bertujuan untuk mencapai keseimbangan antara jumlah, pertumbuhan, dan persebaran penduduk yang ideal sesuai dengan daya dukung serta keadaan perkembangan sosial, ekonomi, dan budaya. Dengan tersedianya fasilitas dan kesempatan pendidikan maka kualitas penduduk terbangun, sedangkan mobilitas penduduk bertujuan untuk pemerataan (Farisyah et al., 2019).

Kota Semarang berada di tengah-tengah pantai utara Pulau Jawa. Sebelah barat dibatasi dengan Kabupaten Kendal berbatasan di sebelah barat, Kabupaten Demak di sebelah timur, berbatasan dengan Kabupaten Semarang di sebelah selatan, dan berbatasan dengan Laut Jawa di sebelah utara. Kota Semarang memiliki luas wilayah sebesar 373,7 kilometer persegi yang terbagi atas 16 Kecamatan dan 177 Kelurahan. Pada tahun 2019, terdapat 1.689.655 jiwa yang bermukim di Kota Semarang, terdiri dari 837.526 laki-laki dan 852.129 perempuan. Jumlah penduduk Kota Semarang tumbuh hampir 1,26% selama tahun 2018 (BPS, 2019).

Di Kota Semarang, laju pertumbuhan penduduk sangat bervariasi menurut kecamatan. Hal ini terjadi karena karakteristik 16 Kabupaten yang beragam. Pertumbuhannya cenderung kecil di daerah yang dekat dengan pusat kota. Untuk kecamatan yang berada di pinggir kota pertumbuhan penduduknya cukup tinggi karena banyak terdapat pengembangan area perumahan atau industri (BPS, 2019).

Sebagai salah satu kota metropolitan Kota Semarang memiliki kepadatan penduduk yang cukup padat. Pada tahun 2019 kepadatan penduduknya sebesar 4.521 jiwa per km². Tiga kecamatan dengan kepadatan penduduk terendah jika dilihat secara individual adalah Tugu, Gunung Pati, dan Mijen. Walaupun Semarang mengalami laju pertumbuhan penduduk yang cukup tinggi, namun tingkat kepadatannya masih tergolong rendah karena dua dari tiga kecamatan tersebut merupakan daerah pertanian atau perkebunan dan juga termasuk wilayah yang dianggap paling luas di kota tersebut (BPS, 2019).

3. *Neural Network*

Neural Network merupakan suatu metode pembelajaran yang terinspirasi dari jaringan sistem pembelajaran biologis yang berlangsung dalam jaringan sel syaraf (*neuron*) yang saling terhubung satu sama lain. *Neural Network* adalah model nonlinear kompleks yang dapat diterapkan di banyak bidang penelitian seperti klasifikasi, aproksimasi fungsi, peramalan, dan pengelompokan (Khalil et al., 2019). Struktur *Neural Network* yang digunakan adalah metode

Backpropagation. *Backpropagation* menyebabkan jaringan belajar secara efektif dengan menjembatani jarak antara output dan target yang diharapkan (Abiodun et al., 2019). Metode ini memiliki dasar matematis yang kuat dan objektif. Metode *Backpropagation* sering digunakan untuk memecahkan masalah yang sulit (Badrul, 2011). Algoritma *Backpropagation* menentukan nilai koefisien dan bentuk persamaan dalam rumus dengan meminimalkan jumlah kuadrat galat *error* melalui model yang dihasilkan (*training set*) (Yanti, 2011).

a. Keunggulan dan Kelemahan *Neural Network*

Berikut beberapa keunggulan dari *Neural Network* (Puspitaningrum, 2006b):

- 1) Dapat memecahkan masalah yang sulit disimulasikan dengan menggunakan teknik analitikal logika seperti teknologi *software* standar.
- 2) *Neural Network* dapat belajar dari pengalaman.
- 3) Mampu memahami data yang dimasukkan meskipun data tersebut terkena gangguan atau kurang.

- 4) Lebih nyaman digunakan dan hemat biaya daripada menulis dengan keharusan untuk menulis program seperti *software* standar. Hal ini disebabkan *Neural Network* membutuhkan pelatihan jaringan untuk belajar dengan menginputkan data yang berisikan sekumpulan kasus ke dalam jaringan.

Neural Network juga memiliki kelemahan, beberapa diantaranya yaitu:

- 1) *Neural Network* masih membutuhkan campur tangan peneliti untuk menginputkan parameter dan menguji data.
 - 2) Teknik yang paling efisien untuk merepresentasikan input data, memilih model arsitektur yang ideal, mencari tahu berapa banyak node yang ada, dan mencari tahu berapa banyak layer yang belum ditemukan. Cara yang digunakan adalah dengan metode *trial and error*.
- b. Tahapan Pemrosesan Informasi dalam *Neural Network*

Secara garis besar *Neural Network* memiliki dua tahap pemrosesan informasi, yaitu tahap pelatihan dan tahap pengujian (Warsito, 2009).

1) Tahap Pelatihan

Tahap ini dimulai dengan memasukkan data pelatihan ke dalam jaringan. Dengan menggunakan data ini jaringan akan mengubah-ubah bobot yang menjadi penghubung antar node. Jaringan akan dilatih untuk dapat menghasilkan data sesuai dengan target yang diharapkan. Pada setiap iterasi dilakukan evaluasi.

2) Tahap pengujian

Pada tahap ini dilakukan pengujian terhadap data masukan yang belum pernah dilatihkan sebelumnya menggunakan bobot- bobot yang dihasilkan pada tahap pelatihan. Diharapkan bobot- bobot hasil pelatihan yang memiliki *error* minimal akan memberikan *error* yang kecil pada tahap pengujian.

c. Arsitektur Jaringan

Arsitektur jaringan pola keterkaitan antar *neuron*. Lapisan penyusun *Neural Network* terdiri atas tiga lapisan yaitu (Sudarsono, 2016):

1) Lapisan *Input (Input Layer)*

Unit-unit dalam lapisan *input* bertugas menerima pola masukkan dari luar yang menggambarkan suatu masalah.

2) Lapisan *Output (Output Layer)*

Unit-unit dalam lapisan *output* merupakan solusi dari *Neural Network* terhadap suatu permasalahan.

3) Lapisan Tersembunyi (*Hidden Layer*)

Lapisan tersembunyi terletak antara lapisan *output* dan lapisan *input*. Lapisan tersembunyi di sini berfungsi untuk membantu proses. Semakin banyak jumlah lapisan tersembunyi yang digunakan maka semakin bagus nilai output yang diinginkan, tetapi waktu pelatihan akan semakin lama. Menurut Siang (2009) tidak ada teori pasti untuk

menentukan jumlah lapisan tersembunyi. Tetapi bisa dicoba dengan jumlah lapisan paling kecil terlebih dahulu.

Berdasarkan jumlah lapisannya *Neural Network* memiliki dua bentuk arsitektur, diantaranya:

1) Jaringan Lapisan Tunggal (*Single Layer Network*)

Sebuah jaringan disebut jaringan lapisan tunggal jika jaringan tersebut terdiri dari 1 lapisan *input* dan 1 lapisan *output*. Setelah menerima *output*, secara langsung jaringan ini akan mengolahnya menjadi *output* tanpa melalui lapisan tersembunyi.

2) Jaringan Banyak Lapisan (*Multilayer Network*)

Jaringan *multilayer* merupakan perluasan dari jaringan lapisan tunggal. Jaringan ini terdiri dari lapisan *input* (*input layer*), lapisan tersembunyi (*hidden layer*), dan lapisan *output* (*ouput layer*). Dibanding jaringan lapis tunggal, jaringan *multilayer* dapat menyelesaikan

permasalahan yang lebih kompleks. Namun proses pelatihan data membutuhkan waktu yang lama. Akan tetapi, pada beberapa kasus pembelajaran pada jaringan *multilayer* ini lebih sukses dalam menyelesaikan masalah.

4. *Neuron*

Neuron merupakan unit pemrosesan informasi yang menjadi dasar dalam *Neural Network*. Terdapat 3 elemen pembentuk yang ada dalam *neuron*, diantaranya (Siang, 2009):

- a. Himpunan unit-unit yang dihubungkan dengan jalur koneksi yang memiliki nilai bobot yang berbeda disetiap jalurnya. Bobot yang nilainya positif akan menguatkan sinyal dan yang bernilai negatif akan memperlemah sinyal yang dibawa. Jumlah, struktur, dan pola antar unit akan menentukan arsitektur jaringan.
- b. Suatu unit penjumlah yang menjumlahkan sinyal inputan yang sudah dikalikan dengan bobotnya.

- c. Fungsi aktivasi yang menentukan diteruskannya sinyal dari input *neuron* ke *neuron* yang lain.

Penentuan jumlah *neuron* lapisan tersembunyi pada *Neural Network* sangat penting karena jumlah *neuron* pada lapisan tersembunyi berpengaruh terhadap hasil dari pelatihan jaringan. Jaringan dengan sebuah *neuron* tersembunyi sudah cukup untuk mengenali masukan dan target dengan tingkat ketelitian yang sudah ditentukan. Penambahan jumlah *neuron* lapisan tersembunyi kadang-kala membuat pelatihan menjadi lebih mudah (Siang, 2009). Oleh karena itu perlu adanya aturan untuk menentukan jumlah *neuron* pada lapisan tersembunyi (Heaton, 2008).

Terdapat banyak aturan yang digunakan untuk menentukan jumlah *neuron* pada lapisan tersembunyi. Aturan yang banyak digunakan diantaranya:

- a. Jumlah *neuron* pada lapisan tersembunyi lebih besar dari jumlah *input* atau *output*.
- b. Sebaiknya *neuron* pada lapisan tersembunyi berjumlah 2 atau 3 lebih

besar dari jumlah *input* ditambah jumlah *output*.

- c. Jumlah *neuron* pada lapisan tersembunyi harus lebih kecil atau sama besar dengan dua kali jumlah inputan.

5. Metode *Backpropagation*

Backpropagation merupakan metode sistematis untuk pelatihan Jaringan Banyak lapisan *Neural Network* (Badrul, 2011). Metode *Backpropagation* merupakan salah satu metode pelatihan yang mengatur bobot berdasarkan selisih antara target yang diinginkan dengan *output* yang diterima untuk mengurangi tingkat kesalahan (Puspitaningrum, 2006).

Metode *Backpropagation* banyak digunakan dalam berbagai aplikasi termasuk peramalan, pengenalan pola, dan pengoptimalan. Hal ini dimungkinkan karena metode *Backpropagation* menggunakan pembelajaran yang terawasi. Pola *input* dan target *output* diberikan sebagai sepasang data. Pelatihan bobot-bobot awal dengan melalui tahap maju untuk mendapat *error* keluaran yang kemudian *error* ini digunakan

sebagai tahap mundur untuk memperoleh nilai bobot yang sesuai agar dapat memperkecil nilai *error* sebagai target *output* yang dikehendaki dapat tercapai (Amri, 2015). Tujuan dari metode *Backpropagation* adalah untuk mencapai keseimbangan antara kapasitas jaringan untuk mengenali pola yang digunakan selama tahap pelatihan dan kapasitasnya untuk merespon secara tepat terhadap pola masukkan yang berbeda dari yang digunakan selama tahap pelatihan (Hadi Wijaya, 2019).

Backpropagation terdiri dari n buah *input* (ditambah sebuah bias), lapisan tersembunyi yang terdiri dari p unit (ditambah sebuah bias), serta m buah *output*. v_{j0} dan w_{k0} masing-masing merupakan bias untuk lapisan tersembunyi ke- j dan untuk *output* ke- k (Sudarsono, 2016).

6. Pelatihan *Backpropagation*

Pelatihan *Backpropagation* memiliki tiga fase. Fase yang pertama yaitu fase propagasi maju, fase kedua yaitu fase propagasi mundur, dan fase ketiga adalah fase perubahan bobot.

Secara rinci langkah- langkah dalam algoritma *Backpropagation* adalah sebagai berikut (Puspitaningrum, 2006b):

- a. Langkah 0: Inisialisasi bobot dengan bilangan acak kecil dengan menentukan angka pembelajaran (α). Menentukan nilai toleransi *error* yang inginkan dengan set maksimal *epoch* jika ingin membatasi jumlah *epoch* yang digunakan.
- b. Langkah 1: Jika kondisi penghentian belum terpenuhi maka lakukan langkah 2 sampai dengan 9.
- c. Langkah 2: Untuk setiap pasangan pola pelatihan, lakukan langkah ke-3 sampai langkah ke-9.
- d. Fase Propagasi Maju
 - 1) Langkah 3: Setiap unit *input* ($X_i, i = 1, 2, 3, \dots, n$) menerima sinyal *input* dan meneruskan sinyal tersebut ke setiap unit pada lapisan tersembunyi.
 - 2) Langkah 4: Setiap unit di lapisan tersembunyi ($Z_j, j = 1, 2, 3, \dots, m$) menjumlahkan sinyal- sinyal *input* yang berbobot, yaitu

$$Z_{netj} = v_{j0} + \sum_{i=1}^n x_i v_{ji} \quad (2.1)$$

Fungsi aktivasi untuk menghitung sinyal outputnya, yaitu:

$$Z_j = f(Z_{netj}) = \frac{1}{1+e^{-Z_{netj}}} \quad (2.2)$$

Dan mengirimkan sinyal tersebut ke semua unit pada lapisan *output*.

Keterangan:

Z_{netj} = *Output* unit lapisan tersembunyi

Z_j = Nilai fungsi aktivasi unit lapisan tersembunyi

v_{j0} = Bias untuk lapisan tersembunyi ke- j

x_i = Unit *input*

v_{ji} = Bobot dari lapisan input ke lapisan tersembunyi

- 3) Langkah 5: Setiap unit di lapisan *ouput* (Y_k , $k = 1, 2, 3, \dots, p$) menjumlahkan sinyal *hidden* yang berbobot, yaitu:

$$Y_{netk} = w_{k0} + \sum_{j=1}^m Z_j w_{kj} \quad (2.3)$$

Fungsi aktivasi untuk menghitung sinyal outputnya, yaitu:

$$Y_k = f(Y_{netk}) = \frac{1}{1+e^{-Y_{netk}}} \quad (2.4)$$

Keterangan:

Y_{netj} = Output unit lapisan *output*

Y_k = Nilai fungsi aktivasi unit lapisan *output*

Z_j = Output untuk lapisan tersembunyi ke- i

w_{k0} = Bias untuk output ke- k

w_{kj} = Nilai bobot antara lapisan tersembunyi dan lapisan *output*

e. Fase Propagasi Mundur

1) Langkah 6: Setiap unit di lapisan output

Y_k menerima pola target t_k untuk menghitung *error* (δ_k), yaitu:

$$\delta_k = (t_k - Y_k) f'(Y_{netk}) = (t_k - Y_k) Y_k (1 - Y_k) \quad (2.5)$$

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

$$\Delta w_{kj} = \alpha \delta_k Z_j \quad (2.6)$$

Akan dihitung juga koreksi bias yang digunakan untuk memperbaiki nilai bias antara lapisan tersembunyi dan lapisan *output* (w_{k0}), yaitu:

$$\Delta w_{k0} = \alpha \delta_k \quad (2.7)$$

Keterangan:

Δw_{k0} = Nilai perubahan bias untuk *output* ke-k

t_k = Target yang dicapai

Δw_{kj} = Nilai perubahan bobot antara lapisan tersembunyi dan lapisan output

δ_k = Nilai *error* lapisan *output*

α = Laju pemahaman

- 2) Langkah 7: menghitung faktor δ unit tersembunyi berdasarkan nilai *error* di setiap unit tersembunyi ($Z_j, j = 1, 2, \dots, m$)

$$\delta_{netj} = \sum_{k=1}^p \delta_k w_{kj} \quad (2.8)$$

Kemudian mengalikan nilai tersebut dengan fungsi aktivasi untuk menghitung nilai *error* pada lapisan tersembunyi (δ_j), yaitu:

$$\delta_j = \delta_{netj} f'(Z_{netj}) = \delta_{netj} Z_j (1 - Z_j) \quad (2.9)$$

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

$$\Delta v_{ji} = \alpha \delta_j x_i \quad (2.10)$$

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

$$\Delta v_{j0} = \alpha \delta_j \quad (2.11)$$

Keterangan:

δ_j =Nilai *error* pada lapisan tersembunyi

v_{ji} =Nilai bobot antara lapisan *input* dan lapisan tersembunyi

v_{j0} =Bias untuk lapisan tersembunyi ke- j

x_i =Unit *input*

α =Laju pemahaman

f. Fase Perubahan Bobot

Langkah 8: Setiap unit *output* (Y_k , $k = 1, 2, 3, \dots, p$) memperbaiki bobot dan bias, yaitu:

$$W_{kj}(\text{baru}) = W_{kj}(\text{lama}) + \Delta W_{kj} \quad (2.12)$$

Setiap unit lapisan tersembunyi (Z_j , $j = 1, 2, 3, \dots, m$) memperbaiki bobot dan biasnya, yaitu:

$$V_{ji}(\text{baru}) = V_{ji}(\text{lama}) + \Delta V_{ji} \quad (2.13)$$

Langkah 9: Uji syarat berhenti

7. Parameter *Backpropagation*

Dalam melakukan peramalan menggunakan metode *Backpropagation* terdapat beberapa parameter yang akan mengatur laju peramalan dalam lapisan tersembunyi, diantaranya:

a. Fungsi Aktivasi

Fungsi aktivasi digunakan untuk membatasi nilai *output* agar sesuai dengan batasan nilai outputnya (Prasetyo, 2012). Dalam *Backpropagation* fungsi aktivasi yang akan digunakan harus memenuhi syarat, yaitu: Kontinu, merupakan fungsi yang tidak turun, dan terdeferensial dengan mudah. Fungsi aktivasi yang sering digunakan dan memenuhi syarat tersebut adalah fungsi sigmoid biner (Sudarsono, 2016). Maka dari itu dalam penelitian ini digunakan fungsi sigmoid biner yang menghasilkan nilai *output* pada interval 0 hingga 1 dengan persamaan:

$$f(x) = \frac{1}{1+e^{-x}} \text{ dengan } f'(x) = f(x)(1 - f(x)) \quad (2.14)$$

Fungsi aktivasi sigmoid biner digunakan dengan alasan bahwa fungsi ini

dapat diturunkan dengan mudah (Siang, 2009).

b. *Epoch*

Epoch merupakan salah satu parameter yang digunakan sebagai *stopping condition*. *Epoch* mengacu pada kuantitas pengulangan atau iterasi yang dilakukan pada suatu pola hingga mendapatkan nilai *error* yang diinginkan. Proses perulangan akan berhenti jika nilai *epoch* melebihi *maksimum epoch* yang telah ditentukan dalam pembelajaran *Neural Network* (Kholis, 2015).

c. Bobot

Bobot atau *weight* memiliki fungsi sebagai penghubung antar lapisan. Pengaturan bobot dapat diatur untuk menghasilkan nilai keluaran yang diinginkan dari nilai input tertentu (Ryandhi, 2017).

d. Bias

Bias merupakan nilai ketidakpastian pada suatu node dalam lapisan. Bias

diberikan untuk setiap node pada lapisan tersembunyi (Ryandhi, 2017).

e. Error

Error merupakan nilai perbedaan yang terjadi antara *ouput* dan target yang diinginkan. Semakin kecil nilainya maka semakin dekat nilai *output* dan nilai target (Rahman, 2012).

f. Alpha

Parameter jaringan yang digunakan untuk mengatur proses penyesuaian bobot disebut *alfa* atau laju pembelajaran. (Rahman, 2012).

8. Pembagian Data

Backpropagation digunakan untuk mendapatkan keseimbangan antara pengenalan pola pelatihan dan respon yang baik terhadap pola lain yang sejenis (data pengujian). Data dibagi menjadi dua bagian, yaitu data yang digunakan untuk pelatihan dan data yang digunakan untuk pengujian. Komposisi data pelatihan dan data pengujian yang sering digunakan adalah sebagai berikut:

- a. 80% untuk data pelatihan dan 20% untuk data pengujian
- b. 70% untuk data pelatihan dan 30% untuk data pengujian
- c. $\frac{1}{3}$ untuk data pelatihan dan $\frac{2}{3}$ untuk data pengujian
- d. 50% untuk data pelatihan dan 50% untuk data pengujian
- e. 60% untuk data pelatihan dan 40% untuk data pengujian

Pembagian data harus diperhatikan agar jaringan memiliki data pelatihan secukupnya dan data pengujian dapat melakukan proses pengujian berdasarkan hasil pelatihan yang dilakukan. Jumlah data yang kurang pada proses pelatihan dapat menyebabkan jaringan mungkin tidak dapat mempelajari data dengan baik. Jumlah data yang terlalu banyak juga akan menjadi masalah karena akan memperlambat proses pemusatan (konvergensi) pada proses pelatihan. permasalahan overtraining (data pelatihan yang berlebihan) menyebabkan jaringan menghafal data yang diinputkan dari pada menggeneralisasi (Rufiyanti, 2015).

9. Normalisasi Data

Normalisasi adalah penskalaan terhadap nilai-nilai masuk kedalam suatu *range* tertentu. Pada proses peramalan menggunakan *Backpropagation Neural Network*, data *input* dan data target *output* harus dinormalisasi terlebih dahulu sebelum dilakukan tahap pelatihan. Proses normalisasi dilakukan supaya nilai *input* dan target *output* sesuai dengan *range* dari fungsi aktivasi yang digunakan dalam jaringan yakni sigmoid biner yang nilai fungsinya berada pada *range* 0 dan 1. Data yang digunakan dinormalisasikan dahulu ke interval yang lebih kecil yaitu antara 0,1 dan 0,9. (Kusumadewi & Purnomo, 2010). Rumus normalisasi datanya adalah sebagai berikut (Arofah et al., 2020):

$$X'_t = \frac{0,8(X_t - \min(x))}{(\max(x) - \min(x))} + 0,1 \quad (2.15)$$

Keterangan:

X'_t = nilai data setelah dinormalisasi

X_t = nilai data sebelum dinormalisasi

$\min(x)$ = nilai minimal dari keseluruhan data

$\max(x)$ = nilai maximal dari keseluruhan data

10. Metode Evaluasi Peramalan

Evaluasi digunakan untuk menilai keakuratan hasil peramalan. Banyak teknik yang dapat digunakan untuk melakukan evaluasi metode peramalan. Pada penelitian ini *Mean Squared Error* (MSE) dan *Mean Absolute Percentage Error* (MAPE) adalah metode yang digunakan dalam mengevaluasi hasil peramalan.

a. *Mean Squared Error* (MSE)

Mean Squared Error (MSE) merupakan metode perhitungan untuk menghitung rata-rata kesalahan berpangkat (Sukerti, 2015), dengan rumus (Montgomery et al., 2015):

$$MSE = \frac{\sum(\text{Actual}-\text{Forecast})^2}{n} \quad (2.16)$$

Keterangan:

n = jumlah data observasi

b. *Mean Absolute Percent Error* (MAPE)

Mean Absolute Precent Error (MAPE) merupakan metode yang digunakan untuk menghitung rata-rata persentase kesalahan mutlak pada setiap periode dibagi dengan nilai observasi yang nyata

untuk periode tersebut, dengan rumus (Montgomery et al., 2015):

$$MAPE = \frac{\sum |Actual - forecast|}{Actual} \times \frac{100}{n} \quad (2.17)$$

Semakin rendah nilai MAPE, kemampuan dari model peramalan yang digunakan dikatakan baik.

Interpretasi rentang nilai MAPE yang dihasilkan adalah (Moreno, Abad, dan Blasco, 2013):

- 1) $MAPE < 10\%$ = peramalan sangat akurat
- 2) $10\% \leq MAPE < 20\%$ = peramalan akurat
- 3) $20\% \leq MAPE < 50\%$ = peramalan masih dalam kewajaran
- 4) $MAPE \geq 50\%$ = peramalan tidak akurat

11. Denormalisasi Data

Denormalisasi data dilakukan untuk mengembalikan data yang masih ternormalisasi hasil dari jaringan yang berkisar antara 0,1 sampai 0,9 menjadi nilai normal yang sebenarnya. Berikut adalah rumusnya (Arofah et al., 2020):

$$X_t = \frac{(X'_t - 0,1)(\max(x_p) - \min(x_p))}{0,8} + \min(x_p) \quad (2.18)$$

Keterangan:

X'_t = nilai data sebelum didenormalisasi

X_t = nilai data setelah didenormalisasi

$\min(x_p)$ = nilai minimal dari keseluruhan data

$\max(x_p)$ = nilai maximal dari keseluruhan data

B. Kajian Penelitian yang Relevan

Penelitian terdahulu dengan menggunakan Neural Network beberapa diantaranya yaitu:

1. Anggi Hadi Wijaya (2019) dengan judul *Artificial Neural Network* untuk Memprediksi Beban Listrik dengan Menggunakan Metode *Backpropagation* dalam jurnal CoreIT. Dalam penelitiannya Anggi Hadi Wijaya menggunakan *Artificial Neural Network* metode *Backpropagation* dalam melakukan proses prediksi terhadap beban pemakaian listrik di masa yang akan datang pada PT. PLN Regional Sumatera Barat. Dalam memprediksi pemakaian listrik menggunakan Matlab dan pengujian dilakukan terhadap data *testing* tahun 2016 didapatkan nilai MSE 0,00095761. Artinya aplikasi Matlab menggunakan metode *Backpropagation* dapat

memprediksi beban pemakaian listrik jangka menengah dengan baik. Setelah dilakukan pengolahan data menggunakan aplikasi Matlab dan membandingkannya dengan tingkat akurasi dari hasil prediksi dengan data real diperoleh tingkat akurasi sebesar 96,48%. Penelitian tersebut dengan penulis sama-sama menggunakan *Backpropagation Neural Network*. Perbedaan penelitian tersebut dengan yang peneliti lakukan adalah pemilihan objek/ variabel penelitiannya.

2. Mohammed Al-Shawwa dan Samy S. Abu-Naser (2019) dengan judul Memprediksi Berat Badan Lahir Menggunakan Jaringan Syaraf Tiruan dalam *International Journal of Academic Health and Medical Research*. Dalam penelitian tersebut, model Jaringan Syaraf Tiruan dikembangkan dan diuji untuk memprediksi berat badan lahir. Sejumlah faktor yang mempengaruhi berat badan lahir diidentifikasi. Faktor- faktor seperti ras,usia, berat badan pada periode menstruasi terakhir, hipertensi, iritabilitas uterus, jumlah kunjungan dokter pada trisemester pertama, sebagai variabel input model Jaringan Syaraf Tiruan. Sebuah model berdasarkan topologi konsep *multilayer*

dikembangkan dan dilatih menggunakan data dari beberapa kasus kelahiran di rumah sakit. Evaluasi pengujian dataset menunjukkan bahwa model jaringan syaraf tiruan mampu memprediksi berat badan lahir dengan benar dengan nilai akurasi 100%. Penelitian tersebut dengan penulis sama-sama menggunakan *Neural Network/ Jaringan Syaraf Tiruan*. Perbedaan penelitian tersebut dengan yang peneliti lakukan adalah pemilihan objek/ variabel penelitiannya. Perbedaan lainnya, pada penelitian tersebut hanya menentukan model terbaik dalam *Neural Network*, tidak sampai peramalan untuk masa depan.

3. Aji Sudarsono (2016) dengan judul Jaringan Syaraf Tiruan untuk Memprediksi Laju Pertumbuhan Penduduk Menggunakan Metode *Backpropagation* (Studi Kasus di Kota Bengkulu) dalam Jurnal Media Infotama. Dalam penelitiannya Aji Sudarsono menggunakan jaringan syaraf tiruan dalam memprediksi laju pertumbuhan penduduk di Kota Bengkulu. Beliau melakukan prakiraan dengan 2 variabel input yaitu jumlah penduduk perempuan dan jumlah penduduk laki-laki. Didapatkan nilai MSE pada akhir penelitian

adalah sebesar 0,075491. Penelitian tersebut dengan penulis sama-sama menggunakan metode *Backpropagation* pada Matlab, tetapi dalam penelitiannya tidak ada variasi jumlah lapisan tersembunyi.

4. Devi Monika, dkk (2019) dengan judul Model Jaringan Syaraf Tiruan dalam Memprediksi Ketersediaan Cabai Berdasarkan Provinsi dalam jurnal TEKNIKA. Dalam penelitiannya Devi Monika dkk menggunakan Jaringan Syaraf Tiruan dengan metode *Backpropagation* untuk memprediksi ketersediaan tanaman cabai menurut Provinsi Indonesia tahun 2012-2017. Variabel input yang digunakan pada penelitiannya adalah data 2012 sebagai X1, data 2013 sebagai X2, data 2014 sebagai data X3, dan data 2015 sebagai data X4. Data target diambil dari data tahun 2016. Dilakukan 4 pelatihan dengan variasi lapisan tersembunyi yang model arsitekturnya adalah 4-2-1, 4-3-1, 4-4-1, dan 4-5-1. Didapatkan model arsitektur terbaik adalah 4-5-1 dengan nilai MSE 0,010651 dan tingkat akurasinya adalah 99,97%. Penelitian tersebut dengan penulis sama-sama menggunakan *Neural Network/ Jaringan Syaraf*

Tiruan. Perbedaan penelitian tersebut dengan penelitian ini adalah pemilihan objek/ variabel penelitian. Perbedaan lainnya adalah dalam penelitian tersebut hanya sampai menentukan model arsitektur terbaik saja.

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Penelitian yang dilakukan bersifat kuantitatif. Menurut Wahidmurni (2017) penelitian kuantitatif merupakan suatu teknik yang digunakan untuk menjawab masalah suatu penelitian yang berkaitan dengan data dan program statistik. Penelitian adalah suatu proses untuk penemuan pengetahuan yang menggunakan data dalam bentuk angka sebagai alat untuk menemukan keterangan mengenai apa yang ingin diketahui (Arikunto, 2010).

Penelitian ini menganalisis data jumlah penduduk Kota Semarang tahun 2010-2021 sebagai data pelatihan dan data pengujian.

B. Ruang Lingkup

Ruang lingkup pada penelitian ini adalah data jumlah penduduk Kota Semarang tahun 2010-2021. Data diperoleh dari Badan Pusat Statistik.

C. Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian adalah dokumentasi, yaitu mengambil data

dari Badan Pusat Statistik berupa data jumlah penduduk Kota Semarang tahun 2010-2021 yang meliputi data jumlah penduduk perempuan, data jumlah penduduk laki-laki, dan data jumlah penduduk secara keseluruhan.

D. Tahapan Penelitian

Tahapan penelitian pada peramalan jumlah penduduk Kota Semarang dengan pendekatan *Neural Network* adalah sebagai berikut:

1. Pengumpulan Data

Data yang dibutuhkan berasal dari Badan Pusat Statistik Kota Semarang, meliputi:

- a. Jumlah penduduk perempuan
- b. Jumlah penduduk laki-laki
- c. Jumlah keseluruhan penduduk

2. Pembagian data Pelatihan dan data Pengujian

Data jumlah penduduk Kota Semarang tahun 2010-2017 menjadi data pelatihan. Data pengujian pada penelitian ini adalah jumlah penduduk Kota Semarang tahun 2018-2021.

3. Normalisasi Data

Normalisasi data adalah kegiatan mengubah skala nilai data yang memiliki nilai sebenarnya

yang akan diinputkan supaya menjadi lebih kecil tanpa mengubah makna kandungan data tersebut.

4. Pengaturan Parameter Jaringan

Setelah penentuan data, akan dilakukan pengaturan parameter jaringan untuk mendapatkan arsitektur terbaik dalam analisa pengujian peramalan jumlah penduduk Kota Semarang.

5. Pelatihan Jaringan

Pelatihan jaringan dilakukan untuk meminimalkan nilai kesalahan sehingga model jaringan sesuai dan dapat dipakai untuk peramalan. Data tahun 2010-2017 digunakan sebagai data pelatihan model.

6. Pengujian Jaringan pada Data Pengujian

Pengujian jaringan ini menggunakan pemodelan arsitektur, parameter, fungsi aktivasi, dan bobot yang diperoleh dari proses pelatihan jaringan.

7. *Trial dan Error*

Trial dan error dilakukan dengan menentukan banyaknya jumlah layer tersembunyi. Dari hasil yang diperoleh, dipilih banyaknya lapisan tersembunyi yang dapat menghasilkan MSE dan

MAPE terkecil atau dengan kata lain dipilih model arsitektur terbaik (Arofah et al., 2020).

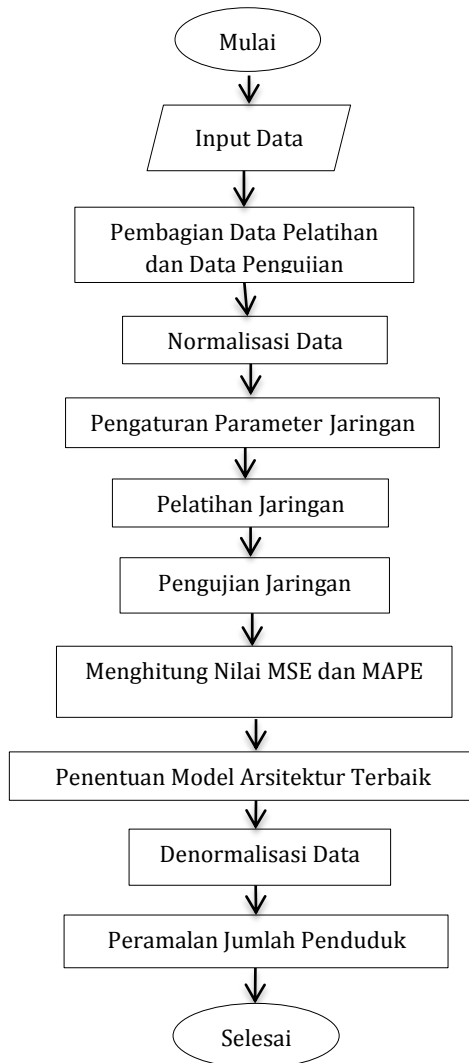
8. Denormalisasi Data

Denormalisasi data dilakukan untuk mengembalikan data yang masih ternormalisasi ke dalam bentuk angka desimal yang sebenarnya.

9. Peramalan Jumlah Penduduk

Dilakukan peramalan jumlah penduduk Kota Semarang pada tahun 2022-2025.

Berikut merupakan *flowchart* pada peramalan jumlah penduduk Kota Semarang dengan pendekatan *Neural Network* menggunakan metode *Backpropagation*:



Gambar 3.2 *Flowchart Neural Network Menggunakan Metode Backpropagation*

BAB IV

HASIL DAN PEMBAHASAN

A. Pengumpulan Data

Berikut merupakan variabel-variabel yang digunakan pada arsitektur meliputi *input* dan *output* model jaringan.

Tabel 4. 1 *Input* dan *Output* Model Jaringan

<i>Input</i>	X_1	Jumlah Penduduk Perempuan
	X_2	Jumlah Penduduk Laki- Laki
<i>Output</i>	Y_1	Total Jumlah Penduduk

B. Pembagian Data

Data yang meliputi data jumlah penduduk perempuan, data jumlah penduduk laki-laki, dan data jumlah keseluruhan penduduk pada tahun 2010-2021. Jumlah data sebanyak 36 data yang akan dibagi menjadi 2 bagian menggunakan teknik rasio yaitu 60% data pelatihan (data tahun 2010-2017) dan 40% data pengujian (data tahun 2018-2021). Pembagian data dilakukan untuk pengenalan pola data pelatihan dan merespon benar terhadap pola data masukan yang serupa (data pengujian). Pembagian data yang

sering digunakan adalah 60% data pelatihan dan 40% data pengujian.

Jumlah keseluruhan data tersaji pada Tabel 4. 2 sebagai berikut:

Tabel 4. 2 Data Jumlah Penduduk Kota Semarang

No	Tahun	Jumlah Penduduk Perempuan	Jumlah Penduduk Laki-Laki	Jumlah Keseluruhan
1	2010	795.426	764.587	1.560.013
2	2011	809.330	779.181	1.588.511
3	2012	823.766	792.728	1.616.494
4	2013	838.050	806.324	1.644.374
5	2014	852.521	820.473	1.672.994
6	2015	866.990	834.124	1.701.114
7	2016	881.168	847.915	1.729.083
8	2017	895.692	861.994	1.757.686
9	2018	910.539	875.575	1.786.114
10	2019	924.812	889.298	1.814.110
11	2020	835.083	818.441	1.653.524
12	2021	836.779	819.785	1.656.564

C. Normalisasi Data

Sebelum melakukan pengujian lebih lanjut menggunakan *Backpropagation Neural Network*, akan dilakukan normalisasi data sesuai dengan fungsi aktivasi yang digunakan dalam penelitian ini. Pada penelitian ini fungsi aktivasi yang digunakan

merupakan sigmoid biner yang nilai fungsinya berada pada *range* 0 dan 1. Data yang digunakan dinormalisasikan dahulu ke interval yang lebih kecil yaitu antara 0,1 dan 0,9.

Berdasarkan data pada Tabel 4. 2, nilai minimal data adalah 764.587 dan nilai maksimalnya adalah 1.814.110. Misalnya data pada tahun 2010 dengan persamaan normalisasi (2.15) maka akan didapat:

$$X'_t = \frac{0,8(X_t - \min(x))}{(\max(x) - \min(x))} + 0,1$$

$$X'_1 = \frac{0,8(795426 - 7644587)}{(1814110 - 7644587)} + 0,1 = 0,12350706$$

$$X'_2 = \frac{0,8(764587 - 7644587)}{(1814110 - 7644587)} + 0,1 = 0,1$$

$$Y' = \frac{0,8(1560013 - 7644587)}{(1814110 - 7644587)} + 0,1 = 0,706314297$$

Perhitungan dilakukan sampai semua data dinormalisasikan, baik data pelatihan maupun data pengujian. Hasil perhitungan tersaji pada Tabel 4. 3.

Tabel 4. 3 Data Setelah Dinormalisasi

Tahun	X_1	X_2	Y
2010	0,12350706	0,1	0,706314297
2011	0,134105398	0,111124292	0,728036927
2012	0,145109254	0,121450507	0,749366998
2013	0,155997248	0,131814072	0,770618557
2014	0,167027783	0,142599162	0,792434182
2015	0,178056793	0,153004651	0,813868681
2016	0,188863989	0,163516855	0,835188081
2017	0,199934923	0,174248587	0,856990747
2018	0,211252064	0,184600719	0,87866002
2019	0,222131673	0,19506109	0,9
2020	0,153735649	0,141050268	0,777593154
2021	0,155028427	0,142074733	0,779910397

Hasil normalisasi data pada Tabel 4. 3 akan digunakan dalam proses pelatihan dan pengujian.

E. Pengaturan Parameter Arsitektur *Backpropagation Neural Network*

Sebelum dilakukan pelatihan terhadap *Neural Network* dengan menggunakan Matlab maka ditentukan parameter berikut:

1. Parameter *epoch* adalah 1000

Epoch adalah parameter yang menentukan jumlah iterasi maksimum.

2. Parameter goal adalah 0

Goal adalah parameter yang digunakan untuk menentukan batas *error* yang ingin diraih.

3. Fungsi aktivasi yang digunakan adalah sigmoid biner dengan *range* 0 sampai 1

4. Jumlah *neuron* lapisan *input* adalah 2 yaitu X1 (jumlah penduduk perempuan) dan X2 (jumlah penduduk laki-laki).

5. Jumlah *neuron* pada lapisan tersembunyi adalah 1,2,3,4,5,6,7,8,9,10.

Jumlah *neuron* digunakan untuk pembentukan model arsitektur jaringan dan penentuan nilai MSE yang paling kecil. Jumlah *neuron* dipilih sesuai dengan ketentuan menurut Siang (2009) yang menyatakan bahwa dalam menentukan jumlah *neuron* tersembunyi tidak ada teori yang pasti dapat dipakai. Sebaiknya dapat dicoba dengan jaringan yang kecil misal terdiri dari 1 *neuron* tersembunyi. Selanjutnya jaringan dapat diperbesar dengan menambah jumlah *neuron*nya. Maka pada penelitian ini digunakan jumlah *neuron* 1,2,3,4,5,6,7,8,9,10 yang bertujuan untuk membandingkan mana yang mendapatkan model

arsitektur terbaik dengan melihat nilai MSE terkecil.

6. Jumlah *neuron* lapisan *output* adalah 1 yaitu Y (Jumlah keseluruhan penduduk).

D. Pelatihan Jaringan

Pelatihan jaringan akan dilakukan beberapa kali untuk mendapatkan jaringan terbaik dengan memodifikasi jumlah *neuron* pada lapisan tersembunyi. Pada penelitian ini dibentuk 10 model untuk mendapatkan arsitektur terbaik pada model yang dikembangkan diantaranya:

1. Model 1 (2-1-1) menggunakan 2 *neuron* pada lapisan *input*, 1 *neuron* pada lapisan tersembunyi, 1 *neuron* pada lapisan *output*.
2. Model 2 (2-2-1) menggunakan 2 *neuron* pada lapisan *input*, 2 *neuron* pada lapisan tersembunyi, 1 *neuron* pada lapisan *output*.
3. Model 3 (2-3-1) menggunakan 2 *neuron* pada lapisan *input*, 3 *neuron* pada lapisan tersembunyi, 1 *neuron* pada lapisan *output*.
4. Model 4 (2-4-1) menggunakan 2 *neuron* pada lapisan *input*, 4 *neuron* pada lapisan tersembunyi, 1 *neuron* pada lapisan *output*.

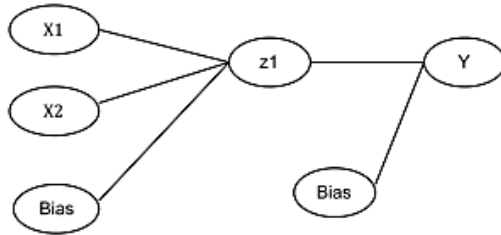
5. Model 5 (2-5-1) menggunakan 2 *neuron* pada lapisan *input*, 5 *neuron* pada lapisan tersembunyi, 1 *neuron* pada lapisan *output*.
6. Model 6 (2-6-1) menggunakan 2 *neuron* pada lapisan *input*, 6 *neuron* pada lapisan tersembunyi, 1 *neuron* pada lapisan *output*.
7. Model 7 (2-7-1) menggunakan 2 *neuron* pada lapisan *input*, 7 *neuron* pada lapisan tersembunyi, 1 *neuron* pada lapisan *output*.
8. Model 8 (2-8-1) menggunakan 2 *neuron* pada lapisan *input*, 8 *neuron* pada lapisan tersembunyi, 1 *neuron* pada lapisan *output*.
9. model 9 (2-9-1) menggunakan 2 *neuron* pada lapisan *input*, 9 *neuron* pada lapisan tersembunyi, 1 *neuron* pada lapisan *output*.
10. Model 10 (2-10-1) menggunakan 2 *neuron* pada lapisan *input*, 10 *neuron* pada lapisan tersembunyi, 1 *neuron* pada lapisan *output*.

Berikut hasil pada pelatihan jaringan:

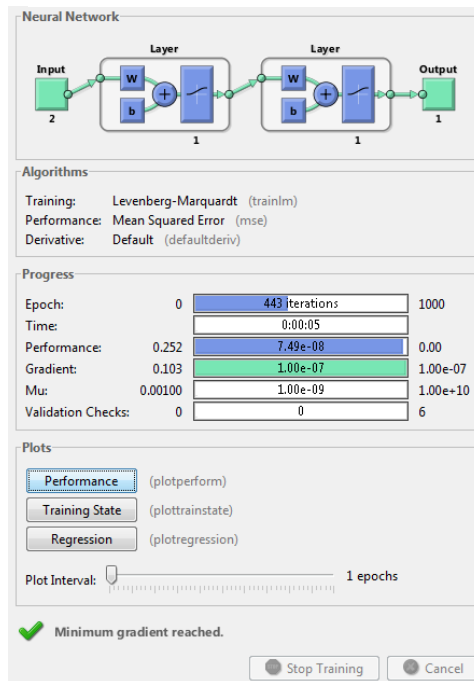
1. Pelatihan Model 1

Pelatihan Model 1 yaitu model arsitektur 2-1-1 menggunakan 2 *neuron* pada lapisan *input* (X1, X2) ditambah sebuah bias, 1 *neuron* pada lapisan

tersembunyi (z_1) ditambah sebuah bias, 1 *neuron* pada lapisan *output* (Y). Berikut adalah model *Neural Network* arsitektur 2-1-1:

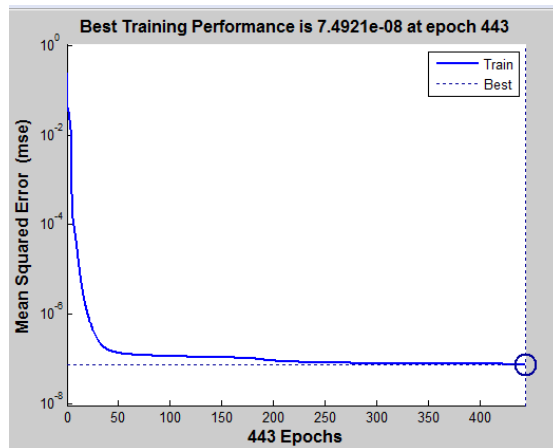


Gambar 4. 1 Model *Neural Network* Arsitektur 2-1-1



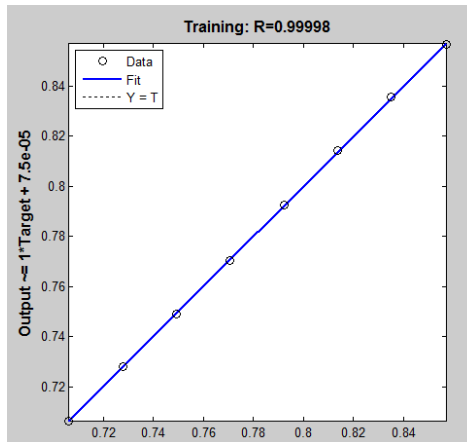
Gambar 4. 2 Hasil Pelatihan Jaringan dengan Arsitektur 2-1-1

Gambar 4.2 menunjukkan bahwa *epoch* yang terjadi memiliki 443 iterasi dan berlangsung selama 5 detik. Berikutnya akan ditampilkan grafik pada Gambar 4.3.



Gambar 4.3 Performa Data Pelatihan dengan Arsitektur 2-1-1

Metode pelatihan yang terbaik digambarkan secara grafik pada Gambar 4.3, yang menghasilkan nilai MSE sebesar $7,4921 \times 10^{-8}$ yang dicapai pada *epoch* ke 443.

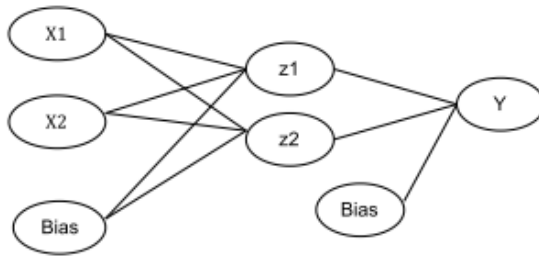


Gambar 4. 4 Tampilan *Regression Plot* dengan Arsitektur 2-1-1

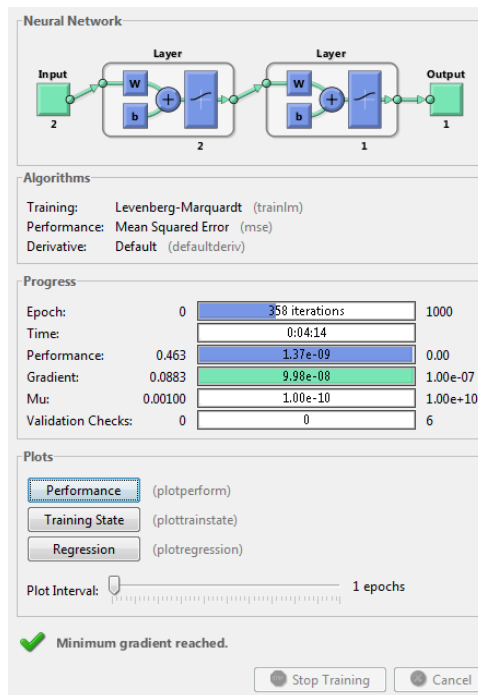
Tampilan plot regresi ditunjukkan pada Gambar 4. 4. Nilai korelasi untuk plot regresi adalah 0,99998. Mengingat nilainya mendekati 1, hal ini menunjukkan adanya hubungan yang kuat antara nilai output dengan nilai target.

2. Pelatihan Model 2

Pelatihan Model 2 yaitu model arsitektur 2-2-1 menggunakan 2 *neuron* pada lapisan *input* (X_1 , X_2) ditambah sebuah bias, 2 *neuron* pada lapisan tersembunyi (z_1 , z_2) ditambah sebuah bias, 1 *neuron* pada lapisan *output* (Y). Berikut adalah model *Neural Network* arsitektur 2-2-1:

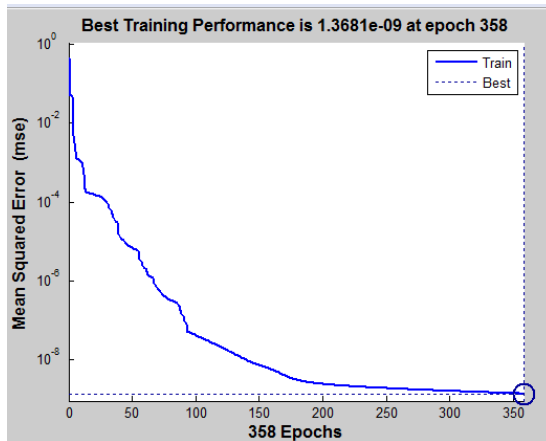


Gambar 4. 5 Model *Neural Network* Arsitektur 2-2-1



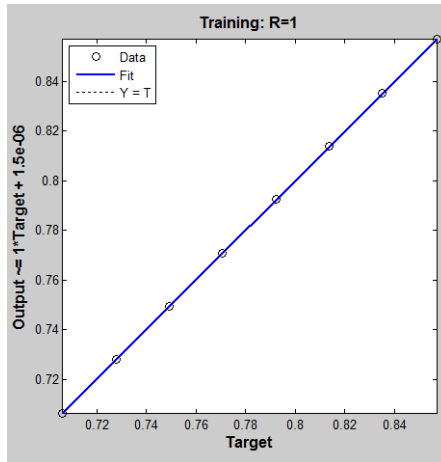
Gambar 4. 6 Hasil Pelatihan Jaringan dengan Arsitektur 2-2-1

Gambar 4. 6 menunjukkan bahwa *epoch* yang terjadi memiliki 358 iterasi dan berlangsung selama 4 menit 14 detik. Berikutnya akan ditampilkan grafik pada Gambar 4. 7.



Gambar 4. 7 Performa Data Pelatihan dengan Arsitektur 2-2-1

Metode pelatihan yang terbaik digambarkan secara grafik pada Gambar 4.7, yang menghasilkan nilai MSE sebesar $1,3681 \times 10^{-9}$ yang dicapai pada *epoch* ke 358.

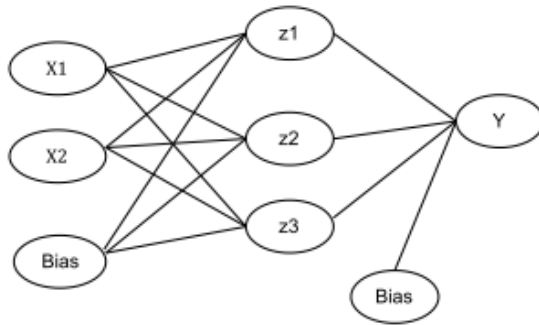


Gambar 4. 8 Tampilan *Regression Plot* dengan Arsitektur 2-2-1

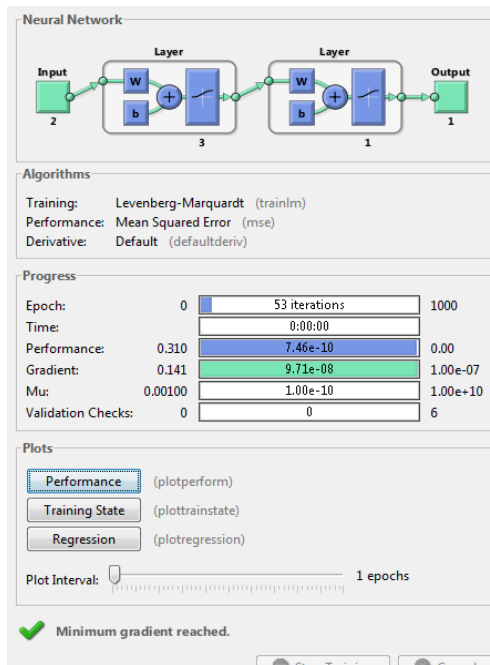
Tampilan plot regresi ditunjukkan pada Gambar 4. 8 merupakan tampilan *Regression Plot*. Nilai korelasi untuk plot regresi adalah 1. Mengingat nilainya sama dengan 1, hal ini menunjukkan adanya hubungan yang kuat antara nilai output dengan nilai target.

3. Pelatihan Model 3

Pelatihan Model 3 yaitu model arsitektur 2-3-1 menggunakan 2 *neuron* pada lapisan *input* (X_1 , X_2) ditambah sebuah bias, 3 *neuron* pada lapisan tersembunyi (z_1 , z_2 , z_3) ditambah sebuah bias, 1 *neuron* pada lapisan *output* (Y). Berikut adalah model *Neural Network* arsitektur 2-3-1:

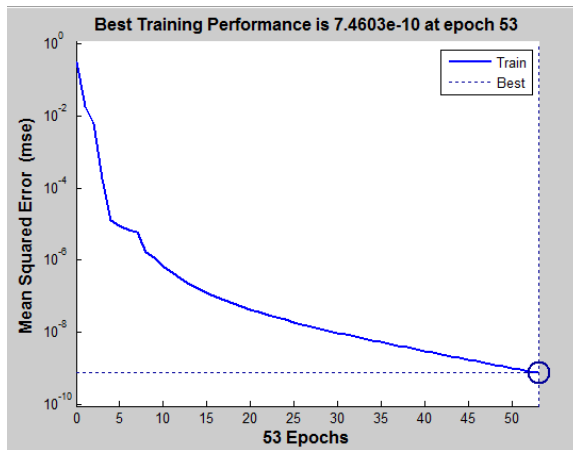


Gambar 4. 9 Model *Neural Network* Arsitektur 2-3-1



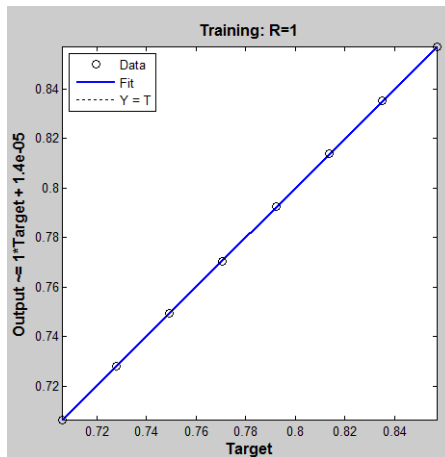
Gambar 4. 10 Hasil Pelatihan Jaringan dengan Arsitektur 2-3-1

Gambar 4. 10 menunjukkan bahwa *epoch* yang terjadi memiliki 53 iterasi dan berlangsung selama 0 detik. Berikutnya akan ditampilkan grafik pada Gambar 4.11.



Gambar 4. 11 Performa Data Pelatihan dengan Arsitektur 2-3-1

Metode pelatihan yang terbaik digambarkan secara grafik pada Gambar 4.11, yang menghasilkan nilai MSE sebesar $7,4603 \times 10^{-10}$ yang dicapai pada *epoch* ke 53.

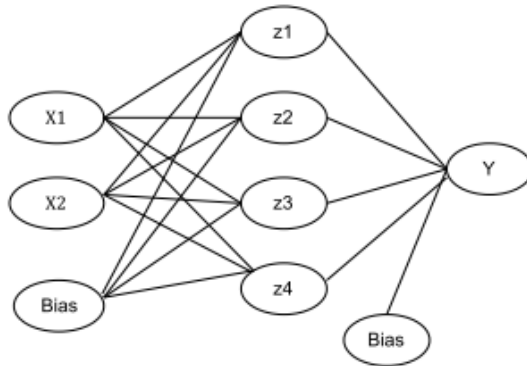


Gambar 4. 12 Tampilan *Regression Plot* dengan Arsitektur 2-3-1

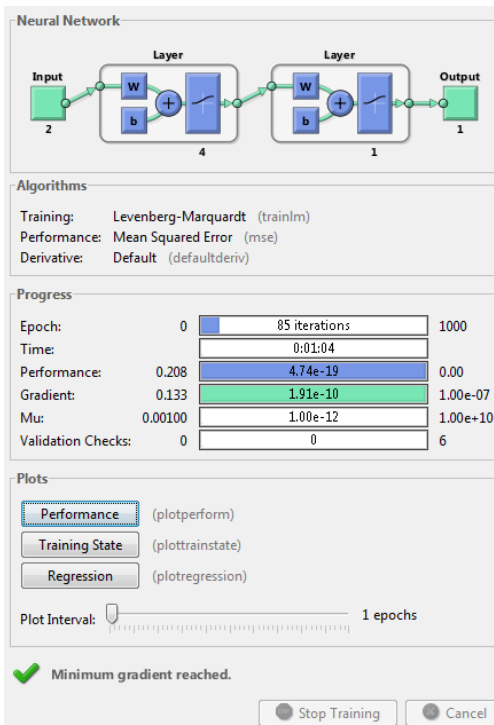
Tampilan plot regresi ditunjukkan pada Gambar 4. 12 merupakan tampilan *Regression Plot*. Nilai korelasi untuk plot regresi adalah 1. Mengingat nilainya sama dengan 1, hal ini menunjukkan adanya hubungan yang kuat antara nilai output dengan nilai target.

4. Pelatihan Model 4

Pelatihan Model 4 yaitu model arsitektur 2-4-1 menggunakan 2 *neuron* pada lapisan *input* (X_1 , X_2) ditambah sebuah bias, 4 *neuron* pada lapisan tersembunyi (z_1 , z_2 , z_3 , z_4) ditambah sebuah bias, 1 *neuron* pada lapisan *output* (Y). Berikut adalah model *Neural Network* arsitektur 2-4-1:

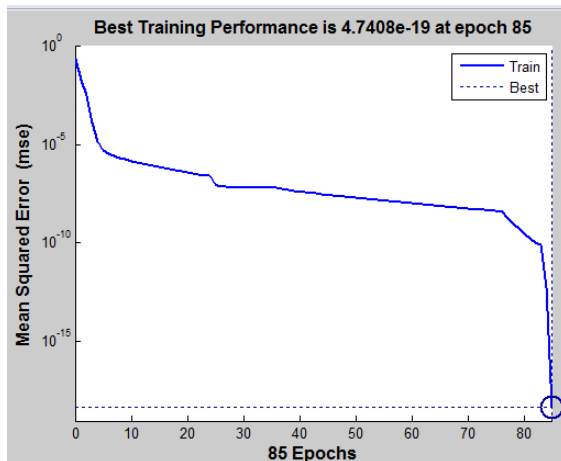


Gambar 4. 13 Model *Neural Network* Arsitektur 2-4-1



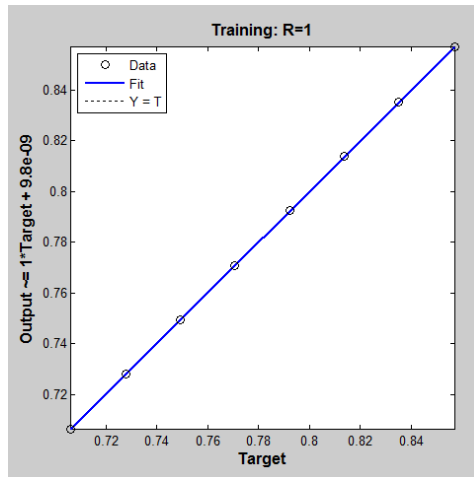
Gambar 4. 14 Hasil Pelatihan Jaringan dengan Arsitektur 2-4-1

Gambar 4.14 menunjukkan bahwa *epoch* yang terjadi memiliki 85 iterasi dan berlangsung selama 1 menit 4 detik. Berikutnya akan ditampilkan grafik pada Gambar 4.15.



Gambar 4. 15 Performa Data Pelatihan dengan Arsitektur 2-4-1

Metode pelatihan yang terbaik digambarkan secara grafik pada Gambar 4.15, yang menghasilkan nilai MSE $4,7408 \times 10^{-19}$ yang dicapai pada *epoch* ke 85.

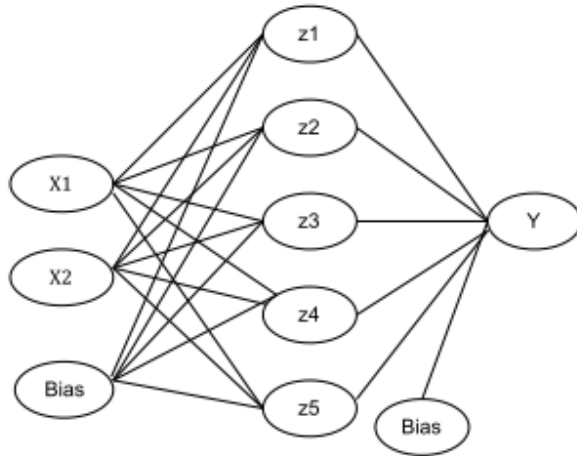


Gambar 4. 16 Tampilan *Regression Plot* dengan Arsitektur 2-4-1

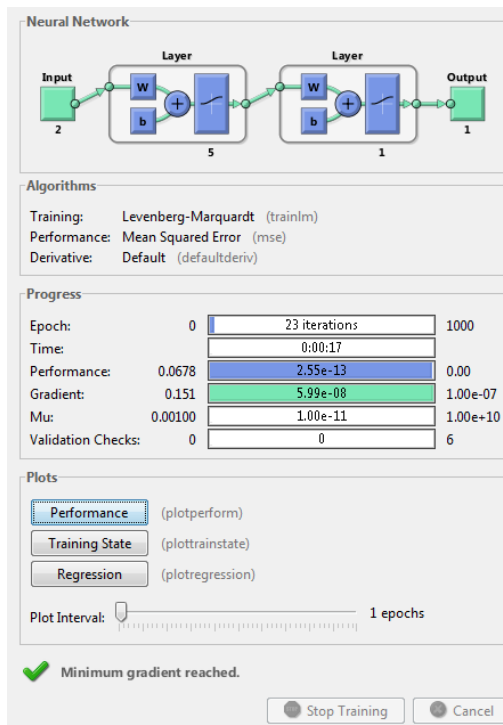
Tampilan plot regresi ditunjukkan pada Gambar 4. 16 merupakan tampilan *Regression Plot*. Nilai korelasi untuk plot regresi adalah 1. Mengingat nilainya sama dengan 1, hal ini menunjukkan adanya hubungan yang kuat antara nilai output dengan nilai target.

5. Pelatihan Model 5

Pelatihan Model 5 yaitu model arsitektur 2-5-1 menggunakan 2 *neuron* pada lapisan *input* (X_1 , X_2) ditambah sebuah bias, 4 *neuron* pada lapisan tersembunyi (z_1 , z_2 , z_3 , z_4 , z_5) ditambah sebuah bias, 1 *neuron* pada lapisan *output* (Y). Berikut adalah model *Neural Network* arsitektur 2-5-1:

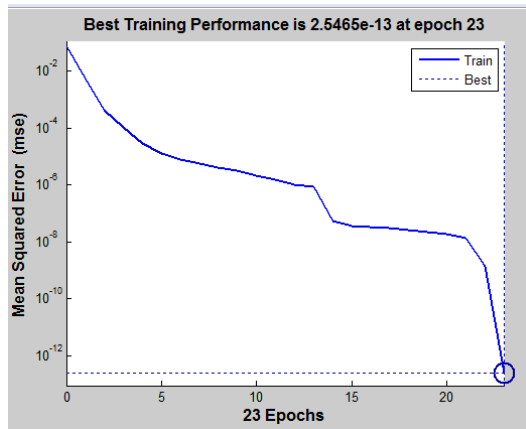


Gambar 4. 17 Model *Neural Network* Arsitektur 2-5-1



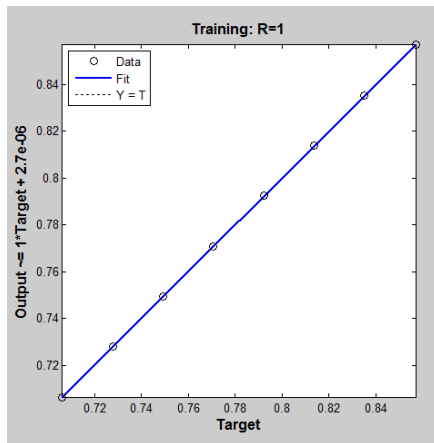
Gambar 4. 18 Hasil Pelatihan Jaringan dengan Arsitektur 2-5-1

Gambar 4.18 menunjukkan bahwa *epoch* yang terjadi memiliki 23 iterasi dan berlangsung selama 17 detik. Berikutnya akan ditampilkan grafik pada Gambar 4.19.



Gambar 4. 19 Performa Data Pelatihan dengan Arsitektur 2-5-1

Metode pelatihan yang terbaik digambarkan secara grafik pada Gambar 4.19, yang menghasilkan nilai MSE sebesar $2,5465 \times 10^{-13}$ yang dicapai pada *epoch* ke 23.

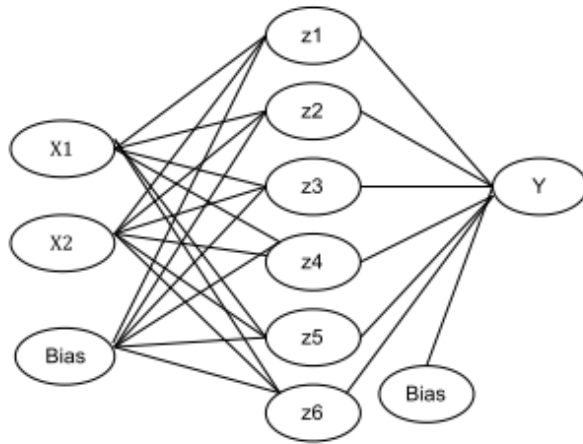


Gambar 4. 20 Tampilan *Regression Plot* dengan Arsitektur 2-5-1

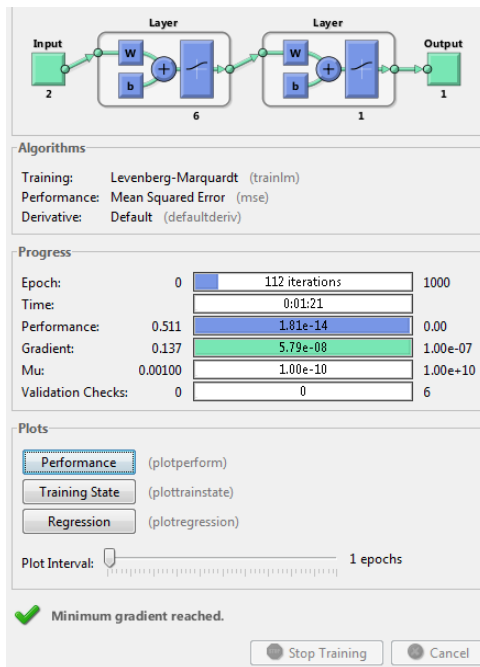
Tampilan plot regresi ditunjukkan pada Gambar 4. 20 merupakan tampilan *Regression Plot*. Nilai korelasi untuk plot regresi adalah 1. Mengingat nilainya sama dengan 1, hal ini menunjukkan adanya hubungan yang kuat antara nilai output dengan nilai target.

6. Pelatihan Model 6

Pelatihan Model 6 yaitu model arsitektur 2-6-1 menggunakan 2 *neuron* pada lapisan *input*(X_1, X_2) ditambah sebuah bias, 6 *neuron* pada lapisan tersembunyi ($z_1, z_2, z_3, z_4, z_5, z_6$) ditambah sebuah bias, 1 *neuron* pada lapisan *output* (Y). Berikut adalah model *Neural Network* arsitektur 2-6-1:

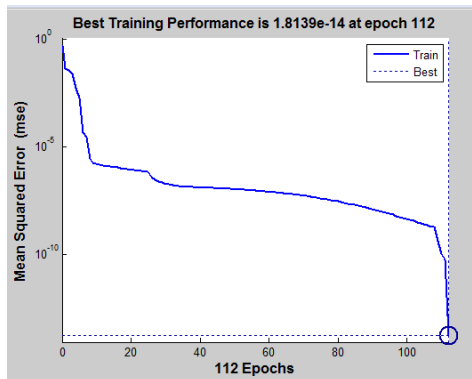


Gambar 4. 21 Model *Neural Network* Arsitektur 2-6-1



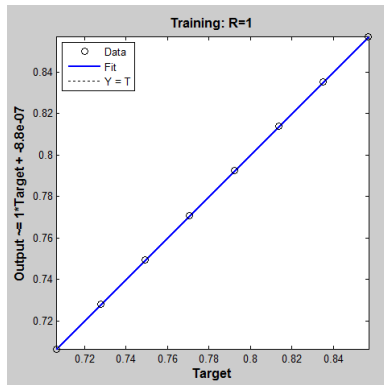
Gambar 4. 22 Hasil Pelatihan Jaringan dengan Arsitektur 2-6-1

Gambar 4.22 menunjukkan bahwa *epoch* yang terjadi memiliki 12 iterasi dan berlangsung selama 1 menit 21 detik. Berikutnya akan ditampilkan grafik pada Gambar 4.23.



Gambar 4. 23 Performa Data Pelatihan dengan Arsitektur 2-6-1

Metode pelatihan yang terbaik digambarkan secara grafik pada Gambar 4.23, yang menghasilkan nilai MSE sebesar $1,8139 \times 10^{-14}$ yang dicapai pada *epoch* ke 112.

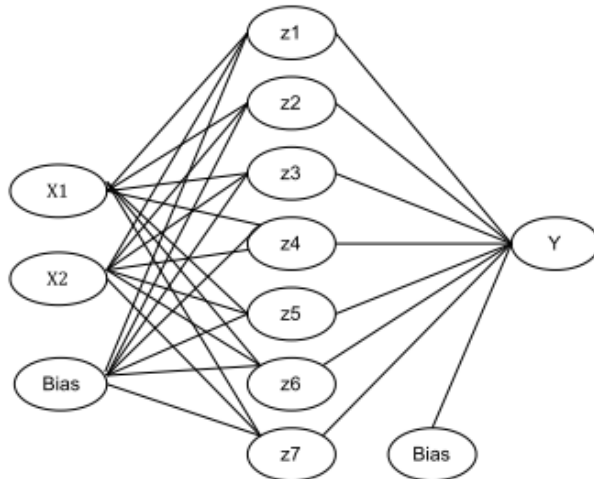


Gambar 4. 24 Tampilan *Regression Plot* dengan Arsitektur 2-6-1

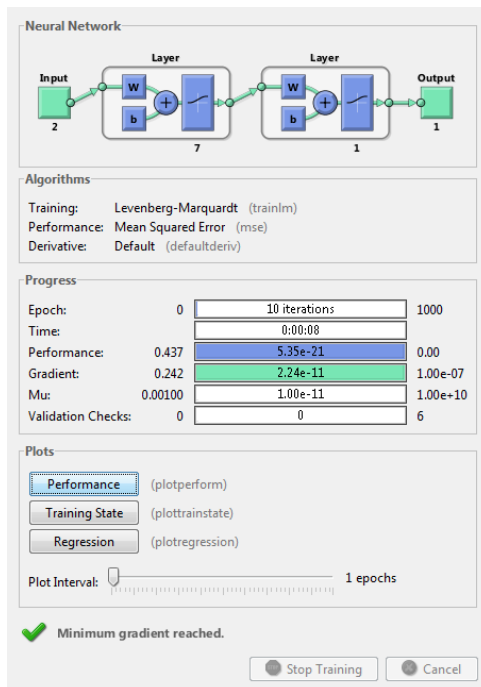
Tampilan plot regresi ditunjukkan pada Gambar 4. 24 merupakan tampilan *Regression Plot*. Nilai korelasi untuk plot regresi adalah 1. Mengingat nilainya sama dengan 1, hal ini menunjukkan adanya hubungan yang kuat antara nilai output dengan nilai target.

7. Pelatihan Model 7

Pelatihan Model 7 yaitu model arsitektur 2-7-1 menggunakan 2 *neuron* pada lapisan *input* (X_1 , X_2) ditambah sebuah bias, 2 *neuron* pada lapisan tersembunyi (z_1 , z_2 , z_3 , z_4 , z_5 , z_6 , z_7) ditambah sebuah bias, 1 *neuron* pada lapisan *output* (Y). Berikut adalah model *Neural Network* arsitektur 2-7-1:

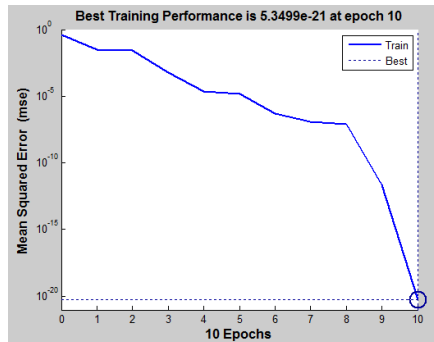


Gambar 4. 25 Model *Neural Network* Arsitektur 2-7-1



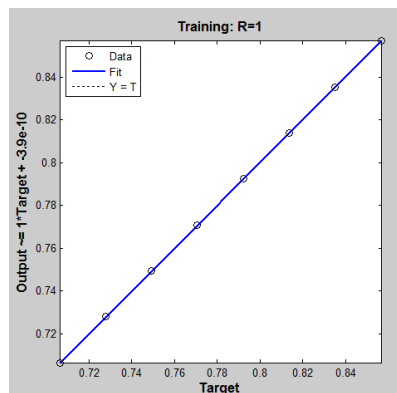
Gambar 4. 26 Hasil Pelatihan Jaringan dengan Arsitektur 2-7-1

Gambar 4.26 menunjukkan bahwa *epoch* yang terjadi memiliki 10 iterasi dan berlangsung selama 8 detik. Berikutnya akan ditampilkan grafik pada Gambar 4.27.



Gambar 4. 27 Performa Data Pelatihan dengan Arsitektur 2-7-1

Metode pelatihan yang terbaik digambarkan secara grafik pada Gambar 4.27, yang menghasilkan nilai MSE sebesar $5,3499 \times 10^{-21}$ yang dicapai pada *epoch* ke 10.

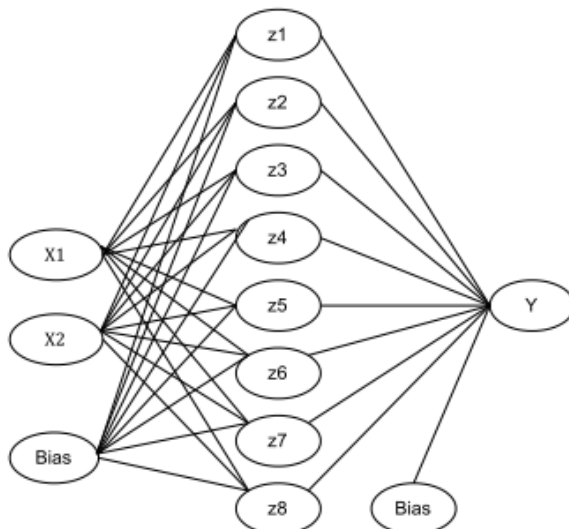


Gambar 4. 28 Tampilan *Regression Plot* dengan Arsitektur 2-7-1

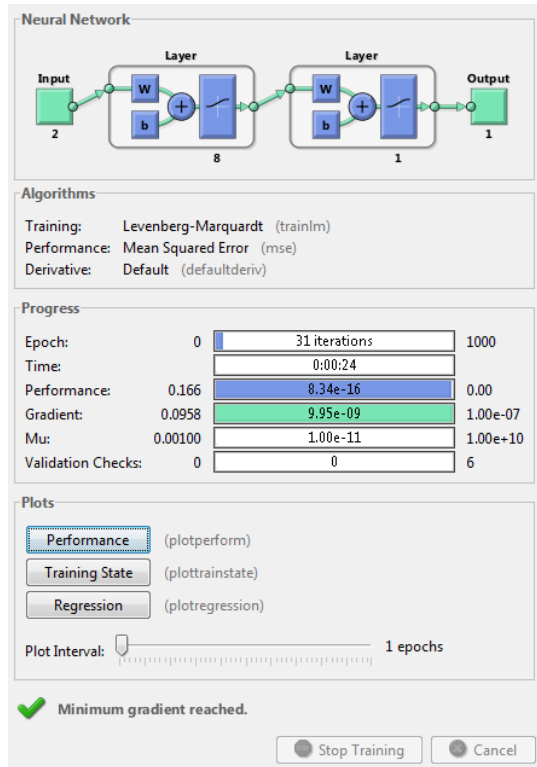
Tampilan plot regresi ditunjukkan pada Gambar 4.28 merupakan tampilan *Regression Plot*. Nilai korelasi untuk plot regresi adalah 1. Mengingat nilainya sama dengan 1, hal ini menunjukkan adanya hubungan yang kuat antara nilai output dengan nilai target..

8. Pelatihan Model 8

Pelatihan Model 8 yaitu model arsitektur 2-8-1 menggunakan 2 *neuron* pada lapisan *input* (X_1 , X_2) ditambah sebuah bias, 8 *neuron* pada lapisan tersembunyi (z_1 , z_2 , z_3 , z_4 , z_5 , z_6 , z_7 , z_8) ditambah sebuah bias, 1 *neuron* pada lapisan *output* (Y). Berikut adalah model *Neural Network* arsitektur 2-8-1:

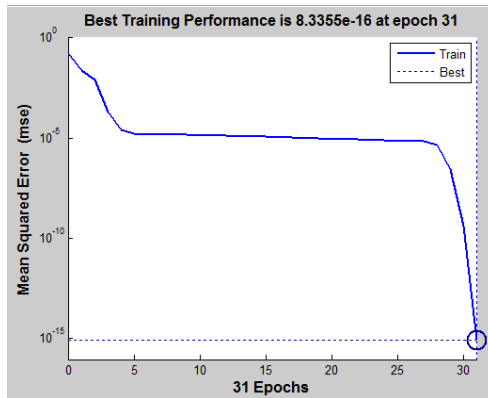


Gambar 4. 29 Model *Neural Network* Arsitektur 2-8-1



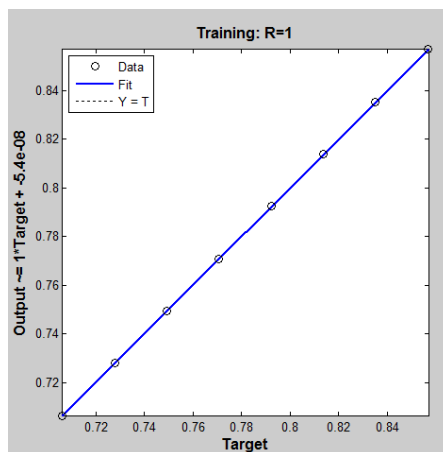
Gambar 4. 30 Hasil Pelatihan Jaringan dengan Arsitektur 2-8-1

Gambar 4.30 menunjukkan bahwa *epoch* yang terjadi memiliki 31 iterasi dan berlangsung selama 24 detik. Berikutnya akan ditampilkan grafik pada Gambar 4.31.



Gambar 4. 31 Performa Data Pelatihan dengan Arsitektur 2-8-1

Metode pelatihan yang terbaik digambarkan secara grafik pada Gambar 4.31, yang menghasilkan nilai MSE sebesar $8,3355 \times 10^{-16}$ yang dicapai pada *epoch* ke 31.

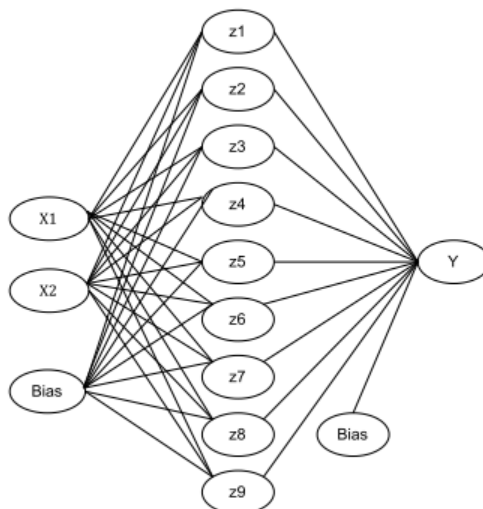


Gambar 4. 32 Tampilan *Regression Plot* dengan Arsitektur 2-8-1

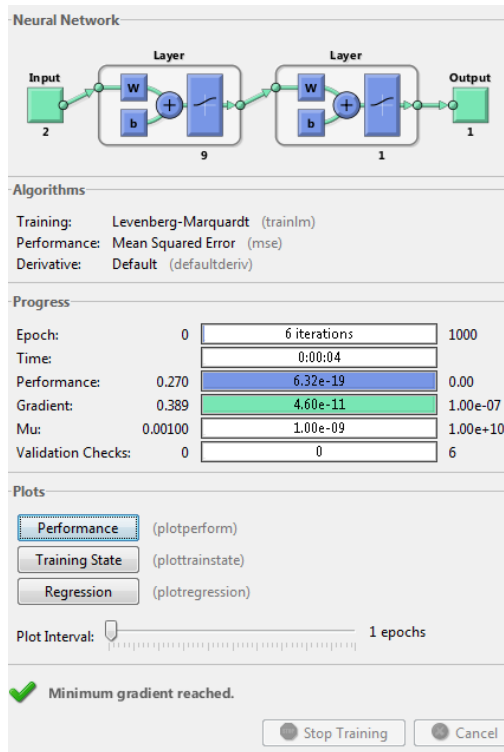
Tampilan plot regresi ditunjukkan pada Gambar 4.32 merupakan tampilan *Regression Plot*. Nilai korelasi untuk plot regresi adalah 1. Mengingat nilainya sama dengan 1, hal ini menunjukkan adanya hubungan yang kuat antara nilai output dengan nilai target.

9. Pelatihan Model 9

Pelatihan Model 9 yaitu model arsitektur 2-9-1 menggunakan 2 *neuron* pada lapisan *input* (X_1 , X_2) ditambah sebuah bias, 9 *neuron* pada lapisan tersembunyi (z_1 , z_2 , z_3 , z_4 , z_5 , z_6 , z_7 , z_8 , z_9) ditambah sebuah bias, 1 *neuron* pada lapisan *output* (Y). Berikut adalah model *Neural Network* arsitektur 2-9-1:

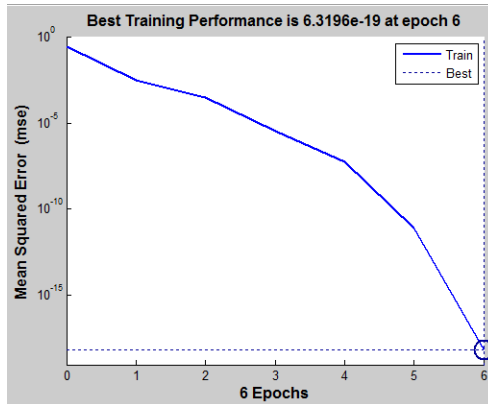


Gambar 4. 33 Model *Neural Network* Arsitektur 2-9-1



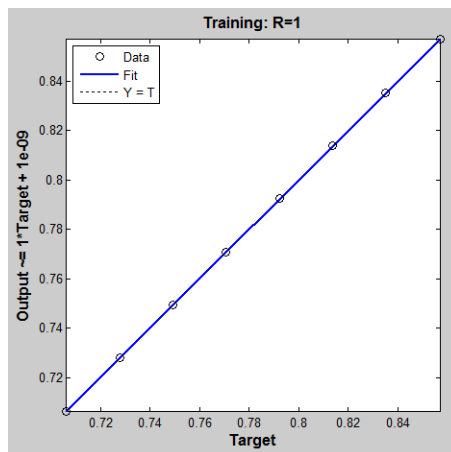
Gambar 4. 34 Hasil Pelatihan Jaringan dengan Arsitektur 2-9-1

Gambar 4.34 menunjukkan bahwa *epoch* yang terjadi memiliki 6 iterasi dan berlangsung selama 4 detik. Berikutnya akan ditampilkan grafik pada Gambar 4.35.



Gambar 4. 35 Performa Data Pelatihan dengan Arsitektur 2-9-1

Metode pelatihan yang terbaik digambarkan secara grafik pada Gambar 4.35, yang menghasilkan nilai MSE $6,3196 \times 10^{-19}$ yang dicapai pada *epoch* ke 6.

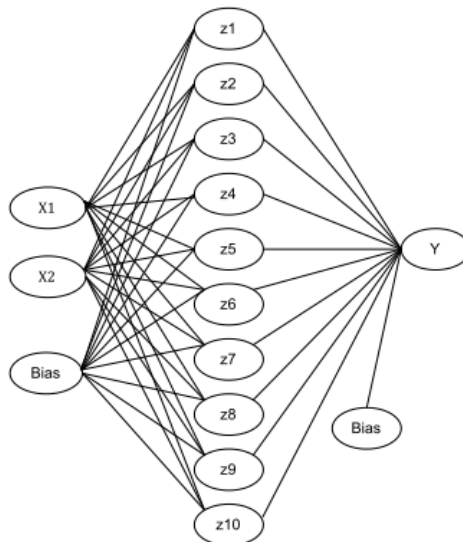


Gambar 4. 36 Tampilan *Regression Plot* dengan Arsitektur 2-9-1

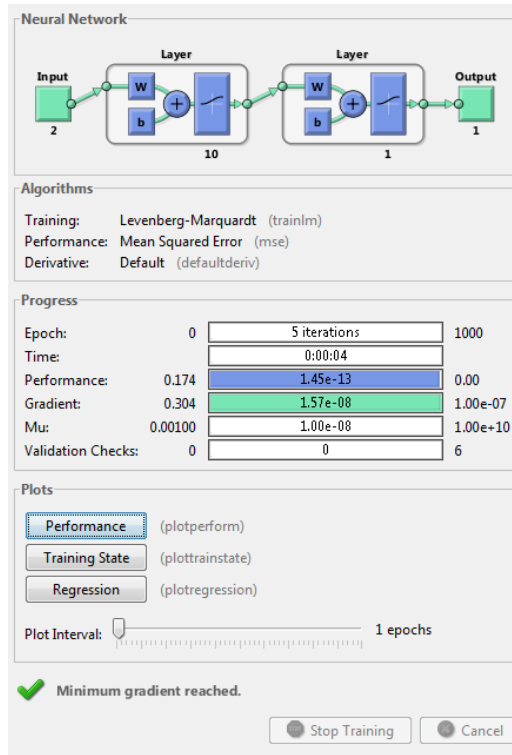
Tampilan plot regresi ditunjukkan pada Gambar 4. 36 merupakan tampilan *Regression Plot*. Nilai korelasi untuk plot regresi adalah 1. Mengingat nilainya sama dengan 1, hal ini menunjukkan adanya hubungan yang kuat antara nilai output dengan nilai target.

10. Pelatihan Model 10

Pelatihan Model 10 yaitu model arsitektur 2-10-1 menggunakan 2 *neuron* pada lapisan *input* (X_1 , X_2) ditambah sebuah bias, 10 *neuron* pada lapisan tersembunyi (z_1 , z_2 , z_3 , z_4 , z_5 , z_6 , z_7 , z_8 , z_9 , z_{10}) ditambah sebuah bias, 1 *neuron* pada lapisan *output* (Y). Berikut adalah model *Neural Network* arsitektur 2-10-1:

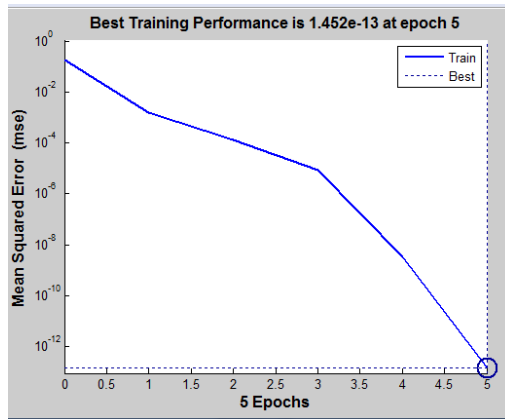


Gambar 4. 37 Model *Neural Network* Arsitektur 2-10-1



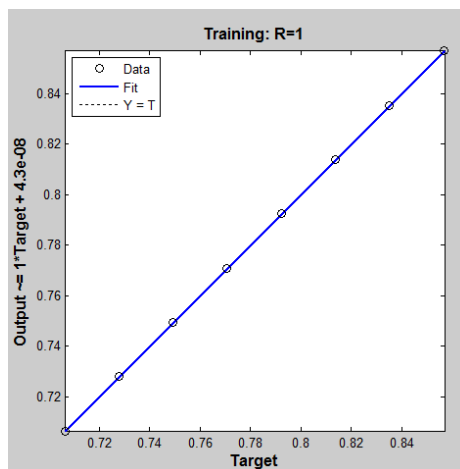
Gambar 4. 38 Hasil Pelatihan Jaringan dengan Arsitektur 2-10-1

Gambar 4.38 menunjukkan bahwa *epoch* yang terjadi memiliki 5 iterasi dan berlangsung selama 4 detik. Berikutnya akan ditampilkan grafik pada Gambar 4.39.



Gambar 4. 39 Performa Data Pelatihan dengan Arsitektur 2-10-1

Metode pelatihan yang terbaik digambarkan secara grafik pada Gambar 4. 39, yang menghasilkan nilai MSE sebesar $1,452 \times 10^{-13}$ yang dicapai pada *epoch* ke 5.



Gambar 4. 40 Tampilan *Regression Plot* dengan Arsitektur 2-10-1

Tampilan plot regresi ditunjukkan pada Gambar 4.40 merupakan tampilan *Regression Plot*. Nilai korelasi untuk plot regresi adalah 1. Mengingat nilainya sama dengan 1, hal ini menunjukkan adanya hubungan yang kuat antara nilai output dengan nilai target.

Dari perhitungan menggunakan *software* MATLAB didapatkan nilai *output* pada 10 model tersaji pada Tabel 4.4.

Tabel 4. 4 Nilai *Output* Data Pelatihan

Target (Y)	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	Model 9	Model 10
0,706 3142 97	0,706 3854 6	0,706 3209 13	0,706 3161 67	0,706 3142 98	0,706 3143 65	0,70 631 4	0,70 631 4	0,70 631 4	0,70 631 4	0,70 631 4
0,728 0369 27	0,728 2504 07	0,728 0294 38	0,728 0325 35	0,728 0369 28	0,728 0369 23	0,72 803 7	0,72 803 7	0,72 803 7	0,72 803 7	0,72 803 7
0,749 3669 98	0,749 0157 84	0,749 3488 54	0,749 3911 97	0,749 3669 99	0,749 3673 6	0,74 936 7	0,74 936 7	0,74 936 7	0,74 936 7	0,74 936 7
0,770 6185 57	0,770 2786 73	0,770 6434 5	0,770 5708 71	0,770 6185 57	0,770 6193 24	0,77 061 9	0,77 061 9	0,77 061 8	0,77 061 9	0,77 061 9
0,792 4341 82	0,792 6427 67	0,792 4563 5	0,792 4825 55	0,792 4341 82	0,792 4331 73	0,79 243 4	0,79 243 4	0,79 243 4	0,79 243 4	0,79 243 4
0,813 8686 81	0,814 1071 59	0,813 7972 33	0,813 8420 72	0,813 8686 81	0,813 8681 72	0,81 386 9	0,81 386 9	0,81 386 9	0,81 386 9	0,81 386 9
0,835 1880 81	0,835 4871 71	0,835 2504 29	0,835 1938 44	0,835 1880 8	0,835 1879 23	0,83 518 8	0,83 518 8	0,83 518 8	0,83 518 8	0,83 518 7
0,856 9907 47	0,856 6443 02	0,856 9704 45	0,856 98857 3	0,856 99074 7	0,856 99063 4	0,85 699 1	0,85 699 1	0,85 699 1	0,85 699 1	0,85 699 1

Berikut merupakan nilai *error* MSE pada pelatihan jaringan untuk kesepuluh model:

Tabel 4. 5 Nilai *Error MSE* untuk 10 Model Pelatihan

Model	<i>Error</i> MSE
1	$7,49211 \times 10^{-8}$
2	$1,36805 \times 10^{-9}$
3	$7,46032 \times 10^{-10}$
4	$4,74081 \times 10^{-19}$
5	$2,54655 \times 10^{-13}$
6	$1,81392 \times 10^{-14}$
7	$5,34994 \times 10^{-21}$
8	$8,33548 \times 10^{-16}$
9	$6,31958 \times 10^{-19}$
10	$1,45205 \times 10^{-13}$

E. Pengujian Jaringan

Pada proses pengujian ini akan digunakan fungsi aktivasi dan parameter yang didapat dari proses pelatihan pada model sebelumnya yaitu model pertama sampai model kesepuluh. Data pengujian yang akan digunakan adalah data jumlah penduduk Kota Semarang pada tahun 2018-2021. Hasil simulasi pengujian pada 10 model tersaji pada Tabel 4. 6.

Tabel 4. 6 Tabel Nilai *Output* Data Pengujian

Ta hu n	Tar get	Mo del 1	Mo del 2	Mo del 3	Mo del 4	Mo del 5	Mo del 6	Mo del 7	Mo del 8	Mo del 9	Mo del 10
20 18	0,8 786 6	0,8 760 57	0,8 775 77	0,8 700 21	0,8 770 75	0,8 755 97	0,8 956 85	0,8 688 95	0,7 924 7	0,8 485 03	0,8 595 61
20 19	0,9	0,8 945 46	0,8 967 13	0,8 752 39	0,8 873 17	0,8 862 86	0,9 490 07	0,8 746 53	0,5 962 12	0,8 281 89	0,8 599 22
20 20	0,7 775 93	0,7 908 93	0,7 812 08	0,7 598 02	0,6 412 13	0,7 608 1	0,8 912 36	0,6 540 74	0,7 965 47	0,8 341 26	0,8 000 38
20 21	0,7 799 1	0,7 929 88	0,7 834 52	0,7 623 74	0,6 455 39	0,7 633 11	0,8 902 24	0,6 609 03	0,7 983 4	0,8 216 58	0,8 009 4

Berikut merupakan nilai *error* MSE untuk 10 model yang tersaji pada Tabel 4. 7.

Tabel 4. 7 Nilai *Error MSE* untuk 10 Model Pengujian

Model ke-	<i>Error</i> MSE
1	$9,61107 \times 10^{-5}$
2	$9,39749 \times 10^{-6}$
3	$3,27957 \times 10^{-4}$
4	$9,2046 \times 10^{-3}$
5	$1,8867 \times 10^{-4}$
6	$6,943792 \times 10^{-3}$
7	$7,539431 \times 10^{-3}$
8	$2,5103776 \times 10^{-2}$
9	$2,751242 \times 10^{-3}$
10	$7,29262 \times 10^{-4}$

Berdasarkan Tabel 4.7 diperoleh model arsitektur terbaik yaitu model 2 dengan arsitektur 2-2-1 yang artinya model dengan 2 *neuron* pada lapisan *input*, 2 *neuron* pada lapisan tersembunyi, dan 1 *neuron* pada lapisan *output*. Model tersebut merupakan model terbaik karena mempunyai nilai MSE terkecil dibanding dengan 9 model yang lain yaitu sebesar $9,39749 \times 10^{-6}$.

Tabel 4. 8 Hasil Nilai *Output* Data Model Arsitektur Terbaik

Tahun	Nilai <i>Output</i>
2018	0,877577
2019	0,896713
2020	0,781208
2021	0,783452

Pada Tabel 4. 8 dapat dilihat nilai output data pengujian jumlah penduduk Kota Semarang yang merupakan hasil dari model 2-2-1 yang memiliki nilai MSE terkecil seperti yang tersaji pada Tabel 4.7. Hasil tersebut masih dalam bentuk normalisasi. Maka dari itu data akan didenormalisasikan untuk mengembalikan data ke dalam bentuk angka desimal yang sebenarnya.

Nilai minimal dan nilai maksimal yang digunakan sama dengan pada saat melakukan proses normalisasi data jumlah penduduk yang mana nilai minimalnya adalah sebesar 764.587 dan nilai maksimalnya adalah sebesar 1.814.110. Dengan menggunakan persamaan rumus denormalisasi (2.18) maka didapatkan:

$$X_{2018} = \frac{(0,877577 - 0,1)(1.814.110 - 764.587)}{0,8}$$

$$+ 764.587 = 1.784.693$$

$$X_{2019} = \frac{(0,896713 - 0,1)(1.814.110 - 764.587)}{0,8}$$

$$+ 764.587 = 1.809.798$$

$$X_{2020} = \frac{(0,781208 - 0,1)(1.814.110 - 764.587)}{0,8}$$

$$+ 764.587 = 1.658.267$$

$$X_{2021} = \frac{(0,783452 - 0,1)(1.814.110 - 764.587)}{0,8}$$

$$+ 764.587 = 1.661.210$$

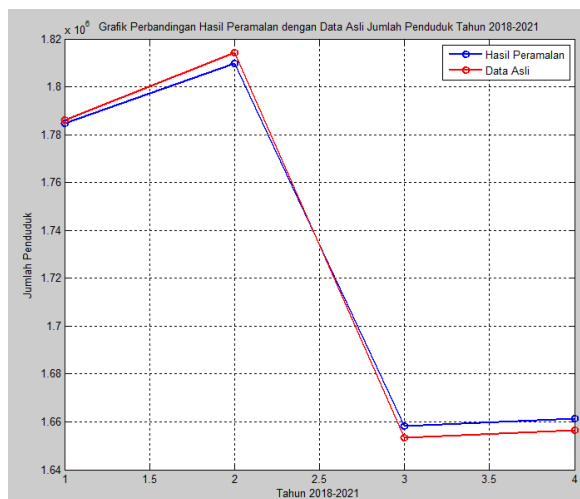
Hasil denormalisasi *output Neural Network* merupakan hasil peramalan jumlah penduduk tahun 2018-2021.

Berikut tabel hasil peramalan jumlah penduduk tahun 2018-2021:

Tabel 4. 9 Hasil Peramalan Data Jumlah Penduduk
Tahun 2018-2021

Tahun	Jumlah Penduduk
2018	1.784.693
2019	1.809.798
2020	1.658.267
2021	1.661.210

Untuk melihat seberapa mirip nilai hasil peramalan jumlah penduduk dengan nilai data asli jumlah penduduk Kota Semarang dapat dilihat dari Gambar 4. 41 di bawah ini:



Gambar 4. 41 Grafik Perbandingan Hasil Peramalan dengan Data Asli

Dari Gambar 4. 41 dapat dilihat bahwa garis warna biru yang menunjukkan data hasil peramalan garisnya berhimpit dengan garis warna merah yang menunjukkan data asli jumlah penduduk. Dengan demikian dapat dikatakan bahwa hasil peramalan jumlah penduduk Kota Semarang pada tahun 2018-2021 mendekati nilai sebenarnya.

Setelah proses pengujian pada data tahun 2018-2021 dilakukan, akan dilakukan proses evaluasi terhadap data asli yaitu data jumlah penduduk Kota Semarang tahun 2018-2021 yang dapat pada *website* BPS Kota Semarang. Hal ini dilakukan untuk mengukur kemampuan model dalam melakukan peramalan. Untuk mengukur keakuratannya akan dipakai metode evaluasi MAPE.

Pada tabel di bawah ini didapat nilai MAPE hasil peramalan dengan menggunakan *Neural Network* metode *Backpropagation*.

Tabel 4. 10 Hasil Nilai MAPE

Tahun	Jumlah Penduduk		MAPE (%)
	Hasil Peramalan	Data Asli	
2018	1784693	1786114	0,318232767
2019	1809798	1814110	0,950769248
2020	1658267	1653524	1,147367683
2021	1661210	1656564	1,121840146
Nilai MAPE rata-rata			0,884552461

Berdasarkan nilai MAPE tersebut dapat dikatakan bahwa hasil peramalan sangat akurat karena nilai MAPE rata-rata $< 10\%$.

F. Hasil Peramalan Jumlah Penduduk Kota Semarang Tahun 2022-2025

Setelah memperoleh model arsitektur terbaik yaitu model 2, kemudian dilakukan peramalan jumlah penduduk Kota Semarang untuk tahun 2022-2025.

Berikut hasil peramalan jumlah penduduk Kota Semarang pada tahun 2022- 2025:

Tabel 4. 11 Hasil Peramalan Jumlah Penduduk

Tahun	Hasil Peramalan
2022	1.863.121
2023	1.878.634
2024	1.891.865
2025	1.902.947

Berdasarkan Tabel 4. 11 dapat kita lihat bahwa jumlah penduduk kota semarang selalu naik dari tahun 2022-2025. Hal ini sesuai dengan proyeksi yang sudah dilakukan oleh Badan Pusat Statistik (BPS) yang menunjukkan bahwa jumlah penduduk Kota Semarang dari tahun 2022-2025 selalu mengalami kenaikan.

Diketahui data aktual jumlah penduduk kota Semarang tahun 2022 adalah 1.658.084 jiwa. Hasil peramalan terlihat lebih besar daripada data aktualnya, dikarenakan ada faktor lain yang mempengaruhi data aktual, diantaranya seperti kematian, kelahiran, migrasi, ataupun faktor lainnya.

BAB V

KESIMPULAN DAN SARAN

A. KESIMPULAN

1. Dari penelitian ini didapatkan model arsitektur terbaik untuk peramalan jumlah penduduk di Kota Semarang dengan pendekatan *Neural Network* menggunakan metode *Backpropagation* adalah pada model 2 dengan 2 *neuron* pada lapisan *input*, 2 *neuron* pada lapisan tersembunyi, dan 1 *neuron* pada lapisan *output*. Dibuktikan dengan dimilikinya nilai MSE terkecil dibanding dengan 9 model yang lain yaitu sebesar $9,39749 \times 10^{-6}$.
2. Pada penelitian ini didapatkan hasil peramalan jumlah penduduk di Kota Semarang dengan pendekatan *Backpropagation Neural Network* pada tahun 2018-2021 memberikan hasil yang mirip dengan data aslinya. Menunjukkan bahwa model ini akurat digunakan karena nilai MAPE < 10%. Dari model yang diperoleh menghasilkan peramalan jumlah penduduk Kota Semarang pada tahun 2022-2025 yaitu 1.863.121 jiwa, 1.878.634 jiwa, 1.891.865 jiwa, dan 1.902.947 jiwa. Dari hasil

tersebut dapat kita lihat bahwa jumlah penduduk Kota Semarang semakin naik setiap tahun sesuai pada proyeksi dari BPS.

B. SARAN

Saran dari penulis adalah sebagai berikut:

1. Diharapkan adanya penelitian lebih lanjut dengan penambahan jumlah data dan variabel input lain yang dapat berdampak pada jumlah penduduk.
2. Diharapkan penelitian lebih lanjut dengan menggunakan variasi parameter seperti jumlah *epoch*, jenis fungsi pelatihan, jumlah layer tersembunyi supaya mendapatkan hasil yang lebih optimal sehingga didapatkan nilai *error* yang lebih kecil dari penelitian ini.

DAFTAR PUSTAKA

- Abiodun, Oludare Isaac, dkk. 2019. *Comprehensive review of artificial neural network applications to pattern recognition*. IEEE Access 7 : 158820-158846.
- Al-Shawwa, M. O., & Abu-Naser, S. S. 2019. *Predicting Birth Weight Using Artificial Neural Network*. International Journal of academic Health and Medical Research, 3(1): 9- 14
- Amri, Fauzul.2015. *Jaringan Syaraf Tiruan Untuk Memprediksi Peringkat Akreditasi Program Studi Perguruan Tinggi*. Jurnal Sains dan Informatika, 1(1): 37-43
- Ananda, R., & Fadhli, M. 2018. *Statistik Pendidikan (Teori dan Praktik dalam Pendidikan)*. Medan: Cv. Widya Puspita.Publishing.
- Anitescu, C., Atroshchenko, E., Alajlan, N., & Rabczuk, T. 2019. *Artificial neural network methods for the solution of second order boundary value problems*. Computers, Materials and Continua, 59(1): 345-359.
- Anwar, Saifudin. 2014. *Metode Penelitian*. Yogyakarta: Pustaka Pelajar.
- Arikunto, S. 2010. *Prosedur Penelitian Suatu Pendekatan Praktek (Edisi Revisi)*. Jakarta: Rineka Cipta.
- Arofah, Suci F.L., Wasono, R., & Arum, Prizka R. 2020. *Peramalan Harga Beras Di Provinsi Jawa Tengah Menggunakan Metode Backpropagation Neural Network*

Dengan Optimasi Conjugate Gradient Beale-Powell Restars. Skripsi. Semarang: Universitas Muhammadiyah Semarang

Awodele, O., & Jegede, O. 2009. *Neural networks and its application in engineering*. Sci IT, 83-95.

Badan Pusat Statistik. 2019. *Profil Kependudukan Kota Semarang 2019*. BPS: Kota Semarang

Badan Pusat Statistik. 2020. *Provinsi Jawa Tengah Dalam Angka 2020*. BPS: Jawa Tengah

Badan Pusat Statistik. 2021. *Provinsi Jawa Tengah Dalam Angka 2021*. BPS: Jawa Tengah

Badrul, Anwar. 2011. *Penerapan Algoritma Jaringan Syaraf Tiruan Backpropagation Dalam Memprediksi Tingkat Suku Bunga Bank*. Jurnal SAINTIKOM. 10(2).

Bappeda. 2017. *Rencana Pembangunan Jangka Menengah Daerah Kota Semarang Tahun 2016 - 2021*. Bappeda: Kota Semarang

Farisyah, A. 2019. *Aplikasi Metode Eksponensial Ganda Brown Dalam Peramalan Jumlah Penduduk Berdasarkan Jenis Kelamin Di Kabupaten Majene*. Doctoral dissertation, UNIVERSITAS NEGERI MAKASSAR.

Gurianto, R. N., Purnamasari, I., & Yuniarti, D. 2017. *Peramalan Jumlah Penduduk Kota Samarinda Dengan Menggunakan Metode Pemulusan Eksponensial Ganda dan Tripel Dari Brown*. Jurnal Eksponensial, 7(1), 23-32.

Hardati, Puji. 2013. *Pertumbuhan Penduduk dan Struktur Lapangan Pekerjaan di Jawa Tengah*. Forum Ilmu Sosial. 40(2).

- Heaton, Jeff. 2008. *Introduction of Neural Networks with Java 2nd Edition*. Heaton Research.
- Hutasuhut, A. H., Anggraeni, W., & Tyasnurita, R. 2014. *Pembuatan Aplikasi Pendukung Keputusan untuk Peramalan Persediaan Bahan Baku Produksi Plastik Blowing dan Inject Menggunakan Metode ARIMA (Autoregressive Integrated Moving Average) di CV. Asia*. Jurnal Teknik ITS, 3(2): A169-A174
- Khalil, A. J., Barhoom, A. M., Abu-Nasser, B. S., Musleh, M. M., & Abu-Naser, S. S. 2019. *Energy Efficiency Prediction using Artificial Neural Network*. International Journal of Academic Pedagogical Research (IJAPR), 3(9).
- Kholis, Ikhwanul. 2015. *Analisis Variasi Parameter Backpropagation Artificial Neural Network Terhadap Pengenalan Pola Data Iris*. Jurnal Teknik dan Ilmu Komputer Ukrida. 1-10.
- Kusumadewi, Sri dan Purnomo, Hari. 2010. *Aplikasi Logika Fuzzy Untuk Pendukung Keputusan*. Yogyakarta: Graha Ilmu.
- Mantra, Ida Bagus. 2003. *Demografi Umum Edisi ke 2*. Yogyakarta: Pustaka Pelajar.
- Moreno, J. J. M., Pol, A. P., Abad, A. S., & Blasco, B. C. 2013. *Using the R-MAPE index as a resistant measure of forecast accuracy*. Psicothema, 25(4), 500-506.
- Monika, D., dkk. *Model Jaringan Syaraf Tiruan dalam Memprediksi Ketersediaan Cabai Berdasarkan Provinsi*. TEKNIKA, 8(1), 17-24.
- Montgomery, D. C., Jennings, C. L., & Kulahci, M. 2015. *Introduction to time series analysis and forecasting*. John Wiley & Sons.

- Purwanto, S. D. (2016). *Implementasi Jaringan Syaraf Tiruan Backpropagation Sebagai Estimasi Laju Tingkat Pengangguran Terbuka Pada Provinsi Jawa Timur*. SEMNASTEKNOMEDIA ONLINE, 4(1), 4-9.
- Puspitaningrum, D. 2006. *Pengantar Jaringan Syaraf Tiruan*. Yogyakarta: Penerbit ANDI.
- Prasetyo, Eko. 2012. *Data Mining Konsep dan Aplikasi Menggunakan Matlab*. Yogyakarta: Penerbit ANDI.
- Rahman, Faja Alya. 2012. *Peramalan Beban Listrik Jangka Pendek Menggunakan Jaringan Syaraf Tiruan*. Skripsi. Jakarta. FT UI.
- Riyandhi, Risky. 2017. *Penerapan artificial Neural Network (ANN) untuk Peramalan Inflasi di Indonesia*. Skripsi. Surabaya: Institut Teknologi Sepuluh Nopember
- Rufiyanti, Dwi Erfi. 2015. *Implementasi Jaringan Syaraf Tiruan Backpropagation dengan Input Model ARIMA untuk Peramalan Harga Saham*. Skripsi. Semarang: Universitas Negeri Semarang
- Saragih, Arini Miranda Aprilyn. 2018. *Peramalan Jumlah Penduduk Kota Medan Tahun 2016– 2019*. Skripsi. Medan: Universitas Sumatera Barat.
- Sarja, N. L. A. K. Y., & Wirawan, I. W. W. 2014. *Peramalan Permintaan Produk Perak Menggunakan Metode Simple Moving Average Dan Exponential Smoothing*. Jurnal Sistem dan Informatika. 9(1): 97-106
- Siang, Jong Jek. 2009. *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB*. Yogyakarta: ANDI.

- Sudarsono, Aji. 2016. *Jaringan Syaraf Tiruan Untuk Memprediksi Laju Pertumbuhan Penduduk Menggunakan Metode Backpropagation (Studi Kasus di Kota Bengkulu)*. Jurnal Media Infotama. 12 (1):61-69
- Sukerti, N. K. 2015. *Peramalan Deret Waktu Menggunakan S-Curve dan Quadratic Trend Model*. Proceedings Konferensi Nasional Sistem dan Informatika (KNS&I).
- Wahidmurni. 2017. *Pemaparan Metode Penelitian Kuantitatif*. Malang: UIN Maulana Malik Ibrahim.
- Wahyuningrum, S. R. 2020. *Statistika Pendidikan (Konsep Data dan Peluang)*. Jakad Media
- Warsito, Budi. 2009. *Kapita Selekta Statistika Neural Network*. Semarang: UNDIP Semarang
- Wijaya, Anggi Hadi. 2019. *Artificial Neural Network Untuk Memprediksi Beban Listrik Dengan Menggunakan Metode Backpropagation*. Jurnal CoreIT. 5(2): 61-70.
- Yanti, N. 2011. *Penerapan Metode Neural Network dengan Struktur Backpropagation Untuk Prediksi Stok Obat di Apotek (Studi Kasus: Apotek ABC)*. In Seminar Nasional Aplikasi Teknologi Informasi (SNATI).

LAMPIRAN

Lampiran 1 Nilai *Error* pada Pelatihan Jaringan

N o.	<i>Err or Mo del 1</i>	<i>Erro r Mo del 2</i>	<i>Erro r Mo del 3</i>	<i>Erro r Mod el 4</i>	<i>Erro r Mo del 5</i>	<i>Err or Mo del 6</i>	<i>Erro r Mod el 7</i>	<i>Erro r Mod el 8</i>	<i>Erro r Mod el 9</i>	<i>Erro r Mod el 10</i>
1	7,1 162 8 × 10 ⁻⁵	6,61 604 × 10 ⁻⁶	1,86 989 × 10 ⁻⁶	1,24 596 × 10 ⁻⁹	6,80 121 × 10 ⁻⁸	2,4 41 × 10 ⁻⁸	- 8,00 978 × 10 ⁻¹¹	1,46 305 × 10 ⁻⁹	7,36 974 × 10 ⁻¹¹	- 2,99 825 × 10 ⁻⁸
2	0,0 002 134 8	- 7,48 897 × 10 ⁻⁶	- 4,39 263 × 10 ⁻⁶	6,81 122 × 10 ⁻¹⁰	- 4,62 749 × 10 ⁻⁹	8,0 910 6× 10 ⁻⁸	- 1,07 842 × 10 ⁻¹⁰	- 1,29 215 × 10 ⁻⁸	- 2,41 657 × 10 ⁻¹⁰	3,70 503 × 10 ⁻⁸
3	- 0,0 003 512 14	- 1,81 44× 10 ⁻⁵	2,41 989 × 10 ⁻⁵	9,01 17× 10 ⁻¹⁰	3,61 578 × 10 ⁻⁷	3,9 988 3× 10 ⁻⁸	1,64 986 × 10 ⁻¹¹	- 2,39 285 × 10 ⁻⁹	2,76 472 × 10 ⁻¹⁰	- 4,87 106 × 10 ⁻⁸
4	- 0,0 003 398 84	2,48 927 × 10 ⁻⁵	- 4,76 861 × 10 ⁻⁵	- 1,54 838 × 10 ⁻¹⁰	7,66 47× 10 ⁻⁷	1,0 695 2× 10 ⁻⁷	- 1,06 031 × 10 ⁻¹⁰	- 7,21 81× 10 ⁻⁸	- 4,91 818 × 10 ⁻¹⁰	3,47 696 × 10 ⁻⁸
5	0,0 002 085 85	2,21 682 × 10 ⁻⁵	4,83 728 × 10 ⁻⁵	2,23 585 × 10 ⁻¹⁰	- 1,00 858 × 10 ⁻⁶	1,1 453 8× 10 ⁻⁷	7,72 243 × 10 ⁻¹¹	- 3,55 066 × 10 ⁻⁸	5,54 57× 10 ⁻¹⁰	1,47 495 × 10 ⁻⁸
6	0,0 002 384 78	- 7,14 482 × 10 ⁻⁵	- 2,66 09× 10 ⁻⁵	- 6,51 672 × 10 ⁻¹⁰	- 5,09 397 × 10 ⁻⁷	1,5 971 7× 10 ⁻⁷	- 7,37 176 × 10 ⁻¹¹	- 4,75 288 × 10 ⁻⁹	- 1,20 172 × 10 ⁻⁹	3,73 559 × 10 ⁻⁷

N o.	<i>Err or Mo del 1</i>	<i>Erro r Mo del 2</i>	<i>Erro r Mo del 3</i>	<i>Erro r Mod el 4</i>	<i>Erro r Mo del 5</i>	<i>Err or Mo del 6</i>	<i>Erro r Mod el 7</i>	<i>Erro r Mod el 8</i>	<i>Erro r Mod el 9</i>	<i>Erro r Mo del 10</i>
7	0,0 002 990 9	6,23 48× 10 ⁻⁵	5,76 359 × 10 ⁻⁶	- 3,56 359 × 10 ⁻¹⁰	- 1,57 518 × 10 ⁻⁷	1,9 531 5× 10 ⁻⁷	1,66 435 × 10 ⁻¹¹	3,83 962 × 10 ⁻¹⁰	1,40 687 × 10 ⁻⁹	- 9,04 748 × 10 ⁻⁷
8	- 0,0 003 464 45	- 2,03 02× 10 ⁻⁵	- 2,17 46× 10 ⁻⁶	- 5,81 867 × 10 ⁻¹⁰	- 1,13 314 × 10 ⁻⁷	2,1 944 7× 10 ⁻⁷	- 3,95 516 × 10 ⁻¹¹	7,58 36× 10 ⁻¹¹	- 9,70 852 × 10 ⁻¹⁰	4,44 357 × 10 ⁻⁷
<i>Er ro r M S E</i>	7,4 9× 10 ⁻⁸	1,37 × 10 ⁻⁹	7,46 × 10 ⁻¹⁰	4,74 × 10 ⁻¹⁹	2,55 × 10 ⁻¹³	1,8 1× 10 ⁻¹⁴	5,35 × 10 ⁻²¹	8,34 × 10 ⁻¹⁶	6,32 × 10 ⁻¹⁹	1,45 × 10 ⁻¹³

Lampiran 2 Nilai Error pada Pengujian Jaringan

No	Erro r Mod el 1	Erro r Mod el 2	Erro r Mod el 3	Erro r Mod el 4	Erro r Mod el 5	Err or Mo del 6	Erro r Mod el 7	Erro r Mod el 8	Erro r Mod el 9	Erro r Mod el 10
1	- 0,00 260 327	- 0,00 108 322 1	- 0,00 863 938 6	- 0,00 158 490 4	- 0,00 306 276 1	0,01 702 453 4	- 0,00 976 492 3	- 0,08 619 034 8	- 0,03 015 675 1	- 0,01 909 901 8
2	- 0,00 545 361 5	- 0,00 328 705 1	- 0,02 476 147 8	- 0,01 268 308 9	- 0,01 371 443 8	0,04 900 730 1	- 0,02 534 740 8	- 0,30 378 844 4	- 0,07 181 079 5	- 0,04 007 825 2
3	0,01 330 004 6	0,00 361 517 6	- 0,01 779 121 6	- 0,13 637 987 7	- 0,01 678 305 6	0,11 364 250 9	- 0,12 351 934 2	0,01 895 409 6	0,05 653 246 2	0,02 244 513 5
4	0,01 307 794	0,00 354 152 5	- 0,01 753 658 3	- 0,13 437 097 1	- 0,01 659 942	0,11 031 317 9	- 0,11 900 777 9	0,01 842 967 9	0,04 174 719 4	0,02 102 915 8
Nil ai M SE	9,61 × 10 ⁻⁵	9,40 × 10 ⁻⁶	0,00 032 795 7	0,00 920 46	0,00 018 867	0,00 694 379 2	0,00 753 943 1	0,02 510 377 6	0,00 275 124 2	0,00 072 926 2

Lampiran 3 Syntax Program *Backpropagation Neural Network*

Pelatihan Jaringan

```

%membaca input pelatihan
input_pelatihan=xlsread('data
penduduk.xlsx',1,'C23:J24');
%membaca target pelatihan
target_pelatihan=xlsread('data
penduduk.xlsx',1,'C25:J25');
%menetapkan parameter neural network
jumlah_neuron1=2;
fungsi_aktivasi1='logsig';
fungsi_aktivasi2='logsig';
fungsi_pelatihan='trainlm';
%membangun arsitektur neural network
backpropagation
rng('default')
jaringan =
newff(minmax(input_pelatihan),[jumlah_neuron1
1],...

{fungsi_aktivasi1,fungsi_aktivasi2},fungsi_pelat
ihan);
%melakukan pelatihan jaringan
jaringan=train(jaringan,input_pelatihan,target_p
elatihan);
% membaca hasil pelatihan
hasil_pelatihan=sim(jaringan,input_pelatihan);
%menghitung nilai MSE
nilai_error=hasil_pelatihan-target_pelatihan;
error_MSE=(1/8)*sum(nilai_error.^2);
% menyimpan arsitektur Neural Network hasil
pelatihan
save jaringan jaringan

```


Model 2

Variables - nilai_error

hasil_pengujian × hasil_pelatihan × error_MSE × nilai_error ×

nilai_error <1x8 double>

	1	2	3	4	5	6	7	8
1	6.6160e-06	-7.4890e-06	-1.8144e-05	2.4893e-05	2.2168e-05	-7.1448e-05	6.2348e-05	-2.0302e-05

Model 3

Variables - nilai_error

hasil_pelatihan × error_MSE × nilai_error × hasil_pengujian × nilai_error_uji × error_MSE_uji ×

nilai_error <1x8 double>

	1	2	3	4	5	6	7	8
1	1.8699e-06	-4.3926e-06	2.4199e-05	-4.7686e-05	4.8373e-05	-2.6609e-05	5.7636e-06	-2.1746e-06
2								

Model 4

Variables - nilai_error

hasil_pelatihan × error_MSE × nilai_error × hasil_pengujian × nilai_error_uji × error_MSE_uji ×

nilai_error <1x8 double>

	1	2	3	4	5	6	7	8
1	1.2460e-09	6.8112e-10	9.0117e-10	-1.5484e-10	2.2359e-10	-6.5167e-10	-3.5636e-10	-5.8187e-10
2								

Model 5

Variables - nilai_error

hasil_pelatihan × error_MSE × nilai_error × hasil_pengujian × nilai_error_uji × error_MSE_uji ×

nilai_error <1x8 double>

	1	2	3	4	5	6	7	8
1	6.8012e-08	-4.6275e-09	3.6158e-07	7.6647e-07	-1.0086e-06	-5.0940e-07	-1.5752e-07	-1.1331e-07

Model 6

Variables - nilai_error

hasil_pelatihan × error_MSE × nilai_error × hasil_pengujian × nilai_error_uji × error_MSE_uji ×

nilai_error <1x8 double>

	1	2	3	4	5	6	7	8
1	2.4410e-08	8.0911e-08	3.9988e-08	1.0695e-07	1.1454e-07	1.5972e-07	1.9531e-07	2.1945e-07
2								

Model 7

Variables - nilai_error

hasil_pelatihan × error_MSE × nilai_error × hasil_pengujian × nilai_error_uji × error_MSE_uji ×

nilai_error <1x8 double>

	1	2	3	4	5	6	7	8
1	-8.0098e-11	-1.0784e-10	1.6499e-11	-1.0603e-10	7.7224e-11	-7.3718e-11	1.6643e-11	-3.9552e-11

Model 8

Variables - nilai_error

hasil_pelatihan × error_MSE × nilai_error × hasil_pengujian × nilai_error_uji × error_MSE_uji ×

nilai_error <1x8 double>

	1	2	3	4	5	6	7	8
1	1.4630e-09	-1.2922e-08	-2.3928e-09	-7.2181e-08	-3.5507e-08	-4.7529e-09	3.8396e-10	7.5836e-11
2								

Model 9

Variables - nilai_error

hasil_pelatihan × error_MSE × nilai_error × hasil_pengujian × nilai_error_uji × error_MSE_uji ×

nilai_error <1x8 double>

	1	2	3	4	5	6	7	8
1	7.3697e-11	-2.4166e-10	2.7647e-10	-4.9182e-10	5.5457e-10	-1.2017e-09	1.4069e-09	-9.7085e-10
2								

Model 10

Variables - nilai_error

hasil_pelatihan × error_MSE × nilai_error × hasil_pengujian × nilai_error_uji × error_MSE_uji ×

nilai_error <1x8 double>

	1	2	3	4	5	6	7	8
1	-2.9982e-08	3.7050e-08	-4.8711e-08	3.4770e-08	1.4750e-08	3.7356e-07	-9.0475e-07	4.4436e-07
2								

3. Nilai MSE

Model 1

hasil_pelatihan × error_MSE ×

error_MSE <1x1 double>

	1	2
1	7.4921e-08	
2		

Model 2

hasil_pelatihan × nilai_error ×

error_MSE <1x1 double>

	1	2	3
1	1.3681e-09		
2			

Model 3

Variables - error_MSE

hasil_pelatihan × error_MSE

error_MSE <1x1 double>

	1	2
1	7.4603e-10	
2		

Model 4

	1	2
1	4.7408e-19	
2		

Model 5

	1	2
1	2.5465e-13	
2		

Model 6

	1	2
1	1.8139e-14	
2		

Model 7

	1	2
1	5.3499e-21	
2		

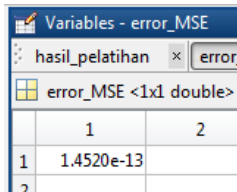
Model 8

	1	2
1	8.3355e-16	
2		

Model 9

	1	2
1	6.3196e-19	
2		

Model 10



	1	2
1	1.4520e-13	
2		

Pengujian Jaringan

```

%masuk ke pengujian
%menyiapkan data pengujian
input_pengujian=xlsread('data
penduduk.xlsx',1,'K23:N24');
target_pengujian=xlsread('data
penduduk.xlsx',1,'K25:N25');
%memanggil arsitektur Neural Network hasil
pelatihan
load jaringan
% membaca hasil pengujian
hasil_pengujian=sim(jaringan,input_pengujian);
% menghitung nilai error MSE
nilai_error_uji=hasil_pengujian-
target_pengujian;
error_MSE_uji=(1/4)*sum(nilai_error_uji.^2);
% menampilkan grafik hasil pelatihan
figure
plot(denor_hasil_pengujian,'bo-','LineWidth',2)
hold on
plot(denor_target_pengujian,'ro-','LineWidth',2)
grid on
title(['Grafik Perbandingan Hasil Peramalan
dengan Data Asli Jumlah Penduduk Tahun 2018-
2021'])
xlabel('Tahun 2018-2021')
ylabel('Jumlah Penduduk')
legend('Hasil Peramalan','Data Asli')

```

hold off

Output Pengujian:

1. Hasil pengujian Model 1

hasil_pengujian <1x4 double>				
	1	2	3	4
	0.8761	0.8945	0.7909	0.7930

Model 2

hasil_pengujian <1x4 double>				
	1	2	3	4
1	0.8776	0.8967	0.7812	0.7835
2				

Model 3

hasil_pengujian <1x4 double>				
	1	2	3	4
1	0.8700	0.8752	0.7598	0.7624
2				

Model 4

hasil_pengujian <1x4 double>				
	1	2	3	4
1	0.8771	0.8873	0.6412	0.6455
2				

Model 5

hasil_pengujian <1x4 double>				
	1	2	3	4
1	0.8756	0.8863	0.7608	0.7633
2				

Model 6

hasil_pengujian <1x4 double>				
	1	2	3	4
1	0.8957	0.9490	0.8912	0.8902
2				

Model 7

hasil_pengujian <1x4 double>				
	1	2	3	4
1	0.8689	0.8747	0.6541	0.6609
2				

Model 8

hasil_pengujian <1x4 double>				
	1	2	3	4
1	0.7925	0.5962	0.7965	0.7983
2				

Model 9

hasil_pengujian <1x4 double>				
	1	2	3	4
1	0.8485	0.8282	0.8341	0.8217
2				

Model 10

hasil_pengujian <1x4 double>				
	1	2	3	4
1	0.8596	0.8599	0.8000	0.8009
2				

2. Nilai *Error* pada Pengujian Jaringan Model 1

nilai_error_uji <1x4 double>				
	1	2	3	4
1	-0.0026	-0.0055	0.0133	0.0131
2				

Model 2

	1	2	3	4
1	-0.0011	-0.0033	0.0036	0.0035
2				

Model 3

	1	2	3	4
1	-0.0086	-0.0248	-0.0178	-0.0175
2				

Model 4

	1	2	3	4
1	-0.0016	-0.0127	-0.1364	-0.1344
2				

Model 5

	1	2	3	4
1	-0.0031	-0.0137	-0.0168	-0.0166
2				

Model 6

	1	2	3	4
1	0.0170	0.0490	0.1136	0.1103
2				

Model 7

	1	2	3	4
1	-0.0098	-0.0253	-0.1235	-0.1190
2				

Model 8

nilai_error_uji <1x4 double>				
	1	2	3	4
1	-0.0862	-0.3038	0.0190	0.0184
2				

Model 9

nilai_error_uji <1x4 double>				
	1	2	3	4
1	-0.0302	-0.0718	0.0565	0.0417
2				

Model 10

nilai_error_uji <1x4 double>				
	1	2	3	4
1	-0.0191	-0.0401	0.0224	0.0210
2				

3. Nilai MSE

Model 1

error_MSE_uji <1x1 double>		
	1	2
1	9.6111e-05	
2		

Model 2

error_MSE_uji <1x1 double>			
	1	2	3
1	9.3975e-06		
2			

Model 3

error_MSE_uji <1x1 double>			
	1	2	
1	3.2796e-04		
2			
3			

Model 4

error_MSE_uji <1x1 double>			
	1	2	
1	0.0092		
2			

Model 5

error_MSE_uji <1x1 double>			
	1	2	
1	1.8867e-04		
2			

Model 6

error_MSE_uji <1x1 double>			
	1	2	
1	0.0069		
2			

Model 7

error_MSE_uji <1x1 double>			
	1	2	
1	0.0075		
2			

Model 8

error_MSE_uji <1x1 double>			
	1	2	
1	0.0251		
2			

Model 9

error_MSE_uji <1x1 double>		
	1	2
1	0.0028	

Model 10

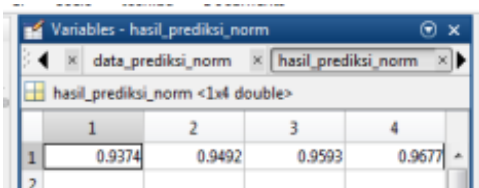
error_MSE_uji <1x1 double>		
	1	2
1	7.2926e-04	

Prediksi

```
% menyiapkan data prediksi normalisasi
data_prediksi_norm=hasil_pengujian(end-3:end);
% melakukan transpose terhadap data hasil
prediksi
data_prediksi_norm=data_prediksi_norm';
%melakukan prediksi
hasil_prediksi_norm=sim(jaringan,data_prediksi_norm);
%melakukan prediksi
hasil_prediksi_norm=sim(jaringan,data_prediksi_norm);
for n = 1:3
    data_prediksi_norm=[data_prediksi_norm(end-
3:end);hasil_prediksi_norm(end) ]

hasil_prediksi_norm=[hasil_prediksi_norm,sim(jar
ingan,data_prediksi_norm) ] ;
end
```

Output Prediksi:



Variables - hasil_prediksi_norm					
		1	2	3	4
1	hasil_prediksi_norm <1x4 double>	0.9374	0.9492	0.9593	0.9677
2					

Lampiran 4 Perhitungan Manual *Backpropagation*

Berikut merupakan perhitungan manual pelatihan jaringan untuk model arsitektur 2-2-1:

Langkah 0:

1. Menentukan nilai bobot pada setiap unit di lapisan input ke setiap unit di lapisan tersembunyi (v_{ji}) dan nilai bias dari lapisan input ke setiap unit di lapisan tersembunyi (v_{j0}).

$$v_{11} = 0,18$$

$$v_{21} = 0,1$$

$$v_{12} = 0,34$$

$$v_{22} = 0,43$$

$$v_{10} = 1$$

$$v_{20} = 0,71$$

2. Menentukan nilai bobot pada setiap unit di lapisan tersembunyi ke setiap unit di lapisan output (w_{kj}) dan nilai bias dari lapisan tersembunyi ke setiap unit di lapisan output (w_{j0}).

$$w_{11} = 0,71$$

$$w_{12} = 0,27$$

$$w_{10} = 0,59$$

Langkah 1: Jika kondisi penghentian belum terpenuhi maka lakukan langkah 2 sampai dengan 9.

Langkah 2: Untuk setiap pasangan pola pelatihan, lakukan langkah ke-3 sampai langkah ke-9.

Fase Propagasi Maju

Langkah 3: Setiap unit *input* ($X_i, i = 1, 2, 3, \dots, n$) menerima sinyal *input* dan meneruskan sinyal tersebut ke setiap unit pada lapisan tersembunyi.

Diketahui: $x_1 = 0,12350706$; $x_2 = 0,1$

Langkah 4: Setiap unit di lapisan tersembunyi ($Z_j, j = 1, 2, 3, \dots, m$) menjumlahkan sinyal- sinyal *input* yang berbobot, yaitu

$$Z_{netj} = v_{j0} + \sum_{i=1}^n x_i v_{ji}$$

$$Z_{net1} = v_{10} + \sum_{i=1}^2 x_i v_{1i} = 1 + (0,12350706 * 0,18) + (0,1 * 0,34) = 1,056231271$$

$$Z_{net2} = v_{20} + \sum_{i=1}^2 x_i v_{2i} = 1 + (0,12350706 * 0,1) + (0,1 * 0,43) = 0,765350706$$

Fungsi aktivasi untuk menghitung sinyal outputnya, yaitu:

$$Z_j = f(Z_{netj}) = \frac{1}{1 + e^{-Z_{netj}}}$$

$$Z_1 = \frac{1}{1+e^{-Z_{net1}}} = \frac{1}{1+e^{-1,056231271}} = 0,741969678$$

$$Z_2 = \frac{1}{1+e^{-Z_{net2}}} = \frac{1}{1+e^{-0,765350706}} = 0,682514301$$

Langkah 5: Setiap unit di lapisan *ouput* (Y_k , $k = 1, 2, 3, \dots, p$) menjumlahkan sinyal *hidden* yang berbobot, yaitu:

$$Y_{netk} = w_{k0} + \sum_{j=1}^m Z_j w_{kj}$$

$$Y_{net} = w_{10} + \sum_{j=1}^m Z_j w_{1j} = 0,59 + (0,741969678 * 0,71) + (0,682514301 * 0,27) = 1,301077332$$

Fungsi aktivasi untuk menghitung sinyal outputnya,yaitu:

$$Y_k = f(Y_{netk}) = \frac{1}{1 + e^{-Y_{netk}}}$$

$$Y_k = \frac{1}{1+e^{-Y_{net}}} = \frac{1}{1+e^{-1,301077332}} = 0,786016241$$

Fase Propagasi Mundur

Langkah 6: Setiap unit di lapisan ouput Y_k menerima pola target t_k untuk menghitung *error* (δ_k), yaitu:

$$\delta_k = (t_k - Y_k) f'(Y_{netk}) = (t_k - Y_k) Y_k (1 - Y_k)$$

Diketahui $t_k = 0,7063143$

$$\begin{aligned}\delta_k &= (t_k - Y_k) Y_k (1 - Y_k) \\ &= (0,7063143 - 0,786016241) * 0,786016241 * \\ &(1 - 0,786016241) = -0,01340545\end{aligned}$$

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

$$\begin{aligned}\Delta w_{kj} &= \alpha \delta_k Z_j \\ \Delta w_{11} &= \alpha \delta_k Z_1 = \\ 0,01 * -0,01340545 * 0,741969678 &= -9,9464 \times 10^{-5} \\ \Delta w_{12} &= \alpha \delta_k Z_2 = \\ 0,01 * -0,01340545 * 0,682514301 &= -9,1494 \times 10^{-5}\end{aligned}$$

Akan dihitung juga koreksi bias yang digunakan untuk memperbaiki nilai bias antara lapisan tersembunyi dan lapisan *output* (w_{k0}), yaitu:

$$\begin{aligned}\Delta w_{k0} &= \alpha \delta_k \\ \Delta w_{10} &= \alpha \delta_k = 0,01 * -0,01340545 = -0,00013405\end{aligned}$$

Langkah 7: menghitung faktor δ unit tersembunyi berdasarkan nilai *error* di setiap unit tersembunyi (Z_j , $j = 1, 2, \dots, m$)

$$\delta_{netj} = \sum_{k=1}^p \delta_k w_{kj}$$

$$\delta_{net1} = \delta_k w_{11} = -0,01340545 * 0,71 = -0,00951787$$

$$\delta_{net2} = \delta_k w_{12} = -0,01340545 * 0,27 = -0,00361947$$

Kemudian mengalikan nilai tersebut dengan fungsi aktivasi untuk menghitung nilai *error* pada lapisan tersembunyi (δ_j), yaitu:

$$\delta_j = \delta_{netj} f'(Z_{netj}) = \delta_{netj} Z_j (1 - Z_j)$$

$$\delta_1 = \delta_{net1} Z_1 (1 - Z_1) = -0,00951787 * 0,741969678 * (1 - 0,741969678) = -0,0018222$$

$$\delta_2 = \delta_{net2} Z_2 (1 - Z_2) = -0,00361947 * 0,682514301 * (1 - 0,682514301) = -0,007843$$

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

$$\Delta v_{ji} = \alpha \delta_j x_i$$

$$\Delta v_{11} = \alpha \delta_1 x_1 = 0,01 * -0,0018222 * 0,12350706 = -2,2505 \times 10^{-6}$$

$$\Delta v_{12} = \alpha \delta_1 x_1 = 0,01 * -0,0018222 * 0,1 = -1,822 \times 10^{-6}$$

$$\Delta v_{21} = \alpha \delta_2 x_1 = 0,01 * -0,007843 * 0,12350706 = -9,6866 \times 10^{-7}$$

$$\Delta v_{22} = \alpha \delta_2 x_2 = 0,01 * -0,007843 * 0,1 = -7,843 \times 10^{-7}$$

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

$$\Delta v_{j0} = \alpha \delta_j$$

$$\Delta v_{10} = \alpha \delta_1 = 0,01 * -0,0018222 = -1,822 \times 10^{-5}$$

$$\Delta v_{20} = \alpha \delta_2 = 0,01 * -0,007843 = -7,843 \times 10^{-6}$$

Fase Perubahan Bobot

Langkah 8: Setiap unit *output* (Y_k , $k = 1, 2, 3, \dots, p$) memperbaiki bobot dan bias, yaitu:

$$W_{kj}(\text{baru}) = W_{kj}(\text{lama}) + \Delta W_{kj}$$

$$W_{11}(\text{baru}) = W_{11}(\text{lama}) + \Delta W_{11} = 0,71 + (-9,9464 \times 10^{-5}) = 0,709900536$$

$$W_{12}(\text{baru}) = W_{12}(\text{lama}) + \Delta W_{12} = 0,27 + (-9,1494 \times 10^{-5}) = 0,269908506$$

$$W_{10}(\text{baru}) = W_{10}(\text{lama}) + \Delta W_{10} = 0,59 + (-0,00013405) = 0,58986595$$

Setiap unit lapisan tersembunyi (Z_j , $j = 1, 2, 3, \dots, m$) memperbaiki bobot dan biasnya, yaitu:

$$V_{ji}(\text{baru}) = V_{ji}(\text{lama}) + \Delta V_{ji}$$

$$V_{11}(\text{baru}) = V_{11}(\text{lama}) + \Delta V_{11} = 0,18 + (-2,2505 \times 10^{-6}) = 0,179997749$$

$$V_{21}(\text{baru}) = V_{21}(\text{lama}) + \Delta V_{21} = 0,1 + (-9,6866 \times 10^{-7}) = 0,099999031$$

$$V_{12}(\text{baru}) = V_{12}(\text{lama}) + \Delta V_{12} = 0,34 + (-1,822 \times 10^{-6}) = 0,339998178$$

$$V_{22}(\text{baru}) = V_{22}(\text{lama}) + \Delta V_{22} = 0,43 + (-7,843 \times 10^{-7}) = 0,429999216$$

$$V_{10}(\text{baru}) = V_{10}(\text{lama}) + \Delta V_{10} = 1 + (-1,822 \times 10^{-5}) = 0,99998178$$

$$V_{20}(\text{baru}) = V_{20}(\text{lama}) + \Delta V_{20} = 0,71 + (-7,843 \times 10^{-6}) = 0,70999216$$

Tahapan di atas dikatakan *epoch* 1 pada data jumlah penduduk tahun 2010 yang mana proses ini belum mencapai *error* yang diinginkan maka dari itu pelatihan jaringan harus dilakukan pengulangan dimulai dari langkah 2 sampai *error* yang dihasilkan sesuai dengan yang diharapkan. Perulangan kedua menggunakan bobot baru yang dihasilkan pada *epoch* 1.

RIWAYAT HIDUP

A. Identitas Diri

Nama : Agustin Absari Wahyu Kuntarini
TTL : Jepara, 15 Agustus 2000
Alamat : Jalan Kuda Rt 04 Rw 07 Wonosari
Ngaliyan Semarang
No. HP : 085875076227
Email : agustinabsariwk1234@gmail.com
Instagram : @agustin_absari

B. Riwayat Pendidikan

1. SD N Wonosari 01 Lulus Tahun 2012
2. SMP N 18 Semarang Lulus Tahun 2015
3. SMA N 6 Semarang Lulus Tahun 2018