# ANALISIS PERBANDINGAN METODE *MULTILAYER PERCEPTRON* DAN *RECURRENT NEURAL NETWORK* DALAM MEMPREDIKSI LAJU PERTUMBUHAN PENDUDUK DI KABUPATEN BREBES

**SKRIPSI**

Diajukan untuk Memenuhi Sebagian Syarat Guna Memperoleh
Gelar Sarjana Matematika
dalam Ilmu Matematika

Oleh: **IZZATUL YAZIDAH**
NIM: 2008046001

**PROGRAM STUDI MATEMATIKA**
**FAKULTAS SAINS DAN TEKNOLOGI**
**UNIVERSITAS ISLAM NEGERI WALISONGO**
**SEMARANG**
**2024**

## PERNYATAAN KEASLIAN

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini :

| | |
|---|---|
| Nama | : Izzatul Yazidah |
| NIM | : 2008046001 |
| Jurusan | : Matematika |

Menyatakan bahwa skripsi yang berjudul :

**ANALISIS PERBANDINGAN METODE *MULTILAYER PERCEPTRON* DAN *RECURRENT NEURAL NETWORK* DALAM MEMPREDIKSI LAJU PERTUMBUHAN PENDUDUK DI KABUPATEN BREBES**

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

Semarang, 27 Juni 2024

Yang membuat pernyataan,


Izzatul Yazidah

NIM : 2008046001

# PENGESAHAN

KEMENTERIAN AGAMA R.I.
UNIVERSITAS ISLAM NEGERI WALISONGO
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Prof. Dr. Hamka Ngaliyan Semarang
Telp. 024-7601295 Fax. 7615387

**PENGESAHAN**

Naskah skripsi berikut ini :

Judul : **ANALISIS PERBANDINGAN METODE *MULTILAYER PERCEPTRON* DAN *RECURRENT NEURAL NETWORK* DALAM MEMPREDIKSI LAJU PERTUMBUHAN PENDUDUK DI KABUPATEN BREBES**

Penulis : Izzatul Yazidah
NIM : 2008046001
Jurusan : Matematika

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

Semarang, 28 Juni 2024

DEWAN PENGUJI

| Ketua Sidang, | Sekretaris Sidang |
|---|---|
| **Dinni Rahma Oktaviani, M.Si.** | **Agus Wayan Yulianto, M.Sc.** |
| NIP : 199410092019032017 | NIP : 198907162019031007 |
| Penguji I, | Penguji II, |
| **Zulaikha, M.Si.** | **Eva Khoirun Nisa, S.Si., M.Si.** |
| NIP : 199204092019032027 | NIP : 198701022019032010 |

Pembimbing

**Emy Siswanah, M.Sc.**
NIP : 198702022011012014

**NOTA DINAS**

**NOTA DINAS**

Semarang, 27 Juni 2024

Yth. Ketua Program Studi Matematika
Fakultas Sains dan Teknologi
UIN Walisongo Semarang

Assalamu'alaikum wr.wb.

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

Judul : **ANALISIS PERBANDINGAN METODE *MULTILAYER PERCEPTRON* DAN *RECURRENT NEURAL NETWORK* DALAM MEMPREDIKSI LAJU PERTUMBUHAN PENDUDUK DI KABUPATEN BREBES**

Nama : Izzatul Yazidah
NIM : 2008046001
Program Studi : Matematika

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

Wassalamu'alaikum wr.wb.

**Pembimbing,**

**Emy Siswanah, M.Sc**
NIP. 19870202 201101 2 014

# ABSTRAK


**Judul : ANALISIS PERBANDINGAN METODE *MULTILAYER PERCEPTRON* DAN *RECURRENT NEURAL NETWORK* DALAM MEMPREDIKSI LAJU PERTUMBUHAN PENDUDUK DI KABUPATEN BREBES**

**Penulis : Izzatul Yazidah**

**NIM : 2008046001**

Kabupaten Brebes merupakan kabupaten yang memiliki populasi terbesar di Jawa Tengah pada tahun 2020-2022. Jumlah penduduk Kabupaten Brebes selalu meningkat dari tahun ke tahun. Perlu dilakukan prediksi laju pertumbuhan penduduk di Kabupaten Brebes agar pemerintah dapat mempersiapkan peningkatan kebutuhan sumber bumi dan lapangan pekerjaan untuk sejumlah penduduk pada suatu wilayah di waktu yang akan datang. Penelitian ini adalah menganalisis hasil prediksi laju pertumbuhan penduduk Kabupaten Brebes menggunakan metode *Multilayer Perceptron* (MLP) dan *Recurrent Neural Network* (RNN). Kedua metode tersebut digunakan untuk memilih metode terbaik dalam memprediksi laju pertumbuhan penduduk. Metode terbaik ditentukan dengan cara membandingkan MAPE yang menghasilkan nilai terkecil. Data yang digunakan bersumber dari Badan Pusat Statistik (BPS) Kabupaten Brebes yaitu data jumlah penduduk Kabupaten Brebes tahun 1991-2022. Persentase pembagian data *training* dan data *testing* yang digunakan adalah 80%:20%. Berdasarkan hasil penelitian, didapatkan bahwa metode *Recurrent Neural Network* sebagai metode terbaik karena menghasilkan nilai MAPE terkecil sebesar 1,9973%.



***Kata kunci***: Prediksi, Laju Pertumbuhan Penduduk, *Multilayer Perceptron, Recurrent Neural Network.*

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT yang telah melimpahkan berkat dan rahmat-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul "Analisis Perbandingan Metode *Multilayer Perceptron* dan *Recurrent Neural Network* dalam Memprediksi Laju Pertumbuhan Penduduk di Kabupaten Brebes" dengan lancar dan baik. Salawat serta salam semoga senantiasa tercurah kepada Nabi Muhammad SAW. yang semoga di akhirat kelak mendapatkan syafaat beliau.

Skripsi ini merupakan syarat untuk memperoleh gelar sarjana dalam Ilmu Matematika Fakultas Sains dan Teknologi Universitas Islam Negeri Walisongo Semarang. Penyusunan skripsi ini dapat diselesaikan karena bantuan dukungan, serta bimbingan beberapa pihak. Penulis mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Nizar, M.Ag. selaku Rektor Universitas Islam Negeri Walisongo Semarang.
2. Bapak Prof. Dr. H. Musahadi, M.Ag. selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Walisongo.
3. Ibu Any Muanalifah, M.Si., Ph.D. selaku Ketua Program Studi Matematika.
4. Bapak Prihadi Kurniawan, M.Sc. selaku Sekretaris Program Studi Matematika.

5. Ibu Emy Siswanah, M.Sc. selaku dosen pembimbing yang telah memberikan bimbingan, masukan, dan semangat dalam penyusunan skripsi ini.
6. Bapak Agus Wayan Yulianto, M.Sc. selaku dosen wali yang telah memberikan arahan, bimbingan, dan semangat selama proses perkuliahan.
7. Seluruh Bapak/Ibu dosen Program Studi Matematika Universitas Islam Negeri Walisongo yang telah memberikan ilmu yang bermanfaat kepada penulis.
8. Kedua orang tua penulis, Bapak Mustafid dan Ibu Emi Sofiyatin yang telah memberikan dukungan, semangat, dan doa yang tulus sehingga Tugas Akhir skripsi ini dapat terselesaikan dengan baik.
9. Muhammad Zulfa Nurfallah, kakak yang selalu memberikan semangat dan bantuan kepada penulis.
10. Shafa, Dwi, dan Ikha yang selalu membantu, berbagi cerita, memberikan informasi, memberikan semangat, dan menjadi sahabat yang selalu menemani penulis.
11. Kharisma, Maulita, Isti, dan Najwa selaku teman-teman Angkatan 2020 dari Bina Insani yang selalu memberikan semangat dan dukungan bagi penulis.
12. Teman-teman Matematika 2020 yang telah memberikan informasi dan pengalaman.

13. Seluruh pihak yang memberikan kontribusi sampai tugas akhir ini terselesaikan yang tidak dapat disebutkan satu persatu oleh penulis.

Semoga kebaikan yang diberikan kepada penulis, mendapatkan pahala dan menjadi amal ibadah yang diterima oleh Allah SWT bagi semuanya. Dalam skripsi ini, penulis menyadari masih banyak kekurangan dan kelemahan maka penulis mengharapkan kritik dan saran yang membangun dari semuanya. Semoga skripsi ini dapat memberikan manfaat.

Semarang, 27 Juni 2024

**Izzatul Yazidah**
NIM. 2008046001

## DAFTAR ISI

## DAFTAR TABEL

## DAFTAR GAMBAR

## DAFTAR LAMPIRAN

# BAB I
# PENDAHULUAN

## A. Latar Belakang

Indonesia adalah negara yang memiliki populasi besar di bumi dengan jumlah penduduk yaitu 275.501.339 jiwa. Indonesia menempati peringkat keempat di dunia dalam daftar jumlah penduduk yang besar setelah India, Cina, dan Amerika Serikat. Setiap tahunnya, pertumbuhan penduduk di Indonesia selalu mengalami peningkatan (World Bank, 2022). Pertumbuhan penduduk yang tinggi akan menjadi tantangan yang dihadapi bagi negara berkembang seperti Indonesia. Keseimbangan sumber daya alam akan terganggu akibat dari meningkatnya jumlah penduduk (Akhirul, 2020).

Jumlah penduduk yang terus meningkat akan membuat bangunan-bangunan permukiman maupun bukan permukiman semakin bertambah. Akibatnya suatu wilayah yang luasnya tetap akan semakin padat oleh jumlah penduduk sehingga pergerakan manusia di suatu wilayah akan berkurang dan terbatas. Ketersediaan konsumsi juga akan berbanding lurus dengan bertambahnya penduduk. Manusia memerlukan sumber kebutuhan untuk bertahan hidup, semakin banyak konsumen maka akan semakin banyak kebutuhan yang diperlukan (Akhirul, 2020).

Pertumbuhan penduduk juga akan mempengaruhi struktur kependudukan dari segi kuantitas dan kualitas. Pengaruh secara kuantitas seperti umur, jenis kelamin, usia kerja, dan angkatan kerja. Sedangkan pengaruh secara kualitas seperti pendidikan, kesehatan, dan faktor lainnya. Pertumbuhan penduduk dapat menjadikan faktor pendorong dan faktor penghambat pembangunan. Jika peningkatan jumlah penduduk tidak diiringi dengan perkembangan kualitasnya maka akan menimbulkan bermacam masalah seperti lapangan pekerjaan yang terbatas, kawasan Ruang Terbuka Hijau (RTH) yang berkurang, dan menurunnya kesejahteraan sosial. Hal tersebut akan mengakibatkan tidak terwujudnya pembangunan ekonomi (Hardati, 2013).

Kabupaten Brebes merupakan kabupaten/kota dengan luas 1.902,40 $km^2$ dan menempati urutan ketiga di Jawa Tengah (BPS, 2021). Pada tahun 2020 sampai tahun 2022 Kabupaten Brebes menjadi kabupaten yang memiliki populasi terbesar di Jawa Tengah dengan jumlah berturut-turut adalah 1.978.759 jiwa, 1.992.685 jiwa, dan 2.010.617 jiwa. Jumlah penduduk Kabupaten Brebes selalu mengalami peningkatan dari tahun 1991-2022. Berdasarkan data jumlah penduduk menurut BPS pada 1991 jumlah penduduk Kabupaten Brebes sebanyak 1.536.534 jiwa meningkat menjadi 2.010.617 jiwa pada

tahun 2022. Jumlah penduduk di Kabupaten Brebes tersebut tersebar di 17 Kecamatan (BPS, 2022).

Jumlah penduduk di Kabupaten Brebes setiap tahunnya mengalami peningkatan. Untuk mencegah terjadinya ledakan penduduk maka perlunya kesiapan bagi pemerintah sekitar untuk melakukan upaya agar bisa menekan laju pertumbuhan dan mempersiapkan peningkatan dari jumlah penduduk pada suatu wilayah. Persiapan sumber bumi untuk mengimbangi kebutuhan manusia yang terus meningkat agar tidak menimbulkan kesengsaraan. Lapangan pekerjaan perlu ditingkatkan agar tidak terjadinya pengangguran sehingga masyarakat dapat memenuhi kebutuhan ekonominya (Akhirul, 2020).

Sejak tahun 2019, Kabupaten Brebes menjadi kabupaten yang disebut sebagai salah satu wilayah miskin ekstrim (Fauziyah & Utomo, 2023). Penduduk dengan jumlah yang besar dapat menjadi peluang sekaligus tantangan bagi Kabupaten Brebes. Dalam hal ini, penduduk yang besar dapat menjadi peluang untuk pembangunan Kabupaten Brebes jika penduduknya berkualitas dan akan menjadi masalah jika penduduknya tidak berkualitas. Oleh karena itu, perlu dilakukan prediksi jumlah kepadatan penduduk beberapa tahun mendatang. Terdapat beberapa metode yang dapat digunakan dalam analisis prediksi,

salah satunya adalah metode jaringan saraf tiruan (Arafat, 2021).

Jaringan saraf tiruan (JST) atau dapat disebut dengan *Artificial Neural Network* (ANN) merupakan suatu pola dalam pemrosesan informasi yang cara operasinya termotivasi dari pemrosesan informasi dalam otak manusia. Beberapa neuron yang saling menghubungkan satu sama lain kemudian menyelesaikan masalah secara bersamaan akan membentuk sistem pemrosesan informasi (Hafizah *et al*., 2023). Jaringan saraf tiruan merupakan paradigma komputasi yang mampu mengambil dan menampilkan koneksi *input-output* yang rumit menggunakan pemodelan data. JST telah berhasil diimplementasikan di berbagai masalah yang melibatkan prediksi karena kemampuannya dalam mengatasi sejumlah masalah relatif mudah digunakan, kemampuan dalam menginput data yang rumit, pengoperasian yang cepat, dan menganalisis proses yang kompleks (Sudarsono, 2016).

Secara umum, ada tiga jenis *neural network* yang sering digunakan berdasarkan jenis jaringannya yaitu, *Single-Layer Neural, Multilayer Perceptron Neural Network dan Recurrent Neural Network*. Jaringan *Single-Layer* adalah jaringan saraf tiruan yang mempunyai lapisan *input* terkoneksi tepat di lapisan *output*. Jaringan *Multilayer*

*Perceptron* (MLP) adalah jaringan saraf tiruan yang di antara lapisan *input* dan lapusan *output* terdapat lapisan tersembunyi. Sedangkan *Recurrent Neural Network* adalah jaringan saraf tiruan yang mempunyai koneksi umpan balik dari *output* menuju *input* (Pakaja *et al.,* 2012).

*Multilayer Perceptron* (MLP) merupakan metode jaringan saraf tiruan yang mempunyai kelebihan pada satu atau lebih lapisan tersembunyi, memiliki beberapa unit dibandingkan dengan metode *perceptron* dengan lapisan tunggal. MLP mempunyai karakteristik yaitu nilai bobot yang ditentukan lebih baik daripada metode lain, MLP dapat diaplikasikan tanpa informasi sebelumnya, dan mampu dalam mengatasi permasalahan linear dan nonlinear. Karakteristik yang ada pada MLP tersebut menghasilkan metode MLP menjadi lebih baik dalam prediksi (Wibawa, 2020).

Pada penelitian Choubin *et al*. (2014) menunjukkan bahwa kinerja jaringan *Multilayer Perceptron* lebih baik dibandingkan dengan metode *Multiple Linear Regression* dan *Adaptive Neuro-Fuzzy inference* dalam memprediksi curah hujan karena menghasilkan nilai RMSE terkecil yaitu 0,86. Pada penelitian Nooriansyah *et al*. (2018) menunjukkan metode *Artificial Neural Network* dengan pelatihan *Multilayer Perceptron* (MLP) mendapatkan MAPE dengan nilai terkecil yaitu 6,88% dibandingkan *Support*

*Vector Regression* (SVR) yaitu 8,51% dalam memprediksi tinggi gelombang. Penelitian yang dilakukan Ariwibowo *et al*. (2019) menunjukkan bahwa untuk metode *Multilayer Perceptron* memiliki nilai MAPE yang paling kecil yaitu 4,48% dibandingkan metode *Holt-Winters* yaitu 28,65% dan metode *Auto Refressive Integrated Moving Average* (ARIMA) yaitu 28,16% dalam memprediksi harga beras IR64 kualitas III.

*Recurrent Neural Network* atau disebut jaringan umpan balik adalah jenis jaringan saraf tiruan yang memiliki *loop* sebagai koneksi umpan balik dalam jaringan (Tarkus *et al*., 2020). Menurut Wang (2016) menyatakan bahwa RNN memiliki keunggulan seperti kemampuan prediksi deret waktu (*time series*) dan nonlinier, kecepatan dalam konvergensi, dan mampu dalam pemetaan yang lebih akurat. Selain itu, kelebihan yang dimiliki RNN yaitu memproses data yang masuk secara berulang-ulang membentuk sebuah *time series*. Secara konseptual, RNN memiliki memori yang kuat terhadap kejadian yang pernah terjadi lebih dahulu, namun hal tersebut sulit apabila dilakukan untuk memiliki sekuens yang panjang (Farhah *et al*., 2021).

Beberapa penelitian terdahulu yang membahas prediksi dengan menggunakan metode RNN di antaranya penelitian yang dilakukan oleh Hermawan (2014) yang

menggunakan metode *Recurrent Neural Network* dan *Recurrent Neuro Fuzzy* dalam memprediksi banyaknya penumpang kereta api di Jabodetabek. Penelitian tersebut menghasilkan nilai MAPE terkecil yaitu 3,785% pada proses pengujian dengan metode RNN. Pada penelitian Achmalia *et al*. (2021) dalam memprediksi penjualan semen menggunakan *Backpropagation Neural Network* dan *Recurrent Neural Network* tipe Elman, diperoleh MAPE sebesar 28,9958% dengan metode RNN. Penelitian yang dilakukan Kusnanti *et al*. (2022) dalam memprediksi kecepatan dan arah arus permukaan laut dengan metode *Elman Recurrent Neural Network* menghasilkan nilai MAPE pada proses pengujian sebesar 3,1253%.

Berdasarkan penelitian terdahulu yang sebelumnya telah disebutkan yaitu metode MLP jika dibandingkan dengan metode yang lain, metode MLP memperoleh nilai *error* terendah. Begitu pula dengan RNN, metode RNN menghasilkan peramalan yang lebih baik jika dibandingkan dengan metode prediksi yang lain. Berdasarkan alasan tersebut maka penelitian ini akan membandingkan metode *Multilayer Perceptron* dan metode *Recurrent Neural Network*. Kedua metode tersebut diimplementasikan untuk memprediksi data *time series* dan akan dibandingkan nilai MAPE yang diperoleh. Oleh karena itu, penulis mengangkat judul "Analisis Perbandingan Metode *Multilayer Perceptron*

dan *Recurrent Neural Network* dalam Memprediksi Laju Pertumbuhan Penduduk di Kabupaten Brebes".

## B. Rumusan Masalah

1. Bagaimana hasil prediksi laju pertumbuhan penduduk di Kabupaten Brebes menggunakan metode *Multilayer Perceptron*?
2. Bagaimana hasil prediksi laju pertumbuhan penduduk di Kabupaten Brebes menggunakan metode *Recurrent Neural Network*?
3. Bagaimana perbandingan tingkat akurasi prediksi laju pertumbuhan penduduk di Kabupaten Brebes menggunakan metode *Multilayer Perceptron* dan *Recurrent Neural Network*?

## C. Tujuan Penelitian

1. Untuk mengetahui hasil prediksi laju pertumbuhan penduduk di Kabupaten Brebes menggunakan metode *Multilayer Perceptron*.
2. Untuk mengetahui hasil prediksi laju pertumbuhan penduduk di Kabupaten Brebes menggunakan metode *Recurrent Neural Network*.
3. Untuk mengetahui perbandingan tingkat akurasi prediksi laju pertumbuhan penduduk di Kabupaten

Brebes menggunakan metode *Multilayer Perceptron* dan *Recurrent Neural Network*.

## D. Manfaat Penelitian

1. Hasil prediksi diharapkan dapat memberi gambaran laju pertumbuhan penduduk kepada pemerintah dalam beberapa tahun ke depan.
2. Mengetahui kebutuhan yang harus dipersiapkan, sehingga pemerintah dapat mempersiapkan sedini mungkin.
3. Memperbanyak pengetahuan bagi penulis dan pembaca mengenai metode prediksi *Multilayer Perceptron* dan *Recurrent Neural Network*.

# BAB II
# LANDASAN PUSTAKA

## A. Laju Pertumbuhan Penduduk

Penduduk adalah setiap jiwa yang dalam jangka waktu enam bulan atau lebih bertempat tinggal di wilayah dan atau mereka yang memiliki tujuan menetap selama kurang dari enam bulan. Pertumbuhan penduduk dapat dikatakan sebagai peningkatan dan penurunan kemampuan populasi dengan keseimbangan dinamis yang selalu dipengaruhi oleh jumlah bayi yang baru lahir (pertumbuhan populasi) dan pada saat yang sama akan dikurangi oleh angka mortalitas di setiap kelompok umur. Dari pengertian tersebut, dapat disimpulkan bahwa yang mempengaruhi pertumbuhan penduduk ada 4 faktor yaitu kelahiran, kematian, imigrasi, dan emigrasi (BKKBN, 2012).

Adanya jumlah penduduk yang besar memiliki dampak yang ditimbulkan ke berbagai masalah dimensi. Menurut Tambunan (2003), jumlah penduduk besar pada sisi permintaan memiliki potensi untuk pasar dapat tumbuh besar, yang berarti faktor pertumbuhan bagi kegiatan ekonomi. Di sisi penawaran, penduduk dengan jumlah besar, berpendidikan dan sehat, disiplin dan etos kerja yang kuat merupakan keuntungan yang penting bagi produksi. Dari sisi yang lain, jumlah penduduk merupakan

faktor terpenting yang menentukan permintaan barang konsumsi yang disediakan dan sarana umum yang didirikan di daerah tersebut akan bertambah (Robinson, 2012).

Laju pertumbuhan penduduk adalah rerata pertumbuhan penduduk tahunan antara dua sensus. Laju pertumbuhan penduduk dapat pula dikatakan sebagai angka yang memperlihatkan tingkatan pertambahan penduduk setiap tahun selama rentang waktu tertentu. Nelson dan Leibstein menunjukkan bahwa negara berkembang dengan pertumbuhan penduduk yang cepat berarti tingkat kesejahteraan masyarakat tidak mengalami kenaikan secara signifikan dan dalam waktu yang lama terjadi penurunan tingkat kesejahteraan. Jumlah penduduk miskin akan meningkat yang dapat menyebabkan tumbuhnya kejahatan di masyarakat.

Berikut rumus perhitungan laju pertumbuhan penduduk (BPS, 2010) dengan metode geometri:

$$r = \left(\frac{P_t}{P_0}\right)^{\frac{1}{t}} - 1 \qquad (2.1)$$

Keterangan:

$P_t$ = jumlah penduduk tahun ke-t

$P_0$ = jumlah penduduk tahun dasar

$r$ = laju pertumbuhan penduduk

$t =$ periode waktu antara tahun ke-t dan tahun dasar (dalam tahun)

Berikut adalah tabel 2.1 kategori laju pertumbuhan penduduk (Desriwendi *et al*., 2015).

Tabel 2.1 Kategori Laju Pertumbuhan Penduduk

| **Laju pertumbuhan penduduk (r)** | **Kategori** |
|---|---|
| $r > 0$ | Terjadi penambahan jumlah penduduk |
| $r = 0$ | Tidak ada perubahan jumlah penduduk |
| $r < 0$ | Terjadi pengurangan jumlah penduduk |

## B. Pengertian Prediksi

Prediksi adalah suatu cara mengasumsikan atau menduga secara struktur suatu peristiwa yang kemungkinan besar terjadi di masa mendatang dengan sumber data dari masa lampau dan saat ini, agar kecacatannya (perbedaan antara hal yang sebenarnya dengan hasil pendugaan) dapat diperkecil (Setyowanto, 2014). Hasil prediksi atas kejadian yang akan terjadi tidak harus tepat, tetapi berupaya agar mendapatkan hasil atau jawaban yang paling mendekati dengan peristiwa yang akan terjadi di masa depan (Herdianto, 2013). Prediksi bertujuan untuk menghasilkan jawaban yang paling mendekati dengan suatu kejadian yang sebenarnya, bukan untuk memberikan jawaban secara pasti (Irfan *et al*., 2018).

## C. Jaringan Saraf Tiruan

Jaringan saraf tiruan (JST) dirancang pertama kali oleh Neurophysiologist Waren McCulloch dan Walter Pits pada tahun 1943. Waren McCulloch dan Walter Pits menemukan bahwa sekumpulan neuron sederhana yang digabungkan ke dalam sistem saraf dapat memperbaiki kekuatan komputasinya. Bobot neuron diatur sedemikian sehingga fungsi logika yang dilakukan neuron sangat sederhana. Setiap neuron yang berbeda, tugas yang dilakukannya juga akan berbeda. Neuron dirancang membentuk suatu jaringan untuk memberikan *output* (keluaran) apapun yang dapat digambarkan menjadi suatu kombinasi operasi logis. Aliran informasi melewati jaringan adalah satu langkah sinyal bergerak dari satu neuron menuju neuron selanjutnya (Puspitaningrum, 2006).

Neuron merupakan unit dasar pada sistem saraf yang berfungsi untuk menyampaikan dan menerima suatu informasi. Hubungan antar neuron disebut bobot. Bobot mempunyai fungsi sebagai pengatur jaringan sehingga *output* yang dihasilkan oleh jaringan saraf tiruan sesuai yang dinginkan (Jaya *et al*., 2018). Jaringan saraf tiruan adalah struktur pemrosesan pengetahuan atau informasi dengan penrilaku menyerupai jaringan saraf manusia (Hermawan, 2006). Jaringan saraf mensimulasi sistem kerja otak (fungsi saraf biologis) untuk selanjutnya dibawa

kepada perangkat lunak kelas baru yang dapat mengetahui pola yang rumit serta belajar dari pengalaman yang telah terjadi (Suyanto, 2014).

Secara umum, karakteristik jaringan saraf tiruan dapat ditentukan oleh tiga hal berikut.

1). Model hubungan antar neuron (arsitektur)
2). Metode penentuan bobot penghubung yaitu pelatihan (*training*), pembelajaran (*learning*), dan algoritma
3). Fungsi aktivasi (Jong Hek Siang, 2004).

**1. Komponen Jaringan Saraf Tiruan**

Jaringan saraf tiruan mempunyai sejumlah lapisan penyusun yang dapat dibagi ke dalam tiga bagian, yaitu:

a. Lapisan *input* (*input layer*)

Neuron-neuron di dalam lapisan *input* disebut unit-unit *input*. Unit-unit *input* memperoleh *input* dari dunia luar. *Input* yang diterima adalah penggambaran dari permasalahan.

b. Lapisan tersembunyi (*hidden layer*)

Neuron-neuron di dalam lapisan tersembunyi disebut unit-unit tersembunyi. Di dalam lapisan ini terjadi mekanisme tahap pelatihan dan tahap pengenalan.

c. Lapisan *output* (*output layer*)

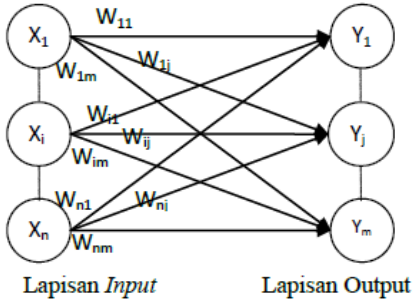
Neuron-neuron di dalam lapisan *output* disebut unit-unit *output*. Pada lapisan ini, dihasilkan *output* yang merupakan keluaran jaringan saraf tiruan terhadap suatu masalah (Jaya *et al*., 2018).

**2. Arsitektur Jaringan Saraf Tiruan**

Struktur jaringan saraf tiruan terbagi menjadi tiga arsitektur, yaitu:

a. Jaringan lapisan tunggal (*single layer network*)

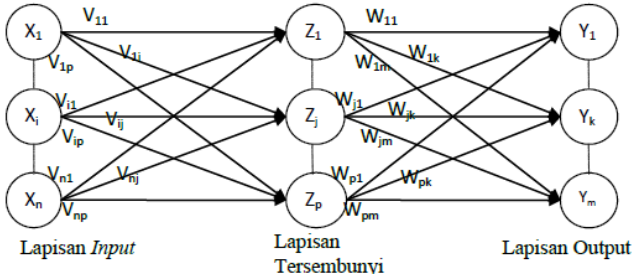
Jaringan lapisan tunggal hanya mempunyai satu lapisan bobot koneksi. Jaringan lapisan tunggal terdiri dari unit-unit *input* yang memperoleh sinyal dari dunia luar. Lapisan *input* ini langsung terkoneksi ke lapisan *output*. Kelemahan *single layer network* yaitu keterbatasan dalam pengenalan pola. Berikut gambar jaringan *single layer.*




Gambar 2.1 Jaringan *Single Layer* (Jaya *et al*., 2018)

b. Jaringan lapisan jamak (*multilayer network*)

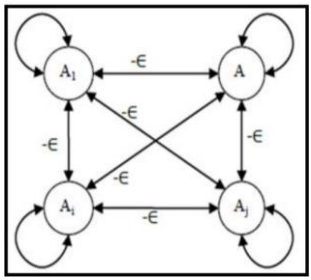
Jaringan lapisan jamak merupakan jaringan dengan satu atau lebih lapisan tersembunyi (*hidden layer*) yang mempunyai node yang disebut neuron tersembunyi. Dalam memecahkan permasalahan, *multilayer network* ini memiliki kemampuan lebih dibandingkan dengan *single layer network*. Berikut gambar jaringan *multilayer*.




Gambar 2.2 Jaringan *Multilayer* (Jaya *et al*., 2018)

c. Jaringan lapisan kompetitif (*competitive network*)

Pada jaringan lapisan kompetitif, sekelompok neuron bersaing untuk memperoleh hak menjadi aktif. Setiap neuron dalam jaringan ini dapat saling terhubung. Salah satu jaringan lapisan kompetitif adalah *Recurrent Neural Network*, yaitu jaringan dengan koneksi umpan balik (*feedback link*) (Nutriyasari, 2014). Berikut gambar jaringan kompetitif.




Gambar 2.3 Jaringan *Competitive* (Jaya *et al*., 2018)

**3. Tahapan Pemrosesan Informasi**

Secara umum jaringan saraf tiruan mempunyai dua langkah pemrosesan informasi, yaitu proses pelatihan (*training*) dan proses pengujian (*learning*) (Warsito, 2009).

a. Proses pelatihan

Proses pelatihan diawali dengan menginputkan data pelatihan pada jaringan. Jaringan akan memperbaiki bobot yang merupakan penghubung antar neuron atau node dengan menggunakan data tersebut. Terwujudnya hasil sesuai dengan yang diinginkan harus melewati proses pelatihan pada jaringan. Evaluasi akan dilakukan pada setiap iterasi.

b. Proses pengujian

Proses pengujian dilakukan dengan menguji data inputan yang berbeda dari data pelatihan menggunakan bobot-bobot yang didapatkan dari

tahap pelatihan. Proses pengujian memberikan *error* yang sedikit akan dipengaruhi oleh bobot-bobot hasil pelatihan dengan *error* seminimal mungkin.

**4. Pembagian Data**

Data diartikan sebagai kumpulan informasi yang menjadi suatu masukan (*input*) dalam sistem sehingga dapat memberikan analisis yang sesuai. Data berfungsi untuk melakukan pembelajaran dalam suatu jaringan saraf. Data dibagi menjadi data latih (*training*) dan data uji (*testing*). Data *training* adalah data yang digunakan untuk melatih atau membentuk pola dalam jaringan saraf. Sedangkan data *testing* adalah data yang digunakan untuk mengetahui kinerja dari pola dalam jaringan saraf yang sebelumnya dilatih yaitu saat mendapatkan data baru yang tidak diberikan pada data *training* (Retnoningsih & Pramudita, 2020).

**5. Algoritma Pembelajaran**

Pembelajaran dalam JST adalah proses penyesuaian parameter-parameter bebas JST melalui mekanisme perangsangan berkelanjutan oleh bagian suatu jaringan (Suyanto, 2008). Tujuan tahap pembelajaran adalah menerapkan kontrol terhadap bobot JST agar bobot akhir yang diperoleh sesuai dengan pola data pelatihan (Kusumadewi & Hartati, 2010). Pada

proses pembelajaran akan terjadi perbaikan bobot, sehingga apabila informasi yang dikirim menuju suatu neuron dapat tersampaikan ke neuron lainnya maka nilai bobot akan naik dan apabila informasi yang dikirim menuju suatu neuron tidak tersampaikan ke neuron lain maka nilai bobot akan turun (Nurtiyasari, 2014).

Ada dua metode pembelajaran dalam jaringan saraf, yaitu sebagai berikut (Kusumadewi, 2003).

a. Pembelajaran terawasi (*supervised learning*)

   Disebut pembelajaran terawasi, jika *output* yang diharapkan sudah diketahui sebelumnya. Pada proses klasifikasi dan pengenalan pola biasanya menggunakan pembelajaran ini. Pada proses pembelajaran, jaringan akan menepatkan bobot sinaptiknya.

b. Pembelajaran tak terawasi (*unsupervised learning*)

   Pembelajaran tak terawasi tidak memerlukan target *output*. Metode ini bertujuan untuk mengelompokkan unit-unit yang mirip dalam bagian tertentu (*Clustering*). *Input* dan *output* yang diterima jaringan diatur sendiri sesuai aturan yang ada.

**6. Fungsi Aktivasi atau Fungsi Transfer**

Fungsi aktivasi merupakan fungsi yang menggambarkan hubungan antara tingkat aktivasi internal atau *summation function* yang dapat berbentuk linier maupun nonlinier. Untuk membangkitkan neuron diperlukan fungsi aktivasi. Tujuan fungsi aktivasi adalah untuk mengganti sinyal *output* menjadi sinyal *input* (Ardilla, 2016).

Dalam tugas akhir ini, fungsi aktivasi yang digunakan adalah sebagai berikut.

a. Fungsi Sigmoid Bipolar

Fungsi sigmoid bipolar bersifat kontinu dan terdiferensiasi dengan nilai keluarannya terletak pada interval -1 sampai 1. Fungsi sigmoid bipolar dalam matlab disebut *tansig* yang dirumuskan sebagai berikut (Sharma, 2020).

$$f(x) = \frac{2}{1 + e^{-x}} - 1 \tag{2.2}$$

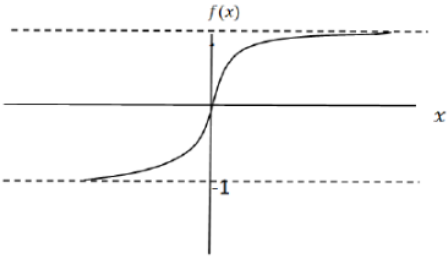
yang dapat disederhanakan menjadi:

$$f(x) = \frac{2}{1 + e^{-x}} - \frac{1 + e^{-x}}{1 + e^{-x}} \tag{2.3}$$

$$f(x) = \frac{2 - (1 + e^{-x})}{1 + e^{-x}} \tag{2.4}$$

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \tag{2.5}$$

Berikut adalah gambar fungsi sigmoid bipolar.



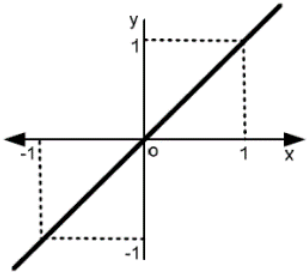

Gambar 2.4 Fungsi Sigmoid Bipolar (Lesnussa *et al*., 2017)

b. Fungsi Identitas

Nilai *output* yang dihasilkan pada fungsi identitas sama dengan nilai *input*nya. Fungsi identitas dalam matlab disebut *purelin* yang dirumuskan sebagai berikut (Yeung *et al*., 2010).

$$f(x) = x \tag{2.6}$$

Berikut adalah gambar fungsi identitas.




Gambar 2.5 Fungsi Identitas (Pamungkas *et al*., 2022)

**7. Parameter Pembelajaran**

Dalam melakukan prediksi, ada beberapa parameter yang akan mengendalikan laju prediksi

dalam lapisan tersembunyi, diantaranya sebagai berikut.

*a. Epoch*

*Epoch* merupakan salah satu parameter yang digunakan sebagai kondisi *stopping* dalam pembelajaran *neural network*. Proses pembelajaran dihentikan apabila besarnya *epoch* lebih besar dari besarnya *epoch* maksimum yang telah ditetapkan (Kholis, 2015).

b. Bobot

Bobot pada *neural network* mempunyai peran sebagai perantara antar lapisan dan mengalikan nilai yang diperoleh dari *input*. Hasil *output* yang diinginkan dari nilai *input* tertentu dipengaruhi oleh pengaturan bobot (Ryandhi, 2017).

c. Bias

Bias adalah nilai ketidakpastian pada suatu neuron dalam lapisan. Pemberian bias dilakukan pada setiap neuron di lapisan tersembunyi (Ryandhi, 2017).

*d. Learning rate*

*Learning rate* merupakan salah satu parameter untuk menghitung nilai koreksi bobot dalam tahap pembelajaran. Rentang nilai *learning rate* yaitu dari nol sampai satu. Nilai *learning rate*

yang semakin besar maka langkah pembelajaran pun akan berjalan semakin cepat. Jika *learning rate* semakin besar maka ketelitian jaringan akan semakin berkurang. Jika *learning rate* memiliki nilai semakin kecil maka akan semakin besar ketelitian jaringan sehingga proses pembelajaran akan berjalan semakin lama (Nurfita, 2018).

## D. Normalisasi Data

Sebelum dilakukan tahap pelatihan, perlu dilakukan normalisasi data yaitu penskalaan terhadap nilai-nilai *input* dan target sehingga masuk ke dalam suatu *range* tertentu. Melakukan normalisasi bertujuan agar data *input* dan target sesuai dengan fungsi aktivasi yang digunakan (Kusumadewi, 2004). Rumus perhitungan normalisasi adalah sebagai berikut.

$$x_i' = \frac{0{,}8(x_i - x_{min})}{(x_{max} - x_{min})} + 0{,}1 \qquad (2.7)$$

Keterangan:

$x_i'$ = hasil normalisasi data x pada indeks ke-i

0,8 = nilai default normalisasi nilai optimum

$x_i$ = data yang akan dinormalisasi

$x_{min}$ = nilai data terendah

$x_{max}$ = nilai data tertinggi

0,1 = nilai default normalisasi nilai minimum (Windarto *et al*., 2020).

## *E. Multilayer Perceptron*

*Multilayer perceptron* merupakan *artificial neural network* turunan dari *perceptron*, berupa ANN dengan satu atau lebih lapisan tersembunyi yang terdapat di antara lapisan *input* dan lapisan *output*. Antara dua lapisan yang bersebelahan terdapat beberapa lapisan yang memiliki bobot. Permasalahan yang mempunyai tingkat kesulitan tinggi lebih baik diselesaikan dengan jaringan *multilayer* daripada jaringan *single layer* karena proses pembelajaran pada *multilayer* yang lebih detail (Ardilla, 2016). Lapisan yang terdapat pada *Multilayer Perceptron* yaitu lapisan masukan, lapisan tersembunyi, dan lapisan luaran yang ditunjukkan oleh gambar 2.2.

Langkah-langkah algoritma *Multilayer Perceptron* adalah sebagai berikut (Khoirudin *et al*., 2018).

1. Menginisiasi bobot acak dengan rentang dari 0 sampai 1,
2. Menginisiasi bias acak dengan rentang dari 0 sampai 1.
3. Menghitung besarnya *input* dari lapisan *input* ke lapisan tersembunyi menggunakan rumus berikut.

$$Input_j = \sum_{i=1}^{n} O_i w_{ij} + \theta_j^- \qquad (2.8)$$

Keterangan:

$O_i$ = output simpul $i$ dari lapisan sebelumnya

$w_{ij}$= bobot relasi dari simpul $i$ pada lapisan sebelumnya ke simpul j

$\theta_j$ = bias (sebagai pembatas)

4. Membangkitan *output* dari lapisan tersembunyi ke lapisan *output* dengan menggunakan fungsi aktivasi.
5. Menghitung nilai *error* antara hasil prediksi dengan nilai sebenarnya menggunakan rumus berikut.

$$Error_j = Output_j \cdot (1 - Output_j) \cdot (Target_j - Output_j) \quad (2.9)$$

Keterangan:

$Output_j$ = keluaran sebenarnya dari simpul j

$Target_j$ = nilai target yang sudah diketahui pada data *training*

6. Setelah menghitung nilai *error*, selanjutnya kembali ke lapisan sebelumnya untuk menghitung *error* pada lapisan tersembunyi menggunakan persamaan berikut.

$$Error_j = Output_j \cdot (1 - Output_j) \cdot \sum_{i=1}^{n} Error_k w_{jk} \quad (2.10)$$

Keterangan:

$Output_j$ = keluaran sebenarnya dari simpul j

$Error_k$ = *error* simpul k

$w_{jk}$ = bobot relasi dari simpil j ke simpul k pada lapisan berikutnya.

7. Nilai *error* sebelumnya diterapkan untuk memperbarui nilai bobot menggunakan persamaan berikut.

$$w_{ij} = w_{ij} + l \cdot Error_j \cdot Output_j \quad (2.11)$$

Keterangan:

$w_{ij}$ = bobot relasi dari unit $i$ pada lapisan sebelumnya ke unit $j$

$l$ = *learning rate* (konstanta, nilainya antara 0 sampai dengan 1).

8. Jika sudah mencapai maksimal *epoch* yang ditetapkan atau nilai *error* adalah 0, maka iterasi dihentikan.

## *F. Recurrent Neural Network*

### 1. Arsitektur *Recurrent Neural Network*

*Recurrent Neural Network* (RNN) adalah jenis jaringan pada *neural network* dengan fasilitas umpan balik menuju neuron itu sendiri dan neuron lainnya sehingga alur suatu informasi dari *input* memiliki banyak tujuan (Purnomo & Kurniawan, 2006). RNN dibuat khusus agar dapat memproses data yang bersambung atau berurutan. Dalam proses pembelajarannya, RNN dapat mengoperasikan data dari masa lampau. Dikarenakan RNN dapat

memproses data bersambung maka RNN dapat juga diaplikasikan ketika mengelola data *time series.* Lapisan yang menyusun RNN yaitu lapisan *input,* lapisan tersembunyi, dan lapisan *output* (Wardani, 2021).

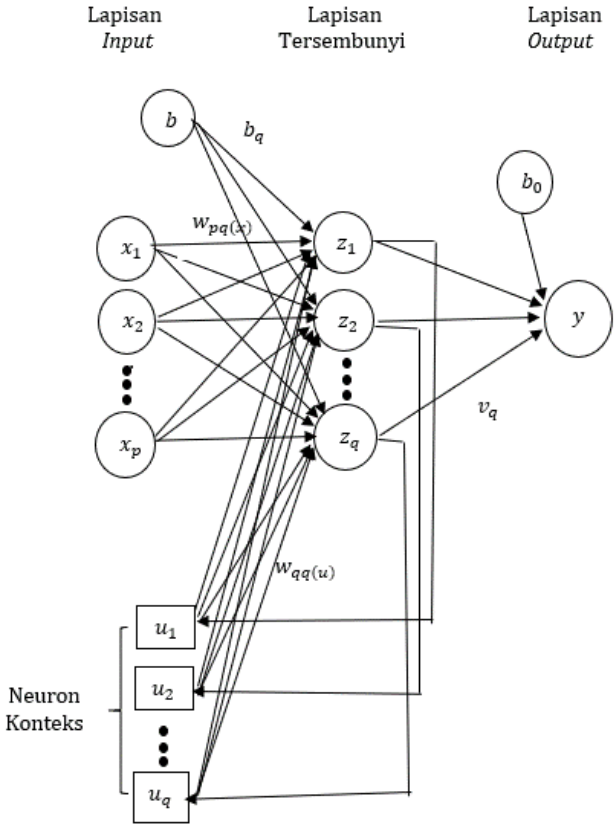
RNN menggunakan data *input* pada waktu sekarang dan juga *input* dari data sebelumnya. Hubungan antar input tersebut berfungsi untuk menyampaikan informasi menuju semua lapisan tersembunyi sehingga RNN memiliki ingatan yang memuat hasil rekaman informasi yang dihasilkan sebelumnya (Gulli, 2017). RNN adalah jaringan yang mengakomodasi *output* jaringan untuk menjadi *input* pada jaringan tersebut yang dimanfaatkan untuk memperoleh *output* baru (Hermawan, 2014).

Ada dua macam model RNN yaitu Jaringan Elman dan Jaringan Hopfield. Jaringan Elman merupakan jaringan *backpropagation* dua lapis dengan peningkatan koneksi *feedback* dari *output* di lapisan tersembunyi menuju *input*. Adanya *feedback* tersebut menjadikan jaringan Elman mampu mempelajari, mengenali, dan membuat pola sementara seperti pola spasial. Jaringan Hopfield digunakan untuk menyimpan satu atau beberapa vektor target yang dianggap sebagai memori yang akan dipanggil

jika ada vektor yang menyerupai dengan memori jaringan (Musashi, 2014).

**2. Jaringan Elman**

Jaringan Elman berupa jaringan dua lapis dengan *feedback* dari *output* lapisan pertama ke *input* lapisan pertama. Dalam jaringan ini, membolehkan output dari lapisan tersembunyi untuk menyampaikan umpan balik ke dirinya sendiri melewati lapisan penyangga, yang dikenal lapisan berulang. Masing-masing neuron tersembunyi hanya terkoneksi untuk satu neuron lapisan berulang dengan bobot nilai yang konstan. Oleh sebab itu, lapisan berulang sebenarnya merupakan salinan kondisi lapisan tersembunyi sebelumnya. Dapat disimpulkan bahwa Jaringan Elman terdiri dari lapisan *input*, lapisan berulang yang menyediakan informasi keadaan, lapisan tersembunyi, dan lapisan *output.* Masing-masing lapisan memuat satu atau beberapa neuron yang menyampaikan informasi ke seluruh lapisan dengan menghitung fungsi nonlinier dan jumlah *input* tertimbangnya (Wang, 2016). Berikut ini ditunjukkan model Jaringan Elman.




Gambar 2.6 Arsitektur Jaringan Elman

Pada gambar 2.6 menunjukkan model Jaringan Elman memiliki jumlah neuron pada lapisan input sejumlah $p$ yaitu $x_1, x_2, \dots, x_p$, pada lapisan tersembunyi sejumlah $q$ yaitu $z_1, z_2, \dots, z_p$, dan satu output yaitu $y$. $u_1, u_2, \dots, u_q$ adalah neuron tambahan (neuron konteks) yang merupakan umpan balik (*feedback link*) dari *hidden layer*. Bobot dari neuron *j*

pada lapisan *input* ($j$ =1,2, ..., p) ke neuron $k$ pada lapisan tersembunyi ($k$ =1,2, ..., q) disimbolkan dengan $w_{jk}$. Sedangkan $b_k$ dan $b_0$ adalah bobot bias dari neuron ke-$k$ pada lapisan tersembunyi dan bobot bias pada neuron lapisan *output*.

Secara matematis, model RNN Elman dapat dirumuskan sebagai berikut.

$$z = \sum_{k=1}^{q} v_k \frac{1 - \exp\left(-\left(\sum_{j=1}^{p} x_j w_{jk(x)} + \sum_{k=1}^{q} u_k w_{kk(u)} + b_k\right)\right.}{1 + \exp\left(-\left(\sum_{j=1}^{p} x_j w_{jk(x)} + \sum_{k=1}^{q} u_k w_{kk(u)} + b_k\right)\right.} + b_0 + \varepsilon \qquad (2.12)$$

Keterangan:

$z =$ nilai *output* (keluaran)

$v_k =$ bobot dari neuron ke-k lapisan tersembunyi ke lapisan output, $k = 1,2, \dots, q$

$x_j =$ variabel input, , $j = 1,2, \dots, p$

$w_{jk(x)} =$ bobot dari neuron ke-j lapisan *input* ke lapisan tersembunyi

$u_k =$ variabel input di neuron konteks, $k = 1,2, \dots, q$

$w_{kk(u)} =$ bobot dari neuron konteks ke-k menuju neuron lapisan tersembunyi ke-k

$b_k =$ bobot bias dari neuron ke-$k$ pada lapisan tersembunyi $b_0 =$ bobot bias pada neuron lapisan *output*

$\varepsilon =$ *error*

**3. Jaringan Hopfield**

Jaringan Hopfield akan merespon untuk mendesain suatu titik dan menyimpan sekelompok titik pada posisi yang proporsional apabila jaringan diberikan suatu kondisi awal. Jaringan ini bekerja secara rekursif sehingga *output* jaringan akan disampaikan kembali sebagai *input* jaringan pada setiap proses. Pengujian jaringan ini dilakukan menggunakan satu atau lebih vektor *input* yang disajikan oleh kondisi awal jaringan. Setelah memberikan *input*, jaringan akan mengahasilkan *output* yang akan disampaikan menjadi *input* kembali. Proses terjadi secara berkelanjutan hingga mendapatkan nilai *output* yang stabil. Setiap vektor *input* akan menyatu pada satu titik keseimbangan terdekat. Dapat disimpulkan bahwa algoritma Hopfield akan mencoba untuk menstabilkan *output* suatu jaringan (Kusumadewi, 2004).

**4. Pelatihan *Recurrent Neural Network* Tipe Elman**

Pada penelitian ini menggunakan RNN tipe Elman dengan algoritma *backpropagation* atau propagasi balik sebagai algoritma pembelajarannya. Algoritma ini merupakan algoritma terawasi dan terdiri dari 3 proses yaitu perambatan maju (*forward propagation*) dan perambatan mundur (*backward*)

dan perubahan bobot. Perambatan maju dijalankan terlebih dahulu dengan mengaktifkan neuron-neuron menggunakan fungsi aktivasi yang didiferensialkan untuk memperoleh *error*. Kemudian proses perambatan mundur menghasilkan *error* yang merupakan selisih antara *output* jaringan dan target yang diharapkan. Tahap perambatan mundur diawali dari sinyal yang terkoneksi neuron di lapisan *output* kembali ke neuron lapisan *input.* Terakhir yaitu fase perubahan bobot guna memperbaiki *error* (Fausett, 1994).

Langkah-langkah pelatihan *backpropagation* pada RNN jaringan Elman selama kondisi $epoch < maximum\ epoch$ dan $MSE < target\ error$ adalah sebagai berikut.

a. Fase Perambatan Maju
    1) Setiap neuron *input* $(x_j, j = 1,2, \dots, p)$ menerima sinyal *input* $x_j$ dan melanjutkan sinyal ini ke setiap neuron pada lapisan berikutnya (lapisan tersembunyi). Setiap neuron yang diterima lapisan tersembunyi dari lapisan *input* dirumuskan dengan persamaan berikut.

$$z_{in(1)_k} = b_k + \sum_{j=1}^{p} x_j w_{jk(x)} \qquad (2.13)$$

Keterangan:

$b_k$ = bobot bias pada neuron ke-$k$ di lapisan tersembunyi, $k = 1,2, \ldots$

$x_j$ = variabel *input* dengan $j = 1,2, \ldots, p$

$w_{jk(x)}$ = bobot dari lapisan *input* ke-$j$ menuju neuron ke-$k$ di lapisan tersembunyi

Perhitungan sinyal *output* menggunakan fungsi aktivasi berikut.

$$z(1)_k = f(z_{in_k}) \qquad (2.14)$$

Kemudian sinyal tersebut disebar ke setiap neuron di lapisan *output* dan neuron tambahan. Proses ini dikerjakan sesuai jumlah lapisan tersembunyi.

2) Sinyal yang diterima setiap neuron tambahan $(u_k, k = 1,2, \ldots, q)$ diteruskan kembali ke neuron lapisan tersembunyi. Sinyal yang diterima neuron tambahan yaitu sebagai berikut.

$$u_k = z(1)_k \qquad (2.15)$$

Kemudian neuron tambahan mengirimkan ke neuron lapisan tersembunyi dengan persamaan berikut.

$$z_{in(2)_k} = \sum_{k=1}^{q} u_k w_{kk(u)} \qquad (2.16)$$

Keterangan:

$z_{in(2)_k}$= sinyal yang diterima lapisan tersembunyi dari neuron tambahan

$u_k$ = sinyal *output* (telah diaktivasi) yang diterima *hidden layer* dari neuron input, $k = 1,2, \dots, q$

$w_{kk(u)}$= bobot dari lapisan *input* (neuron tambahan) ke-$k$ menuju neuron ke- $k$ di lapisan tersembunyi

3) Setiap neuron pada lapisan tersembunyi $(z_k, k = 1,2, \dots, q)$ menjumlahkan setiap sinyal dari lapisan *input* dan neuron tambahan sebagai berikut.

$$z_{in_k} = z_{in(1)_k} + z_{in(2)_k} \qquad (2.17)$$

$$z_{in_k} = b_k + \sum_{j=1}^{p} x_j w_{jk(x)} + \sum_{k=1}^{q} u_k w_{kk(u)} \qquad (2.18)$$

Digunakan fungsi aktivasi untuk menghitung sinyal *output*nya.

$$z_k = f\left(z_{in_k}\right) \qquad (2.19)$$

Kemudian kirimkan sinyal tersebut ke semua neuron pada lapisan *output*.

4) Setiap neuron *output* (Y) menjumlahkan sinyal-sinyal *input* yang berbobot dari lapisan tersembunyi menggunakan persamaan berikut.

$$y_{in} = b_0 + \sum_{k=1}^{q} v_k z_k \qquad (2.20)$$

Keterangan:

$b_0$ = bias pada neuron *output*,

$v_k$ = bobot dari neuron ke-$k$ pada lapisan tersembunyi menuju lapisan *output*, $k = 1,2, \dots, q$,

$z_k$ = sinyal *output* yang telah diaktivasi dari lapisan tersembunyi, $k = 1,2, \dots q$.

Perhitungan sinyal *output* menggunakan fungsi aktivasi berikut.

$$y = f(y_{in}) \qquad (2.21)$$

b. Fase Perambatan Mundur

1) Setiap neuron *output* (y) menerima target pola yang sesuai dengan pola *input* pembelajaran, selanjutnya menghitung informasi errornya:

$$\delta_0 = (t_k - y_k) f'(y_{in}) \qquad (2.22)$$

Kemudian hitung koreksi bobot (dilakukan untuk memperbaiki nilai $v_k$) dengan persamaan sebagai berikut.

$$\Delta v_k = \propto \delta_0 y_k \quad (2.23)$$

dengan $\propto$ = laju percepatan

Selain itu, hitung koreksi bias (dilakukan untuk memperbaiki nilai $b_0$) dengan persamaan berikut.

$$\Delta b_0 = \propto \delta_0 \quad (2.24)$$

Kemudian kirimkan $\delta_0$ ke neuron-neuron pada lapisan tersembunyi.

2) Setiap neuron tersembunyi $(z_k, k = 1,2, \dots, q)$ menjumlahkan delta *input*nya dari setiap neuron pada lapisan di atasnya dengan persamaan berikut.

$$\delta_{in_k} = \sum_{k=1}^{q} \delta_0 v_k \quad (2.25)$$

Kalikan dengan turunan dari fungsi aktivasinya untuk menghitung *error*.

$$\delta_k = \delta_{in_k} f'(z_{in_k}) \quad (2.26)$$

Hitung koreksi bobot yang nanti digunakan untuk memperbaiki nilai $w_{jk(x)}$ dan $w_{kk(u)}$ dengan persamaan berikut.

$$\Delta w_{jk(x)} = \propto \delta_k x_j \quad (2.27)$$

$$\Delta w_{kk(u)} = \propto \delta_k u_k \quad (2.28)$$

Selanjutnya menghitung koreksi bias yang nanti digunakan untuk meurunkan nilai $b_k$.

$$\Delta b_k = \propto \delta_k \quad (2.29)$$

c. Perubahan Bobot

1) Setiap neuron *output* $(y_k)$ memperbaiki bias dan bobotnya dengan persamaan berikut.

$$v_k(baru) = v_k(lama) + \Delta v_k \quad (2.30)$$

$$b_0(baru) = b_0(lama) + \Delta b_0 \quad (2.31)$$

2) Setiap neuron lapisan tersembunyi $(z_k, k = 1,2, \dots, , q)$ memperbaiki bias dan bobotnya dengan persamaan berikut.

$$w_{jk(x)}(baru) = w_{jk(x)}(lama) + \Delta w_{jk(x)} \quad (2.32)$$

$$w_{kk(u)}(baru) = w_{kk(u)}(lama) + \Delta w_{kk(u)} \quad (2.33)$$

$$b_k(baru) = b_k(lama) + \Delta b_k \quad (2.34)$$

## G. Metode Evaluasi Peramalan

Tingkat akurasi pada suatu prediksi dengan model *time series* dilakukan karena ada 2 alasan utama. Pertama, ukuran akurasi digunakan dalam membandingkan beberapa model alternatif dan untuk memutuskan nilai parameter dalam fungsi prediksi. Setiap model dianggap diterapkan pada data masa lalu dalam mengidentifikasi model prediksi yang paling akurat dan memilih nilai *error* terendah atau minimum pada setiap model. Kedua, menilai keakuratan perlu dilakukan secara berkala untuk mendeteksi kelemahan dalam model yang mungkin muncul

di lain waktu sehingga dapat ditentukan model masih akurat atau membutuhkan perbaikan (Varcellis, 2009).

Beberapa rumus yang dapat digunakan dalam menentukan standar perbedaan (*error*) menurut Nasution dan Arman Hakim (2008) sehingga dapat menentukan metode yang paling akurat antara lain, *Mean Absolute Deviation* (MAD), *Mean Forecast Error* (MFE), *Mean Square Error* (MSE), *Mean Absolute Presentation Error* (MAPE) dan *Comulative Forecast Error* (CFE). Selain itu, ada *Root Mean Square Error* (RMSE) dan *Mean Absolute Error* (MAE) untuk menghitung kesalahan peramalan (Muharrom, 2023). Dalam penelitian ini, ukuran kesalahan yang digunakan adalah sebagai berikut (Hermawan, 2014).

1. *Mean Squared Error* (MSE)

   *Mean Squared Error* (MSE) adalah metode perhitungan dengan mengkuadratkan nilai *error* untuk setiap periode. MSE digunakan untuk mengukur kesalahan nilai prediksi model yang ditunjukkan dalam *mean* dari kuadrat kesalahan. Berikut adalah persamaan untuk menentukan nilai MSE (Hendikawati, 2015).

$$MSE = \frac{\sum_{t=1}^{n}(Y_t - \hat{Y}_t)^2}{n} \qquad (2.35)$$

   Keterangan:

   $Y_t$ = nilai pengamatan pada periode ke-t

$\hat{Y}_t$ = nilai peramalan pada periode ke-t

$n$ = banyak pengamatan

2. *Mean Absolute Percentage Error* (MAPE)

*Mean Absolute Percentage Error* (MAPE) adalah nilai absolut dari persentase *error* terhadap *mean*. MAPE digunakan untuk mengukur kesalahan nilai prediksi model yang ditunjukkkan dalam bentuk persentase *mean* absolut kesalahan. Berikut adalah persamaan untuk menentukan nilai MAPE (Hendikawati, 2015).

$$MAPE = 100\% \frac{\sum_{t=1}^{n} \left| \frac{Y_t - \hat{Y}_t}{Y_t} \right|}{n} \qquad (2.36)$$

Keterangan:

$Y_t$ = nilai pengamatan pada periode ke-t

$\hat{Y}_t$ = nilai peramalan pada periode ke-t

$n$ = banyak pengamatan

Perhitungan nilai MAPE digunakan untuk mencari akurasi. Berikut adalah rumus untuk menentukan nilai akurasi.

$$Akurasi = 100\% - MAPE \qquad (2.37)$$

MAPE memiliki retang nilai yang dapat dijadikan kriteria dalam membuat keputusan yang menunjukkan suatu prediksi termasuk dalam kategori baik atau tidak baik. Rentang nilai MAPE yang

dihasilkan adalah sebagai berikut (Moreno *et al*., 2013).

Tabel 2.2 Kriteria rentang nilai MAPE

| Rentang MAPE | Kategori |
|---|---|
| MAPE $< 10\%$ | Peramalan sangat baik |
| $10\% \leq$ MAPE $< 20\%$ | Peramalan baik |
| $20\% \leq$ MAPE $< 50\%$ | Peramalan cukup baik |
| MAPE $\geq 50\%$ | Peramalan buruk |

## H. Denormalisasi Data

Denormalisasi data adalah proses untuk pengembalian data hasil normalisasi dari jaringan yang bernilai antara 0,1 sampai 0,9 menjadi data yang asli. Berikut adalah rumus denormalisasi data (Arofah *et al*., 2020).

$$x_i = \frac{(x_i' - 0{,}1)(x_{max} - x_{min})}{0{,}8} + x_{min} \qquad (2.38)$$

Keterangan:

$x_i$ = hasil denormalisasi data x pada indeks ke-i

0,8 = nilai default denormalisasi nilai optimum

$x_i'$ = data yang akan didenormalisasi

$x_{min}$ = nilai data terendah

$x_{max}$ = nilai data tertinggi

0,1 = nilai default denormalisasi nilai minimum

## I. Kajian Pustaka Terdahulu

Dalam proses penelitian tugas akhir ini telah digunakan beberapa penelitian sebelumnya untuk dijadikan referensi dan acuan dalam melakukan penulisan dan penelitian. Berikut ini beberapa penelitian terdahulu yang berhubungan dengan masalah yang akan diselesaikan oleh peneliti dari berbagai sumber.

1. Penelitian yang dilakukan oleh Maulidya Ayu Putri (2021) yang berjudul "Perbandingan Model *Extreme Learning Machine* dan *Multilayer Perceptron* Dalam Peramalan Saham Nippon Paint (Studi Kasus: Harga Penutupan Saham Bulanan Nippon Paint dari Tahun 2016-2021)". Data sampel yang digunakan dalam penelitian ini adalah data harga penutupan saham harian Nipoon Paint Holdings Co, periode Januari 2016 hingga Oktober 2021, dengan jumlah data yang digunakan dalam penelitian ini adalah 70 data. Berdasarkan nilai MAPE yang dihasilkan bahwa metode *Multilayer Perceptron* lebih tepat digunakan karena menghasilkan nilai terkecil sebesar 4,46%. Penelitian tersebut dengan penelitian yang dilakukan penulis sama-sama menggunakan metode *Multilayer Perceptron*. Adapun yang membedakan dari penelitian tersebut adalah metode yang dibandingkan dengan *Multilayer Perceptron* dan pemilihan objek penelitian.

2. Penelitian yang dilakukan oleh Anifat Olawoyin dan Yangjuin Chen (2018) yang berjudul "*Predicting the Future with Artificial Neural Network*". Data yang digunakan adalah data transaksi pelanggaran parkir yang diperbarui oleh Pemerintah Kota Winnipeg setiap bulannya pada *website* data pemerintah. Berdasarkan analisis yang dilakukan, didapatkan bahwa metode *Multilayer Perceptron* yang menggunakan fungsi aktivasi sigmoid bipolar untuk lapisan tersembunyi dan pada lapisan *output* menggunakan fungsi linear menghasilkan nilai RMSE terendah yaitu 0,099. Sedangkan metode ARIMA menghasilkan nilai RMSE yaitu 0,145. Dari hasil tersebut menunjukkan bahwa *Multilayer Perceptron* memperoleh nilai yang lebih baik dibandingkan metode ARIMA. Penelitian tersebut dengan penelitian yang dilakukan penulis sama-sama menggunakan fungsi aktivasi sigmoid bipolar dan linear pada *Multilayer Perceptron*. Adapun yang membedakan dari penelitian tersebut adalah pemilihan objek penelitian.
3. Penelitian yang dilakukan oleh Eka Alifia Kusnanti *et al* (2022) yang berjudul "*Predicting Velocity and Direction of Ocean Surface Currents using Elman Recurrent Neural Network Method"*. Data yang digunakan adalah data sekunder yang diperoleh dari BMKG Maritim Tanjung Perak Surabaya berupa kecepatan arus permukaan laut

di Selat Bali selama tiga bulan. Penelitian tersebut menggunakan komposisi pembagian data yaitu 90% untuk data latih dan 10% untuk data uji. . Dilakukan 3 pelatihan dengan arsitektur ERNN yang berbeda yaitu, 5-20-50, 5-50-100, 5-80-150. Berdasarkan hasil dari penelitian, didapatkan bahwa model ERNN dengan arsitektur 5-20-50 merupakan arsitektur terbaik untuk prediksi dengan nilai MAPE paling sedikit pada pengujian yaitu 3,1253%. Penelitian tersebut dengan penelitian yang dilakukan penulis sama-sama menggunakan menggunakan metode ERNN dan neuron pada lapisan input sebanyak 5. Adapun yang membedakan dari penelitian tersebut adalah pemilihan objek penelitian.

4. Penelitian yang dilakukan oleh Nanang Hermawan (2014) yang berjudul "Aplikasi Model *Recurrent Neural Network* dan *Recurrent Neuro Fuzzy* untuk Peramalan Banyaknya Penumpang Kereta Api Jabodetabek". Penelitian tersebut menggunakan RNN tipe Elman dengan algoritma pembelajaran *backpropagation*. Data yang digunakan adalah data jumlah penumpang kereta api di wilayah Jabodetabek yang didapatkan dari *website* Badan Pusat Statistik (BPS) dari periode Januari 2006 hingga Agustus 2013 pada setiap bulan secara detail. Penelitian tersebut menggunakan komposisi pembagian

data yaitu 75% untuk data latih dan 25% untuk data uji. Berdasarkan analisis yang dilakukan, dipilih model terbaik yaitu *Recurrent Neural Network* karena pengujian yang dilakukan menghasilkan nilai MAPE sebesar 3,785%. Penelitian tersebut dengan penelitian yang dilakukan penulis sama-sama menggunakan metode RNN tipe Elman. Adapun yang membedakan dari penelitian tersebut adalah metode yang dibandingkan dengan RNN tipe Elman dan pemilihan objek penelitian.

5. Penelitian yang dilakukan oleh Jaka Pratama Musashi (2014) yang berjudul "Prediksi Harga *High* Nilai Tukar EUR-USD menggunakan *Recurrent Neural Network* (RNN)". Penelitian tersebut menggunakan RNN tipe Elman. Data yang digunakan dalam penelitian ini adalah data nilai tukar mata uang dan pengambilan data didapatkan dari *server* Alpari UK melalui aplikasi MetaTrader 5 (MT5) dengan periode mulai dari bulan Januari 2012 – Desember 2013. Berdasarkan hasil dari penelitian, diperoleh model terbaik dalam memprediksi dengan nilai MAPE sebesar 0,141% dan nilai koefisien determinasi sebesar 99,5%. Penelitian tersebut dengan penelitian yang dilakukan penulis sama-sama menggunakan metode RNN tipe Elman. Adapun yang membedakan dari penelitian tersebut adalah pemilihan objek penelitian.

6. Penelitian yang dilakukan oleh Khoirudin, Dewi Nurdiyah, dan Nur Wakidah (2018) yang berjudul "Prediksi Penerimaan Mahasiswa Baru dengan *Multilayer Perceptron*". Penelitian tersebut menggunakan fungsi aktivasi tansig atau sigmoid bipolar. Dalam penelitiannya, peneliti menggunakan dataset *time series* penerimaan mahasiswa baru dari periode 2008 sampai 2017. Penelitian tersebut menggunakan komposisi pembagian data yaitu 70% untuk data latih dan 30% untuk data uji. Dilakukan 3 pelatihan dengan arsitektur MLP yang berbeda yaitu, 5-9-1, 5-6-1, 5-12-1. Berdasarkan hasil dari penelitian, didapatkan bahwa model MLP dengan arsitektur 5-9-1 merupakan arsitektur terbaik untuk memprediksi dengan nilai MSE paling sedikit yaitu 0,1. Penelitian tersebut dengan penelitian yang dilakukan penulis sama-sama menggunakan metode *Multilayer Perceptron* dan neuron pada lapisan input sebanyak 5. Adapun yang membedakan dari penelitian tersebut adalah pemilihan objek penelitian.
7. Penelitian yang dilakukan oleh Muhammad Rifaldo, Harun Mukhtar, Reny Medikawati Taufiq, dan Yoze Rizki (2021) yang berjudul "Peramalan Kedatangan Wisatawan Mancanegara ke Indonesia menurut Kebangsaan Perbulannya menggunakan Metode

*Multilayer Perceptron"*. Dalam penelitiannya, peneliti menggunakan dataset *time series* kunjungan wisatawan dari bulan Januari 2019 – Desember 2020. Penelitian tersebut akan memprediksi data kujungan wisatawan bulan Januari 2021 dengan data x1 yaitu bulan November 2020 dan data x2 yaitu Desember 2020. Nilai MSE yang didapatkan mendekati nol yaitu sebesar 0,0003 sehingga dapat dikatakan prediksi yang diperoleh sudah akurat. Penelitian tersebut dengan penelitian yang dilakukan penulis sama-sama menggunakan metode *Multilayer Perceptron*. Adapun yang membedakan dari penelitian tersebut adalah pemilihan objek penelitian.

# BAB III
# METODE PENELITIAN

## A. Jenis Penelitian

Jenis penelitian yang digunakan dalam penelitian ini adalah penelitian terapan. Penelitian terapan (*applied research*) adalah suatu penelitian terhadap masalah tertentu dengan tujuan untuk memecahkan atau menemukan solusi dari masalah tersebut (Mulyatiningsih, 2011). Penelitian adalah suatu penyelidikan terstruktur atau penyelidikan secara cermat dan teliti untuk menetapkan sesuatu dalam menemukan kebenaran (Zakariah *et al*., 2020). Penelitian ini menganalisis data laju pertumbuhan penduduk Kabupaten Brebes tahun 1991-2022.

## B. Jenis dan Sumber Data

Pada penelitian ini, jenis data yang digunakan adalah data sekunder. Data sekunder adalah data yang didapatkan dari hasil penelitian yang sudah pernah dilakukan oleh para peneliti sebelumnya dan bukan dari pengamatan langsung (Marisya, 2020). Sumber data yang digunakan berupa data jumlah penduduk yang diperoleh dari *website* Badan Pusat Statistik (BPS) Kabupaten Brebes https://brebeskab.bps.go.id/indicator/12/32/1/jumlah-

penduduk-kab-brebes-menurut-kecamatan.html. Data yang didapatkan, disusun seperti pada Tabel 3.1.

**Tabel 3.1** Susunan Data Input

| Pola ke- | Data Input ($X_i$) | | | | | Target (Y) |
|---|---|---|---|---|---|---|
| | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | |
| 1 | Data tahun ke-1 | Data tahun ke-2 | Data tahun ke-3 | Data tahun ke-4 | Data tahun ke-5 | Data tahun ke-6 |
| 2 | Data tahun ke-2 | Data tahun ke-3 | Data tahun ke-4 | Data tahun ke-5 | Data tahun ke-6 | Data tahun ke-7 |
| 3 | Data tahun ke-3 | Data tahun ke-4 | Data tahun ke-5 | Data tahun ke-6 | Data tahun ke-7 | Data tahun ke-8 |
| ... | ... | ... | ... | ... | ... | ... |
| 27 | Data tahun ke-27 | Data tahun ke-28 | Data tahun ke-29 | Data tahun ke-30 | Data tahun ke-31 | Data tahun ke-32 |

## C. Metode Pengumpulan Data

Data yang digunakan berasal dari Badan Pusat Statistik berupa data jumlah penduduk Kabupaten Brebes tahun 1991-2022. Pada penelitian ini, metode pengumpulan data yang digunakan adalah dokumentasi. Metode dokumentasi adalah proses mengumpulkan data dengan cara teks-teks tertulis maupun *soft-copy edition* seperti buku, *e-book*, artikel, jurnal, publikasi pemerintah, dan lain-lain (Nurhadi *et al*., 2021).

## D. Metode Analisis Data

Penelitian ini membahas tentang prediksi laju pertumbuhan penduduk di Kabupaten Brebes menggunakan metode *Multilayer Perceptron* dan metode *Recurrent Neural Network*. Penelitian ini menggunakan alat bantu perangkat lunak yaitu Matlab. Berikut dijelaskan prosedur analisis prediksi menggunakan metode *Multilayer Perceptron* dan metode *Recurrent Neural Network*.

1. Pengumpulan Data

   Data jumlah penduduk tahun 1991-2022 yang sudah didapatkan dari *website* BPS dikumpulkan dalam tabel.
2. Penentuan Nilai Input

   Menetapkan nilai input dari data jumlah penduduk Kabupaten Brebes periode 1991-2022.
3. Normalisasi Data

   Proses normalisasi data dilakukan dengan menggunakan persamaan 2.7.
4. Pembagian Data

   Data dibagi menjadi 2 yaitu data pelatihan (*training*) dan data pengujian (*testing*).
5. Inisialisasi Parameter

   Menentukan parameter jaringan untuk memperoleh model terbaik.

6. Pelatihan Jaringan

   Tujuan dilakukan pelatihan jaringan agar nilai kesalahan dapat diminimalkan sehingga model jaringan dapat digunakan untuk prediksi.

7. Pengujian Jaringan

   Pengujian jaringan dengan menggunakan pemodelan yang didapatkan pada proses pelatihan jaringan yaitu arsitektur, parameter, fungsi aktivasi, dan bobot.

8. Denormalisasi Data

   Melakukan denormalisasi data yang bertujuan mengembalikan data normalisasi ke angka sebenarnya. Perhitungan denormalisasi data menggunakan persamaan 2.38.

9. Membandingkan data asli dan data prediksi tahun 1991-2022.

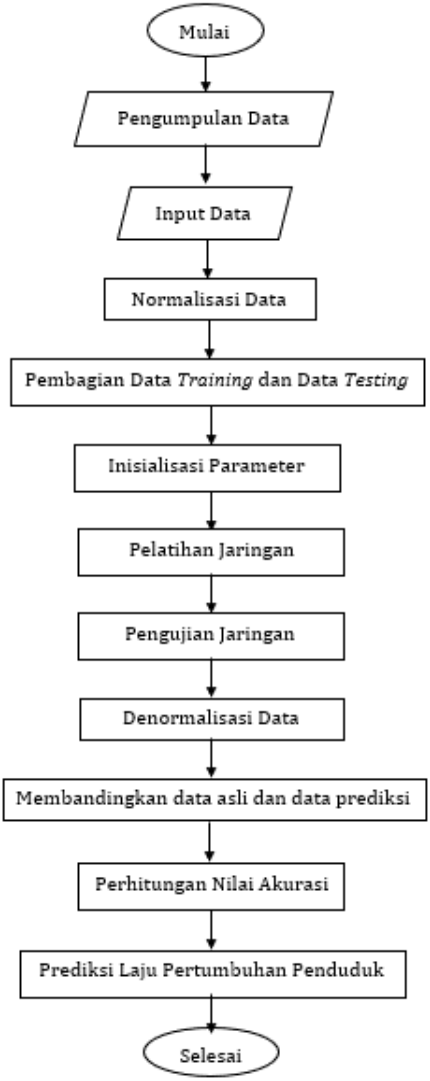
10. Perhitungan Nilai Akurasi

    Menghitung nilai akurasi untuk mengetahui model arsitektur menunjukkan kategori baik atau tidak baik. Nilai akurasi dihitung menggunakan persamaan 2.37.

11. Menghitung hasil prediksi laju pertumbuhan penduduk tahun 2023-2026 menggunakan persamaan 2.1.

12. Membandingkan nilai MAPE kedua metode untuk mendapatkan metode yang terbaik.

Berikut gambaran diagram alir analisis metode *Multilayer Perceptron* dan *Recurrent Neural Network*.




Gambar 3.1 Diagram Alir Metode *Multilayer Perceptron* dan *Recurrent Neural Network*

Berikut diagram alir menentukan metode yang terbaik.


Gambar 3.2 Diagram Alir Menentukan Metode Terbaik

# BAB IV
# HASIL DAN PEMBAHASAN

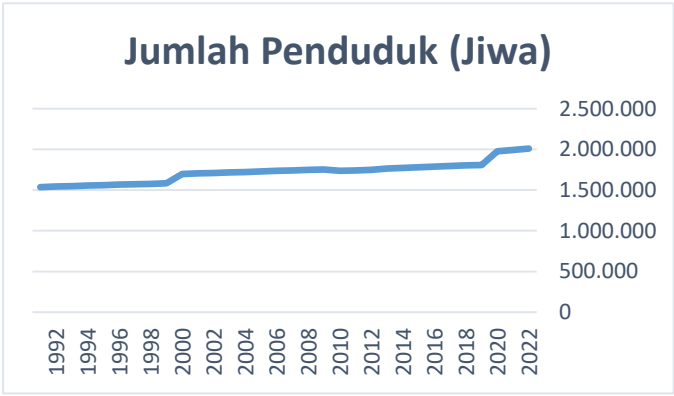
## A. Pengumpulan Data

Data yang digunakan pada Tugas Akhir ini adalah data jumlah penduduk kabupaten Brebes dari tahun 1991-2022. Berikut adalah data jumlah penduduk yang diambil dari website BPS.

Tabel 4.1 Data Jumlah Penduduk Kabupaten Brebes

| No | Tahun | Jumlah Penduduk (Jiwa) |
|---|---|---|
| 1 | 1991 | 1.536.534 |
| 2 | 1992 | 1.542.775 |
| 3 | 1993 | 1.548.928 |
| 4 | 1994 | 1.555.424 |
| 5 | 1995 | 1.561.329 |
| 6 | 1996 | 1.567.044 |
| 7 | 1997 | 1.572.878 |
| 8 | 1998 | 1.577.631 |
| 9 | 1999 | 1.583.426 |
| 10 | 2000 | 1.698.635 |
| 11 | 2001 | 1.705.433 |
| 12 | 2002 | 1.711.657 |
| 13 | 2003 | 1.717.103 |
| 14 | 2004 | 1.722.306 |
| 15 | 2005 | 1.727.708 |
| 16 | 2006 | 1.736.401 |
| 17 | 2007 | 1.743.195 |
| 18 | 2008 | 1.747.430 |
| 19 | 2009 | 1.752.128 |
| 20 | 2010 | 1.736.331 |
| 21 | 2011 | 1.742.511 |
| 22 | 2012 | 1.748.510 |
| 23 | 2013 | 1.764.648 |
| 24 | 2014 | 1.773.379 |

| No | Tahun | Jumlah Penduduk (Jiwa) |
|---|---|---|
| 25 | 2015 | 1.781.379 |
| 26 | 2016 | 1.788.880 |
| 27 | 2017 | 1.796.004 |
| 28 | 2018 | 1.802.829 |
| 29 | 2019 | 1.809.096 |
| 30 | 2020 | 1.978.759 |
| 31 | 2021 | 1.992.685 |
| 32 | 2022 | 2.010.617 |




Gambar 4.1 Grafik Jumlah Penduduk Kabupaten Brebes

Pada Gambar 4.1 jumlah penduduk Kabupaten Brebes menunjukkan bahwa pada tahun 1991 sampai dengan tahun 2022, Kabupaten Brebes memiliki penduduk yang cenderung meningkat pada setiap tahun. Peningkatan yang cukup tinggi terjadi pada peralihan tahun 2019 menuju tahun 2020.

## *B.* Metode *Multilayer Perceptron*

### a. Menetapkan Nilai Input

Pada penelitian ini, pola *time series* yang dibentuk sebanyak lima. Artinya data input yang digunakan berjumlah lima inputan dan output yang dicari adalah data setelah data kelima. Pembentukan pola tersebut merujuk pada penelitian dari Anggoro (2023). Pola *time series* yang digunakan yaitu sebagai berikut.

Tabel 4.2 Data *Time Series* Jumlah Penduduk

| No | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | Y |
|---|---|---|---|---|---|---|
| 1 | 1.536.534 | 1.542.775 | 1.548.928 | 1.555.424 | 1.561.329 | 1.567.044 |
| 2 | 1.542.775 | 1.548.928 | 1.555.424 | 1.561.329 | 1.567.044 | 1.572.878 |
| 3 | 1.548.928 | 1.555.424 | 1.561.329 | 1.567.044 | 1.572.878 | 1.577.631 |
| 4 | 1.555.424 | 1.561.329 | 1.567.044 | 1.572.878 | 1.577.631 | 1.583.426 |
| 5 | 1.561.329 | 1.567.044 | 1.572.878 | 1.577.631 | 1.583.426 | 1.698.635 |
| … | … | … | … | … | … | … |
| 27 | 1.796.004 | 1.802.829 | 1.809.096 | 1.978.759 | 1.992.685 | 2.010.617 |

Secara terperinci, data *time series* jumlah penduduk tertera pada Lampiran 2.

### b. Normalisasi Data

Data jumlah penduduk Kabupaten Brebes dilakukan normalisasi data terlebih dahulu. Nilai

mininal data dan nilai maksimal data berdasarkan data pada Tabel 4.2 adalah 1.536.534 dan 2.010.617. Normalisasi dilakukan dengan menggunakan persamaan 2.7 maka akan didapat nilai berikut.

$$x_1' = \frac{0{,}8(1536534 - 1536534)}{(2010617 - 1536534)} + 0{,}1 = 0{,}1$$

$$x_2' = \frac{0{,}8(1542775 - 1536534)}{(2010617 - 1536534)} + 0{,}1 = 0{,}110531$$

$$x_3' = \frac{0{,}8(1548928 - 1536534)}{(2010617 - 1536534)} + 0{,}1 = 0{,}120914$$

$$x_4' = \frac{0{,}8(1555424 - 1536534)}{(2010617 - 1536534)} + 0{,}1 = 0{,}131876$$

Semua data dilakukan perhitungan normalisasi. Hasil perhitungan normalisasi tersaji pada Tabel 4.3 berikut.

Tabel 4.3 Normalisasi Data

| No | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | Y |
|---|---|---|---|---|---|---|
| 1 | 0,1 | 0,11053 | 0,12091 | 0,13188 | 0,14184 | 0,15148 |
| 2 | 0,11053 | 0,12091 | 0,13188 | 0,14184 | 0,15148 | 0,16133 |
| 3 | 0,12091 | 0,13188 | 0,14184 | 0,15148 | 0,16133 | 0,16935 |
| 4 | 0,13188 | 0,14184 | 0,15148 | 0,16133 | 0,16935 | 0,17913 |
| 5 | 0,14184 | 0,15148 | 0,16133 | 0,16935 | 0,17913 | 0,37354 |
| ... | ... | ... | ... | ... | ... | ... |
| 27 | 0,53785 | 0,54936 | 0,55994 | 0,84624 | 0,86974 | 0,9 |

Hasil normalisasi data pada Tabel 4.3 akan digunakan dalam proses *training* dan *testing*. Data lengkap hasil perhitungan normalisasi data dapat dilihat pada Lampiran 3.

**c. Pembagian Data *Training* dan *Testing***

Pada bagian ini, data akan dibagi menjadi dua bagian. Data pertama digunakan untuk membentuk pola dan data kedua digunakan dalam pengujian. Hasil dari normalisasi data pada Tabel 4.3 dibagi menjadi data *training* dan data *testing*. Berikut adalah hasil presentase pembagian data yang digunakan untuk penelitian.

Tabel 4.4 Pembagian Data MLP

| Pembagian Data | Presentase | Data |
|---|---|---|
| *Training* | 80% | 22 |
| *Testing* | 20% | 5 |
| Total | 100% | 27 |

Perbandingan pembagian data tersebut merujuk pada penelitian dari Hardiyanti (2021) yang menunjukkan bahwa perbandingan data *training* dan data *testing* yaitu 80% dan 20% menghasilkan nilai MSE terendah dibandingkan perbandingan data yang lain.

**d. Inisialisasi Parameter**

Sebelum melakukan proses *training,* maka harus menentukan parameter yang bertujuan membentuk model jaringan. Proses pemodelan jaringan pada Matlab yaitu dengan memilih *new* kemudian membuat jaringan baru. Parameter penelitian yang digunakan adalah sebagai berikut.

1) Jumlah neuron lapisan *input* adalah 5 yaitu $X_1$, $X_2, X_3, X_4, X_5$.
2) Jumlah neuron lapisan tersembunyi adalah 5, 6, 7, 8, 9, dan 10. Jumlah neuron berfungsi untuk membentuk model arsitektur jaringan dan menentukan nilai MSE yang paling kecil. Menurut Siang (2009), bahwa tidak ada teori yang pasti dapat dipakai dalam memilih jumlah neuron tersembunyi. Oleh karena itu, jumlah neuron yang digunakan adalah 5, 6, 7, 8, 9, dan 10 yang bertujuan untuk mendapatkan model arsitektur terbaik berdasarkan nilai MSE terkecil.
3) Jumlah neuron pada lapisan *output* adalah 1 yaitu Y.
4) Fungsi pelatihan (*training function*) yang digunakan adalah *Levenberg-Marquest* (LM). Fungsi pelatihan tersebut merujuk pada penelitian dari Houari (2015) yang menunjukkan

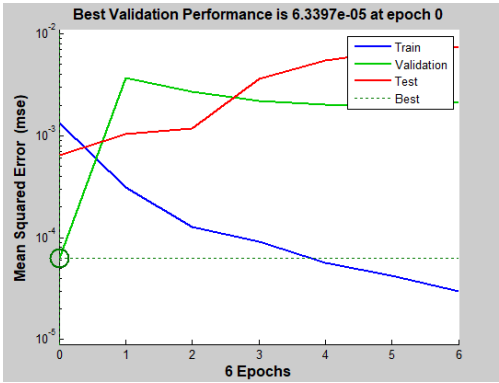
bahwa *trainlm* adakah fungsi pelatihan yang mempunyai kapasitas terbaik untuk memprediksi.

5) Fungsi Pembelajaran (*adaption learning function*) yang digunakan adalah *learngdm*. Fungsi pembelajaran tersebut merujuk pada penelitian dari Deltania (2023) yang menunjukkan bahwa *learngdm* mendapatkan hasil terbaik.
6) Fungsi transfer (*transfer function*) yang digunakan adalah sigmoid bipolar (*tansig*) dan linear (*purelin*). Fungsi transfer tersebut merujuk pada penelitian dari Houari (2015) yang menunjukkan bahwa nilai MSE terkecil dihasilkan dari kombinasi kedua fungsi tersebut.
7) Maksimal *epoch* adalah 1000. Jumlah *epoch* tersebut merujuk pada penelitian dari Santoso (2023) yang menunjukkan bahwa jumlah 1000 mendapatkan nilai MSE terbaik. Proses *training* setiap model dilakukan sebanyak 20 kali karena untuk setiap *training* data, bobot dapat berubah-ubah (Mustafidah, 2019).

**e. Pelatihan Jaringan**

1. Arsitektur 5-5-1

*Training* model arsitektur 5-5-1 artinya terdapat 5 neuron di lapisan *input* ($X_1$, $X_2, X_3, X_4, X_5$) ditambah sebuah bias, 5 neuron di lapisan tersembunyi ($z_1, z_2, z_3, z_4, z_5$) ditambah sebuah bias, 1 neuron di lapisan *output* (Y).

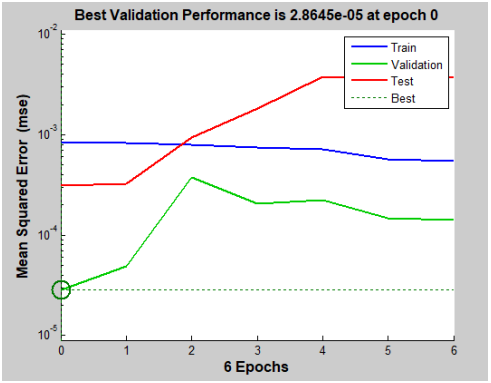



Gambar 4.2 *Plot* Performa MLP Arsitektur 5-5-1

Dari gambar 4.2 terlihat bahwa garis *train* dengan garis *validation* dan garis *test* tidak sejalan sehingga validasi terbaik didapat saat *epoch* ke 0.

2. Arsitektur 5-6-1

*Training* model arsitektur 5-6-1 artinya terdapat 5 neuron di lapisan *input* ($X_1$, $X_2, X_3, X_4, X_5$) ditambah sebuah bias, 6 neuron di lapisan tersembunyi ($z_1$, $z_2, z_3, z_4, z_5, z_6$) ditambah sebuah bias, 1 neuron di lapisan *output* (Y).

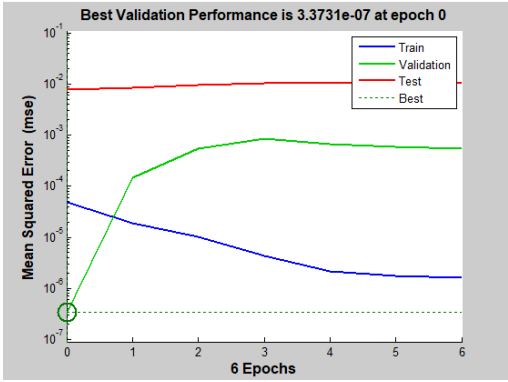



Gambar 4.3 *Plot* Performa MLP Arsitektur 5-6-1

Dari gambar 4.3 menunjukkan bahwa hanya garis *train* saja yang stabil. *Epoch* terbaik terjadi saat *epoch* ke 0.

3. Arsitektur 5-7-1

*Training* model arsitektur 5-7-1 artinya terdapat 5 neuron di lapisan *input* ($X_1$, $X_2, X_3, X_4, X_5$) ditambah bias, 7 neuron di lapisan tersembunyi ($z_1, z_2, z_3, z_4, z_5, z_6, z_7$) ditambah sebuah bias, 1 neuron di lapisan *output* (Y).

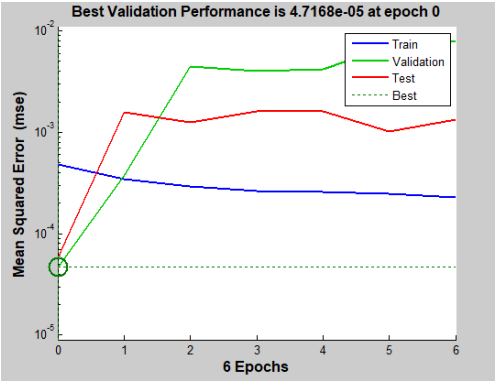



Gambar 4.4 *Plot* Performa MLP Arsitektur 5-7-1

Dari gambar 4.4 menunjukkan bahwa hanya garis *test* saja yang stabil. *Epoch* terbaik terjadi saat *epoch* ke 0.

4. Arsitektur 5-8-1

*Training* model arsitektur 5-8-1 artinya terdapat 5 neuron di lapisan *input* ($X_1$, $X_2, X_3, X_4, X_5$) ditambah sebuah bias, 8 neuron di lapisan tersembunyi ($z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8$) ditambah sebuah bias, 1 neuron di lapisan *output* (Y).

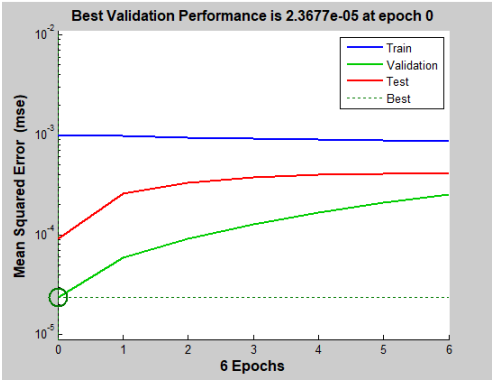



Gambar 4.5 *Plot* Performa MLP Arsitektur 5-8-1

Plot performa dari gambar 4.5 menunjukkan bahwa garis *validation* dan garis *test* memiliki pola yang sama.

5. Arsitektur 5-9-1

   *Training* model arsitektur 5-9-1 artinya terdapat 5 neuron di lapisan *input* ($X_1$, $X_2, X_3, X_4, X_5$) ditambah sebuah bias, 9 neuron di 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 di lapisan *output* (Y).

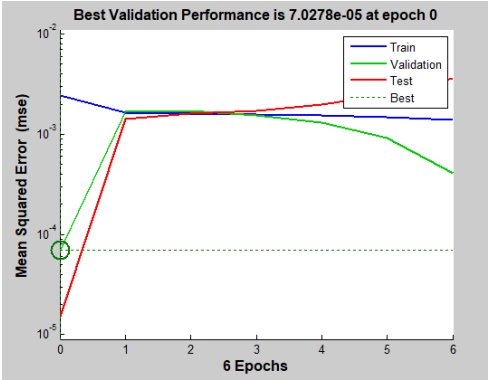



Gambar 4.6 *Plot* Performa MLP Arsitektur 5-9-1

*Plot* performa dari gambar 4.6 menunjukkan bahwa ketiga garis berjalan semakin mendekati satu sama lain.

6. Arsitektur 5-10-1

*Training* model arsitektur 5-10-1 artinya terdapat 5 neuron di lapisan *input* ($X_1$, $X_2, X_3, X_4, X_5$) ditambah bias, 10 neuron di 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 di lapisan *output* (Y).




Gambar 4.7 *Plot* Performa MLP Arsitektur 5-10-1

Dari gambar 4.7 menunjukkan bahwa hanya garis *train* saja yang stabil setelah *epoch* ke 1.

**f. Pengujian Jaringan**

Setelah melakukan proses *training*, maka dilakukan pengujian untuk semua model. Berikut adalah tabel hasil simulasi pengujian pada 6 model.

Tabel 4.5 Nilai *Output* (Y) Hasil *Testing* MLP

| Tahun | Target | Model | | | | | |
|---|---|---|---|---|---|---|---|
| | | 5-5-1 | 5-6-1 | 5-7-1 | 5-8-1 | 5-9-1 | 5-10-1 |
| 2018 | 0,54936 | 0,58578 | 0,54806 | 0,57927 | 0,57043 | 0,57164 | 0,55903 |
| 2019 | 0,55994 | 0,61441 | 0,55362 | 0,61918 | 0,5851 | 0,59966 | 0,5743 |
| 2020 | 0,84624 | 0,63927 | 0,55646 | 0,66062 | 0,59416 | 0,62784 | 0,58621 |
| 2021 | 0,86974 | 0,77802 | 0,59401 | 0,71935 | 0,59036 | 0,86161 | 0,53647 |
| 2022 | 0,9 | 0,78233 | 0,59327 | 0,61146 | 0,83949 | 0,97234 | 0,5676 |

Nilai *error* MSE untuk 6 model disajikan pada tabel 4.6 berikut.

Tabel 4.6 Nilai *Error* MSE *Testing* MLP

| **Model** | ***Error* MSE** |
|---|---|
| 5-5-1 | 0,013877753 |
| 5-6-1 | 0,050824881 |
| 5-7-1 | 0,028946251 |
| 5-8-1 | 0,029267188 |
| 5-9-1 | 0,011014362 |
| 5-10-1 | 0,057894794 |

Berdasarkan tabel 4.6 didapatkan model arsitektur terbaik yaitu model 5-9-1. Model 5-9-1 menjadi model terbaik karena MSE yang dihasilkan mempunyai nilai terkecil dari model lainnya yaitu 0,011014362. Data lengkap hasil pengujian MLP dapat dilihat pada Lampiran 7.

**g. Denormalisasi Data**

Pada Tabel 4.5 dapat dilihat nilai *output* dari model 5-9-1 yang menjadi model terbaik masih dalam bentuk normalisasi. Oleh karena itu, data akan diubah ke dalam nilai yang sebenarnya yang disebut denormalisasi. Proses denormalisasi menggunakan persamaan 2.38, didapatkan nilai sebagai berikut.

$$x_{2018} = \frac{(0{,}57164 - 0{,}1)(2010617 - 1536534)}{0{,}8} + 1536534 = 1816030$$

$$x_{2019} = \frac{(0{,}59966 - 0{,}1)(2010617 - 1536534)}{0{,}8} + 1536534 = 1832634$$

$$x_{2020} = \frac{(0{,}62784 - 0{,}1)(2010617 - 1536534)}{0{,}8} + 1536534 = 1849334$$

$$x_{2021} = \frac{(0{,}86161 - 0{,}1)(2010617 - 1536534)}{0{,}8} + 1536534 = 1987867$$

$$x_{2022} = \frac{(0{,}97234 - 0{,}1)(2010617 - 1536534)}{0{,}8} + 1536534 = 2053486$$

Hasil denormalisasi *output* merupakan hasil prediksi dari tahun 2018-2022. Berikut adalah tabel hasil prediksi jumlah penduduk tahun 2018-2022.

Tabel 4.7 Hasil Prediksi Jumlah Penduduk Tahun 2018-2022 Metode MLP

| Tahun | Jumlah Penduduk |
|---|---|
| 2018 | 1816030 |
| 2019 | 1832634 |
| 2020 | 1849334 |
| 2021 | 1987867 |
| 2022 | 2053486 |

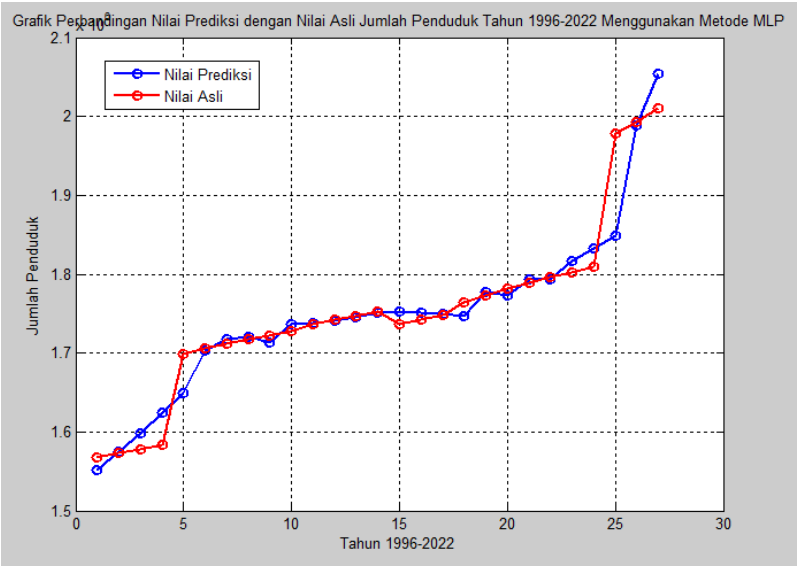
**h. Perbandingan Data Asli dan Data Prediksi Tahun 1991-2022**

Berikut adalah tabel hasil perbandingan data asli dan data prediksi pada proses pelatihan dan pengujian.

Tabel 4.8 Hasil Perbandingan Data Asli dan Data Prediksi Metode MLP

| Tahun | Normalisasi Data | | Denormalisasi Data | |
|---|---|---|---|---|
| | Nilai Asli | Nilai Prediksi | Nilai Asli | Nilai Prediksi |
| 1991 | 0,15148 | - | 1.536.534 | - |
| 1992 | 0,16133 | - | 1.542.775 | - |
| 1993 | 0,16935 | - | 1.548.928 | - |
| 1994 | 0,17913 | - | 1.555.424 | - |
| 1995 | 0,37354 | - | 1.561.329 | - |
| 1996 | 0,38501 | 0,12672 | 1.567.044 | 1552368 |
| 1997 | 0,39551 | 0,16467 | 1.572.878 | 1574858 |
| 1998 | 0,4047 | 0,20471 | 1.577.631 | 1598586 |
| 1999 | 0,41348 | 0,24747 | 1.583.426 | 1623925 |
| 2000 | 0,4226 | 0,28937 | 1.698.635 | 1648755 |
| 2001 | 0,43727 | 0,3822 | 1.705.433 | 1703767 |
| 2002 | 0,44873 | 0,40495 | 1.711.657 | 1717249 |
| 2003 | 0,45588 | 0,41093 | 1.717.103 | 1720792 |
| 2004 | 0,46381 | 0,39719 | 1.722.306 | 1712650 |
| 2005 | 0,43715 | 0,43652 | 1.727.708 | 1735957 |
| 2006 | 0,44758 | 0,43978 | 1.736.401 | 1737889 |
| 2007 | 0,4577 | 0,44563 | 1.743.195 | 1741356 |
| 2008 | 0,48494 | 0,45339 | 1.747.430 | 1745954 |
| 2009 | 0,49967 | 0,4622 | 1.752.128 | 1751175 |
| 2010 | 0,51317 | 0,46523 | 1.736.331 | 1752971 |
| 2011 | 0,52583 | 0,46109 | 1.742.511 | 1750517 |
| 2012 | 0,53785 | 0,4598 | 1.748.510 | 1749753 |
| 2013 | 0,15148 | 0,45436 | 1.764.648 | 1746529 |
| 2014 | 0,16133 | 0,50756 | 1.773.379 | 1778056 |
| 2015 | 0,16935 | 0,49999 | 1.781.379 | 1773570 |
| 2016 | 0,17913 | 0,53294 | 1.788.880 | 1793096 |
| 2017 | 0,37354 | 0,5338 | 1.796.004 | 1793606 |
| 2018 | 0,54936 | 0,57164 | 1.802.829 | 1816030 |
| 2019 | 0,55994 | 0,59966 | 1.809.096 | 1832634 |
| 2020 | 0,84624 | 0,62784 | 1.978.759 | 1849334 |
| 2021 | 0,86974 | 0,86161 | 1.992.685 | 1987867 |
| 2022 | 0,9 | 0,97234 | 2.010.617 | 2053486 |

Grafik perbandingan nilai prediksi dan nilai aktual menggunakan metode MLP pada tabel 4.8 dapat digambarkan sebagai berikut.




Gambar 4.8 *Plot* Perbandingan Data Prediksi dan Data Asli Metode MLP

Pada gambar 4.8 dapat diperhatikan bahwa garis *plot* antara nilai prediksi dengan nilai aktual tidak berbeda jauh. Dapat disimpulkan bahwa hasil prediksi jumlah penduduk Kabupaten Brebes periode 1996-2022 menggunakan metode MLP mendekati nilai sebenarnya.

**i. Perhitungan Nilai Akurasi**

Setelah dilakukan proses *testing* pada data tahun 2018-2022, dilanjutkan proses evaluasi terhadap data sebenarnya tahun 2018-2022. Tujuan dilakukan evaluasi adalah untuk mengukur kemampuan model dalam memprediksi. Metode evaluasi yang digunakan yaitu MAPE. Berikut tabel nilai MAPE hasil peramalan.

Tabel 4.9 Hasil Nilai MAPE MLP

| Tahun | Jumlah Penduduk | | *Absolute Percentage Error* |
|---|---|---|---|
| | Hasil Prediksi | Data Asli | |
| 2018 | 1816030 | 1802829 | 0,732238 |
| 2019 | 1832634 | 1809096 | 1,301092 |
| 2020 | 1849334 | 1978759 | 6,540716 |
| 2021 | 1987867 | 1992685 | 0,241784 |
| 2022 | 2053486 | 2010617 | 2,132132 |
| MAPE | | | 2,189592 |

Berdasarkan Tabel 4.9, ditunjukkan bahwa kategori nilai MAPE hasil prediksi metode *Multilayer Perceptron* adalah sangat baik karena menghasilkan nilai kurang dari 10%.

**j. Hasil Prediksi Laju Pertumbuhan Penduduk Tahun 2023-2026**

Selanjutnya, akan dilakukan prediksi laju pertumbuhan penduduk Kabupaten Brebes untuk tahun 2023-2026. Untuk memprediksi jumlah

penduduk tahun 2023 adalah dengan perintah "sim" sebagai berikut.

prediksi_2023 = sim(network5, hasil_testing)

Keterangan:

*network*5 : jaringan MLP model 5-9-1

hasil_testing : nilai *output* pada proses pengujian sebelumnya

Untuk memprediksi jumlah penduduk tahun 2024 adalah dengan cara seperti tahun 2023 yaitu dengan menambahkan prediksi jumlah penduduk terakhir (tahun 2023) sebagai *input*nya. Perhitungan secara lengkap terdapat di Lampiran 8. Nilai tersebut harus dilakukan denormalisasi agar kembali ke bentuk asli dengan persamaan 2.38.

Berikut hasil prediksi jumlah penduduk Kabupaten Brebes tahun 2023-2026 setelah dilakukan denormalisasi.

Tabel 4.10 Hasil Prediksi Jumlah Penduduk Metode MLP

| Tahun | Hasil Prediksi |
|---|---|
| 2023 | 2062478 |
| 2024 | 2073673 |
| 2025 | 2058963 |
| 2026 | 2069341 |

Berdasarkan Tabel 4.10 dapat dihitung laju pertumbuhan penduduk dari tahun 2023-2026. Jumlah

penduduk dasar adalah pada tahun 2022 yaitu 2.010.617. Berikut nilai yang diperoleh dari perhitungan laju pertumbuhan penduduk menggunakan persamaan 2.1.

$$\text{Tahun } 2023: r = \left(\frac{2062478}{2010617}\right)^{\frac{1}{1}} - 1 = 0{,}02579357$$

$$\text{Tahun } 2024: r = \left(\frac{2073673}{2010617}\right)^{\frac{1}{2}} - 1 = 0{,}0155597$$

$$\text{Tahun } 2025: r = \left(\frac{2058963}{2010617}\right)^{\frac{1}{3}} - 1 = 0{,}0079517$$

$$\text{Tahun } 2026: r = \left(\frac{2069341}{2010617}\right)^{\frac{1}{4}} - 1 = 0{,}0072231$$

Berikut adalah tabel hasil prediksi laju pertumbuhan penduduk Kabupaten Brebes tahun 2023-2026.

Tabel 4.11 Hasil Prediksi Laju pertumbuhan Penduduk Metode MLP

| Tahun | Hasil Prediksi |
|---|---|
| 2023 | 2,579% |
| 2024 | 1,556% |
| 2025 | 0,795% |
| 2026 | 0,722% |

Berdasarkan Tabel 4.11 dapat diketahui bahwa pada tahun 2023, 2024, 2025, dan 2026, nilai prediksi laju pertumbuhan penduduknya adalah > 0. Artinya pada tahun 2023, 2024, 2025, dan 2026 terjadi penambahan jumlah penduduk dari tahun 2022.

## *C.* Metode *Recurrent Neural Network*

### a. Menetapkan Nilai Input

Pada penelitian ini, pola *time series* yang dibentuk sebanyak lima. Artinya data input yang digunakan berjumlah lima inputan dan output yang dicari adalah data setelah data kelima. Pembentukan pola tersebut merujuk pada penelitian dari Anggoro (2023). Data *time series* jumlah penduduk secara lengkap dapat dilihat pada Lampiran 2.

### b. Normalisasi Data

Data jumlah penduduk Kabupaten Brebes dilakukan normalisasi data terlebih dahulu. Normalisasi dilakukan dengan menggunakan rumus persamaan 2.7. Data lengkap hasil perhitungan normalisasi data dapat dilihat pada Lampiran 3.

### c. Pembagian Data *Training* dan Data *Testing*

Berikut adalah hasil presentase pembagian data yang digunakan untuk penelitian.

Tabel 4.12 Pembagian Data RNN

| Pembagian Data | Presentase | Data |
|---|---|---|
| *Training* | 80% | 22 |
| *Testing* | 20% | 5 |
| Total | 100% | 27 |

Perbandingan pembagian data tersebut disesuaikan dengan perbandingan pembagian data

pada MLP karena untuk menentukan metode yang terbaik.

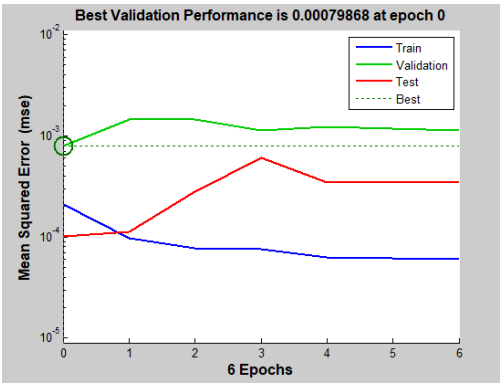
**d. Inisialisasi Parameter**

Sebelum melakukan proses *training,* maka harus menentukan parameter sebagai berikut.

1) Neuron lapisan input yang digunakan berjumlah 5 yaitu $X_1, X_2, X_3, X_4, X_5$.
2) Neuron lapisan tersembunyi berjumlah 5, 6, 7, 8, 9, dan 10.
3) Neuron pada lapisan *output* ada 1 yaitu Y.
4) Fungsi pelatihan (*training function*) yang digunakan adalah *Levenberg-Marquest* (LM).
5) Fungsi pembelajaran (*adaption learning function*) yang digunakan adalah *learngdm*.
6) Fungsi transfer (*transfer function*) yang digunakan adalah fungsi sigmoid bipolar (*tansig*) dan fungsi linear (*purelin*).
7) Maksimal *epoch* adalah 1000. Proses *training* setiap model dilakukan sebanyak 20 kali karena untuk setiap *training* data, bobot dapat berubah-ubah (Mustafidah, 2019).

**e. Pelatihan Jaringan**

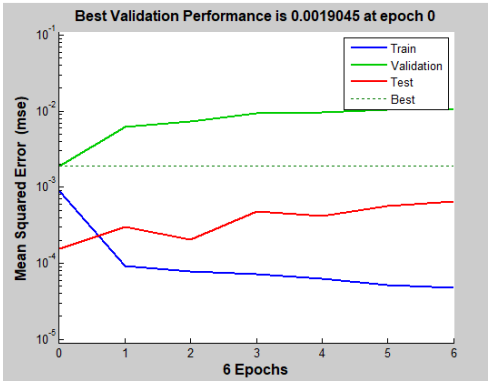
1. Arsitektur 5-5-1




Gambar 4.9 *Plot* Performa RNN Arsitektur 5-5-1

Dari gambar 4.9 menunjukkan bahwa garis *train*, garis *validation* dan garis *test* menghasilkan arah yang berbeda satu sama lain.

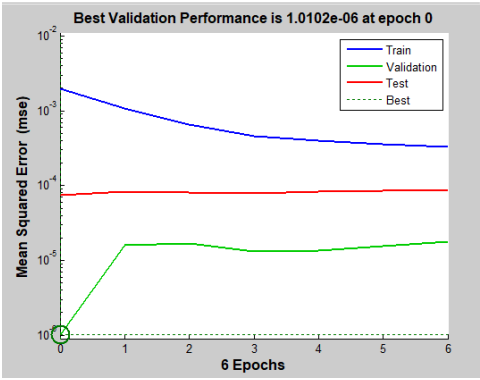
2. Arsitektur 5-6-1




Gambar 4.10 *Plot* Performa RNN Arsitektur 5-6-1

Dari gambar 4.10 menunjukkan bahwa setelah melewati epoch ke 1 hanya garis *train* yang memiliki pola berbeda.

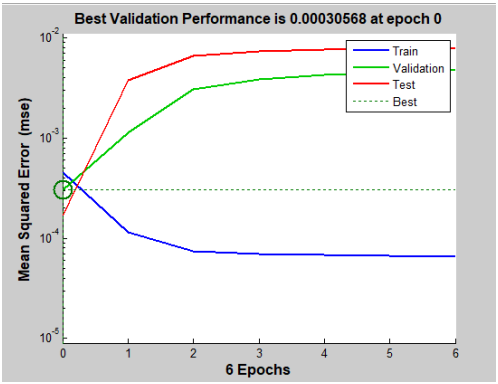
3. Arsitektur 5-7-1




Gambar 4.11 *Plot* Performa RNN Arsitektur 5-7-1

Dari gambar 4.11 menunjukkan bahwa hanya garis *test* saja yang stabil.

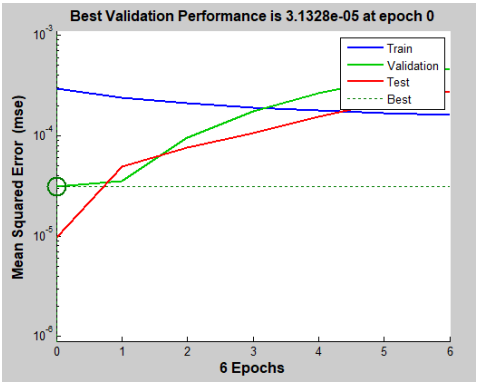
4. Arsitektur 5-8-1




Gambar 4.12 *Plot* Performa RNN Arsitektur 5-8-1

Gambar 4.12 menunjukkan bahwa garis *train* memiliki pola yang berbeda dari yang lain.

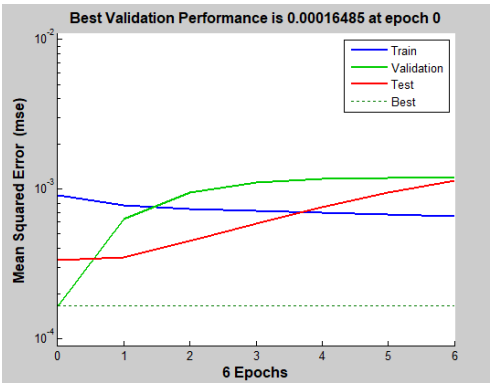
5. Arsitektur 5-9-1




Gambar 4.13 *Plot* Performa RNN Arsitektur 5-9-1

Dari gambar 4.13 menunjukkan bahwa garis *train* memiliki pola yang berbeda dari garis *test* dan garis *validation*.

6. Arsitektur 5-10-1




Gambar 4.14 *Plot* Performa RNN Arsitektur 5-10-1

Dari gambar 4.14 menunjukkan bahwa garis *test* dan garis *validation* berjalan saling mendekat. *Epoch* terbaik terjadi saat *epoch* ke 0.

**f. Pengujian Jaringan**

Setelah melakukan proses *training*, maka akan dilakukan pengujian untuk semua model. Pengujian jaringan dilakukan pada data jumlah penduduk Kabupaten Brebes tahun 2018-2022. Berikut adalah tabel hasil simulasi pengujian terhadap 6 model.

Tabel 4.13 Nilai *Output* (Y) Hasil *Testing* RNN

| Tahun | Target | Model | | | | | |
|---|---|---|---|---|---|---|---|
| | | 5-5-1 | 5-6-1 | 5-7-1 | 5-8-1 | 5-9-1 | 5-10-1 |
| 2018 | 0,54936 | 0,54406 | 0,55531 | 0,55383 | 0,57536 | 0,5774 | 0,55321 |
| 2019 | 0,55994 | 0,55049 | 0,56929 | 0,56919 | 0,5994 | 0,60291 | 0,56055 |
| 2020 | 0,84624 | 0,55517 | 0,58133 | 0,58259 | 0,62193 | 0,62657 | 0,56476 |
| 2021 | 0,86974 | 0,71276 | 0,76134 | 0,92351 | 0,50614 | 0,85436 | 0,66462 |
| 2022 | 0,9 | 0,69969 | 0,72765 | 1,0065 | 0,45841 | 0,92107 | 0,54902 |

Nilai error MSE untuk 6 model disajikan pada tabel berikut.

Tabel 4.14 Nilai *Error* MSE *Testing* RNN

| Model | *Error* MSE |
|---|---|
| 5-5-1 | 0,029921191 |
| 5-6-1 | 0,022351043 |
| 5-7-1 | 0,016770066 |

| Model | *Error* MSE |
|---|---|
| 5-8-1 | 0,075950951 |
| 5-9-1 | 0,010313612 |
| 5-10-1 | 0,048901472 |

Berdasarkan tabel 4.14 model arsitektur terbaik diberikan pada model 5-9-1. Model 5-9-1 menjadi model terbaik karena MSE yang dihasilkan mempunyai nilai terkecil dari model lainnya yaitu 0,010313612. Data lengkap hasil pengujian RNN dapat dilihat pada Lampiran 11.

**g. Denormalisasi Data**

Pada Tabel 4.13 dapat diperhatikan bentuk nilai *output* dari model yang menjadi model terbaik masih dalam normalisasi. Oleh karena itu, data akan diubah ke dalam nilai yang sebenarnya yang disebut denormalisasi. Perhitungan denormalisasi digunakan persamaan 2.38 sehingga didapatkan nilai sebagai berikut.

$$x_{2018} = \frac{(0{,}5774 - 0{,}1)(2010617 - 1536534)}{0{,}8} + 1536534 = 1819443$$

$$x_{2019} = \frac{(0{,}60291 - 0{,}1)(2010617 - 1536534)}{0{,}8} + 1536534 = 1834560$$

$$x_{2020} = \frac{(0{,}62657 - 0{,}1)(2010617 - 1536534)}{0{,}8} + 1536534 = 1848581$$

$$x_{2021} = \frac{(0{,}85436 - 0{,}1)(2010617 - 1536534)}{0{,}8} + 1536534 = 1983571$$

$$x_{2022} = \frac{(0{,}92107 - 0{,}1)(2010617 - 1536534)}{0{,}8} + 1536534 = 2023103$$

Hasil denormalisasi output merupakan hasil prediksi dari tahun 2018-2022. Berikut tabel hasil prediksi jumlah penduduk tahun 2018-2022.

Tabel 4.15 Hasil Prediksi Jumlah Penduduk Tahun 2018-2022 Metode RNN

| Tahun | Jumlah Penduduk |
|---|---|
| 2018 | 1819443 |
| 2019 | 1834560 |
| 2020 | 1848581 |
| 2021 | 1983571 |
| 2022 | 2023103 |

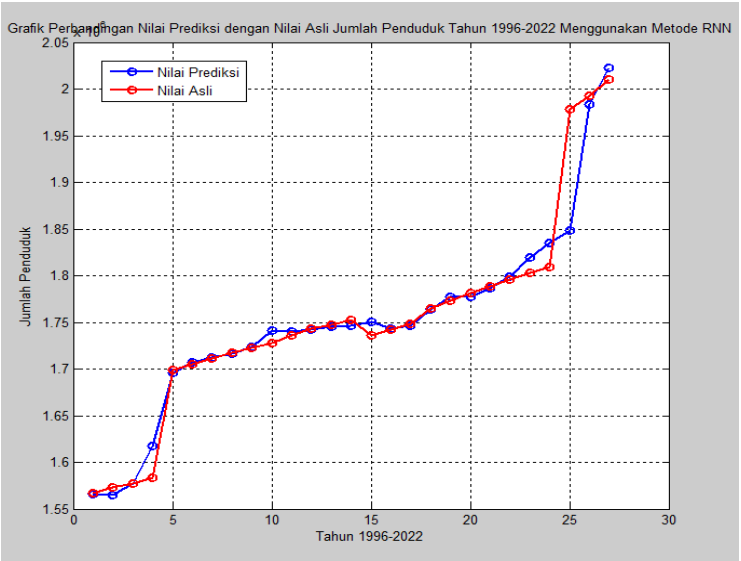
**h. Tabel Perbandingan Hasil Prediksi Tahun 1991-2022**

Berikut adalah tabel hasil perbandingan data asli dan data prediksi pada proses pelatihan dan pengujian.

Tabel 4.16 Hasil Perbandingan Data Asli dan Data Prediksi Metode RNN

| Tahun | Normalisasi Data | | Denormalisasi Data | |
|---|---|---|---|---|
| | Nilai Asli | Nilai Prediksi | Nilai Asli | Nilai Prediksi |
| 1991 | 0,15148 | - | 1.536.534 | - |
| 1992 | 0,16133 | - | 1.542.775 | - |
| 1993 | 0,16935 | - | 1.548.928 | - |
| 1994 | 0,17913 | - | 1.555.424 | - |
| 1995 | 0,37354 | - | 1.561.329 | - |
| 1996 | 0,38501 | 0,15006 | 1.567.044 | 1566200 |
| 1997 | 0,39551 | 0,14866 | 1.572.878 | 1565370 |
| 1998 | 0,4047 | 0,16952 | 1.577.631 | 1577732 |
| 1999 | 0,41348 | 0,2367 | 1.583.426 | 1617543 |
| 2000 | 0,4226 | 0,36898 | 1.698.635 | 1695933 |
| 2001 | 0,43727 | 0,38728 | 1.705.433 | 1706777 |
| 2002 | 0,44873 | 0,39662 | 1.711.657 | 1712312 |
| 2003 | 0,45588 | 0,40427 | 1.717.103 | 1716846 |
| 2004 | 0,46381 | 0,41658 | 1.722.306 | 1724140 |
| 2005 | 0,43715 | 0,44468 | 1.727.708 | 1740793 |
| 2006 | 0,44758 | 0,4438 | 1.736.401 | 1740271 |
| 2007 | 0,4577 | 0,44766 | 1.743.195 | 1742559 |
| 2008 | 0,48494 | 0,45305 | 1.747.430 | 1745753 |
| 2009 | 0,49967 | 0,45495 | 1.752.128 | 1746879 |
| 2010 | 0,51317 | 0,4616 | 1.736.331 | 1750820 |
| 2011 | 0,52583 | 0,44866 | 1.742.511 | 1743151 |
| 2012 | 0,53785 | 0,45371 | 1.748.510 | 1746144 |
| 2013 | 0,15148 | 0,48451 | 1.764.648 | 1764396 |
| 2014 | 0,16133 | 0,50562 | 1.773.379 | 1776906 |
| 2015 | 0,16935 | 0,50707 | 1.781.379 | 1777765 |
| 2016 | 0,17913 | 0,52143 | 1.788.880 | 1786275 |
| 2017 | 0,37354 | 0,54315 | 1.796.004 | 1799146 |
| 2018 | 0,54936 | 0,5774 | 1.802.829 | 1819443 |
| 2019 | 0,55994 | 0,60291 | 1.809.096 | 1834560 |
| 2020 | 0,84624 | 0,62657 | 1.978.759 | 1848581 |
| 2021 | 0,86974 | 0,85436 | 1.992.685 | 1983571 |
| 2022 | 0,9 | 0,92107 | 2.010.617 | 2023103 |

Grafik perbandingan nilai prediksi dan nilai aktual menggunakan metode RNN pada tabel 4.16 dapat digambarkan sebagai berikut.




Gambar 4.15 *Plot* Perbandingan Nilai Prediksi dan Nilai Aktual Metode RNN

Pada gambar 4.15 dapat diperhatikan bahwa garis *plot* antara nilai prediksi dengan nilai aktual tidak berbeda jauh. Dapat disimpulkan bahwa hasil prediksi jumlah penduduk Kabupaten Brebes periode 1996-2022 menggunakan metode RNN mendekati nilai sebenarnya.

**i. Perhitungan Nilai Akurasi**

Setelah dilakukan proses *testing* pada data tahun 2018-2022, dilanjutkan proses evaluasi terhadap data

sebenarnya tahun 2018-2022. Tujuan dilakukan evaluasi adalah untuk mengukur kemampuan model dalam memprediksi. Metode evaluasi yang digunakan yaitu MAPE. Berikut ini tabel nilai MAPE hasil prediksi.

Tabel 4.17 Hasil Nilai MAPE Metode RNN

| Tahun | Jumlah Penduduk | | *Absolute Percentage Error* |
|---|---|---|---|
| | **Hasil Prediksi** | **Data Asli** | |
| 2018 | 1819443 | 1802829 | 0,921553 |
| 2019 | 1834560 | 1809096 | 1,407573 |
| 2020 | 1848581 | 1978759 | 6,578752 |
| 2021 | 1983571 | 1992685 | 0,457395 |
| 2022 | 2023103 | 2010617 | 0,621011 |
| MAPE | | | 1,997257 |

Berdasarkan Tabel 4.17, ditunjukkan bahwa kategori nilai MAPE hasil prediksi metode *Recurrent Neural Network* adalah sangat baik karena menghasilkan nilai kurang dari 10%.

**j. Hasil Prediksi Laju Pertumbuhan Penduduk Tahun 2023-2026**

Selanjutnya, akan dilakukan prediksi laju pertumbuhan penduduk Kabupaten Brebes untuk tahun 2023-2026. Berikut hasil prediksi jumlah penduduk Kabupaten Brebes tahun 2023-2026.

Tabel 4.18 Hasil Prediksi Jumlah Penduduk Metode RNN

| Tahun | Hasil Prediksi |
|---|---|
| 2023 | 2026372 |
| 2024 | 2027091 |
| 2025 | 2022253 |
| 2026 | 2023467 |

Berdasarkan Tabel 4.18 dapat dihitung laju pertumbuhan penduduk dari tahun 2023-2026. Jumlah penduduk dasar adalah pada tahun 2022 yaitu 2.010.617. Berikut nilai yang diperoleh dari perhitungan laju pertumbuhan penduduk menggunakan persamaan 2.1.

Tahun 2023: $r = \left(\frac{2026372}{2010617}\right)^{\frac{1}{1}} - 1 = 0{,}0078359$

Tahun 2024: $r = \left(\frac{2027091}{2010617}\right)^{\frac{1}{2}} - 1 = 0{,}0040884$

Tahun 2025: $r = \left(\frac{2022253}{2010617}\right)^{\frac{1}{3}} - 1 = 0{,}0019254$

Tahun 2026: $r = \left(\frac{2023467}{2010617}\right)^{\frac{1}{4}} - 1 = 0{,}0015939$

Berikut adalah hasil prediksi laju pertumbuhan penduduk Kabupaten Brebes tahun 2023-2026.

Tabel 4.19 Hasil Prediksi Laju pertumbuhan Penduduk Metode RNN

| Tahun | Hasil Prediksi |
|---|---|
| 2023 | 0,784% |
| 2024 | 0,409% |
| 2025 | 0,193% |
| 2026 | 0,159% |

Berdasarkan Tabel 4.19 dapat diketahui bahwa pada tahun 2023, 2024, 2025, dan 2026, nilai prediksi laju pertumbuhan penduduknya adalah > 0. Artinya pada tahun 2023, 2024, 2025, dan 2026 terjadi penambahan jumlah penduduk dari tahun 2022.

## *D.* Perbandingan Metode *Multilayer Perceptron* dan *Recurrent Neural Network*

Pada bagian ini adalah membandingkan metode MLP dan RNN dengan menganalisis perhitungan nilai akurasi. Nilai keakuratan prediksi setiap metode dibandingkan untuk mengetahui metode paling tepat yang mendapatkan nilai kesalahan (*error*) terendah menggunakan MAPE.

Tabel 4.20 Hasil Perbandingan Nilai MAPE MLP dan RNN

| Metode | MAPE |
|---|---|
| MLP | 2,1896 |
| RNN | 1,9973 |

Tabel 4.20 merupakan tabel perbandingan nilai MAPE hasil prediksi jumlah penduduk Kabupaten Brebes menggunakan metode MLP dan RNN. Hasil prediksi menunjukkan bahwa metode MLP dan RNN memiliki nilai MAPE < 10% sehingga metode MLP dan RNN layak digunakan sebagai metode prediksi.

Dari nilai MAPE tersebut dapat digunakan untuk menentukan nilai akurasi. Menghitung nilai akurasi dapat menggunakan persamaan 2.37. Berikut adalah tabel perbandingan nilai akurasi metode MLP dan RNN.

Tabel 4.21 Hasil Perbandingan Nilai Akurasi MLP dan RNN

| Metode | Nilai Akurasi |
|---|---|
| MLP | 97,8% |
| RNN | 98% |

Berdasarkan hasil perbandingan Tabel 4.20 dan Tabel 4.21 menunjukkan bahwa metode RNN lebih baik digunakan dalam prediksi karena menghasilkan nilai MAPE terendah dibandingkan metode MLP yang menghasilkan nilai MAPE lebih besar dan nilai akurasi yang dihasilkan RNN lebih besar dibandingkan metode MLP.

Berdasarkan data yang sudah dikeluarkan BPS Kabupaten Brebes, jumlah penduduk Kabupaten Brebes tahun 2023 adalah 2.043.077 sehingga laju pertumbuhan penduduk dari tahun 2022 ke tahun 2023 adalah 1,614%. Jadi perbedaan nilai laju pertumbuhan penduduk data asli dikurangi data prediksi MLP adalah -0,965% dan perbedaan nilai laju pertumbuhan penduduk data asli dikurangi data prediksi RNN adalah 0,83%.

# BAB V
# PENUTUP

## A. Kesimpulan

Kesimpulan yang diperoleh berdasarkan hasil dan pembahasan yang sudah dijelaskan pada bab empat, yaitu sebagai berikut.

1. Hasil prediksi laju pertumbuhan penduduk Kabupaten Brebes tahun 2023-2026 menggunakan metode *Multilayer Perceptron* adalah sebagai berikut.
    a. Hasil prediksi laju pertumbuhan penduduk tahun 2023 adalah 2,579% yang berarti terjadi penambahan jumlah penduduk sebesar 51.861 jiwa dari tahun 2022 ke tahun 2023.
    b. Hasil prediksi laju pertumbuhan penduduk tahun 2024 adalah 1,556% yang berarti terjadi penambahan jumlah penduduk sebesar 63.056 jiwa dari tahun 2022 ke 2024.
    c. Hasil prediksi laju pertumbuhan penduduk tahun 2025 adalah 0,795% yang berarti terjadi penambahan jumlah penduduk sebesar 48.346 jiwa dari tahun 2022 ke 2025.
    d. Hasil prediksi laju pertumbuhan penduduk tahun 2026 adalah 0,722% yang berarti terjadi

penambahan jumlah penduduk sebesar 58.724 jiwa dari tahun 2022 ke 2026.

2. Hasil prediksi laju pertumbuhan penduduk Kabupaten Brebes tahun 2023-2026 menggunakan metode *Recurrent Neural Network* adalah sebagai berikut.
   a. Hasil prediksi laju pertumbuhan penduduk tahun 2023 adalah 0,784% yang berarti terjadi penambahan jumlah penduduk sebesar 15.755 jiwa dari tahun 2022 ke 2023.
   b. Hasil prediksi laju pertumbuhan penduduk tahun 2024 adalah 0,409% yang berarti terjadi penambahan jumlah penduduk sebesar 16.474 jiwa dari tahun 2022 ke 2024.
   c. Hasil prediksi laju pertumbuhan penduduk tahun 2025 adalah 0,193% yang berarti terjadi penambahan jumlah penduduk sebesar 11.636 jiwa dari tahun 2022 ke 2025.
   d. Hasil prediksi laju pertumbuhan penduduk tahun 2026 adalah 0,159% yang berarti terjadi penambahan jumlah penduduk sebesar 12.850 jiwa dari tahun 2022 ke 2026.
3. Perbandingan nilai MAPE hasil prediksi menggunakan metode *Multilayer Perceptron* (MLP) dan *Recurrent Neural Network* (RNN) dengan pembagian data *training* 80% dan data *testing* 20% dihasilkan bahwa metode

RNN mendapatkan nilai MAPE lebih kecil yaitu 1,9973% dibandingkan metode MLP yaitu 2,1896%. Didapatkan nilai akurasi untuk metode RNN sebesar 98% dan untuk metode MLP didapatkan nilai akurasi sebesar 97,8%. Berdasarkan perbandingan nilai MAPE tersebut, didapatkan metode RNN lebih baik daripada metode MLP.

## B. Saran

Saran yang dapat diberikan untuk penelitian berikutnya berdasarkan hasil dan pembahasan yaitu sebagai berikut.

1. Bagi peneliti selanjutnya, dapat digunakan metode jaringan saraf tiruan yang lain supaya memperoleh nilai yang lebih tepat.
2. Bagi peneliti selanjutnya, data yang digunakan dapat berupa data harian atau data bulanan yang bertujuan untuk memperhatikan nilai prediksi dalam jumlah rentang waktu dekat dan rentang waktu lama.
3. Bagi peneliti selanjutnya, dapat digunakan pembagian data *training* dan data *testing* dengan persentase lain sehingga dapat dijadikan bahan perbandingan dengan hasil penelitian ini.

## DAFTAR PUSTAKA


Akhirul, *et al*. 2020. Dampak Negatif Pertumbuhan Penduduk terhadap Lingkungan dan Upaya Mengatasinya. *Jurnal Kependudukan dan Pembangunan Lingkungan*. 1 (3): 76-84.

Anggoro, A.Y. 2023. Aplikasi Jaringan Syaraf Tiruan dengan Algoritma Backpropagasi untuk Memprediksi Perubahan Harga Saham LQ 45. *Semnaptika*. 1 (1): 46-53.

Ardilla, Y. 2016. *Metode Hibrida ARIMA dan Multilayer Perceptron untuk Peramalan Jangka Pendek Konsumsi Listrik di Jawa Timur*. Tesis. Surabaya: ITS.

Ariwibowo, A.B., *et al*. 2019. Peramalan Harga Beras IR64 Kualitas III Menggunakan Metode *Multilayer Perceptron*, *Holt-Winters* Dan *Auto Refressive Integrated Moving Average*. *Jurnal Teknik Informatika*. 11(2): 60-64.

BPS. 2010. *Pedoman Penghitungan Proyeksi Penduduk dan Angkatan Kerja*. Jakarta: Badan Pusat Statistik.

Choubin, B., *et al*. 2014. Multiple Linear Regression, Multilayer Perceptron Network and Adaptive Neuro-Fuzzy Inference System for the Prediction Of Precipitation Based On Large-Scale Cliate Signals*. Hydrological Sciences Journal.* 61 (6): 1001-1009.

Deltania, D.O. 2023. Prakiraan Kebutuhan Energi Listrik Wilayah Provinsi Jawa Tengah menggunakan Metode Backpropagation Neural Network dan Metode Ekstrapolasi Linier. *Jurnal Simetris*. 14 (2): 351-365.

Desriwendi, Hoyyi A., Wuryandari T. 2015. Pemodelan Geographically Weighted Logistic Regression (GWLR)

dengan Fungsi Pembobot Fixed Gaussian Kernel dan Adaptive Gaussian Kernel. *Jurnal Gaussian*. 4 (2): 193-204.

Arafat, Yaser. 2021. *SP2020, Penduduk Brebes Terbanyak di Jateng.* Portal Resmi Provinsi Jawa Tengah.

Farhah, A., Prasasti, A.L. & Paryasto, M.W. 2021. Implementasi *Recurrent Neural Network* dalam Memprediksi Kepadatan Restoran Berbasis LSTM. *Jurnal Media Informatika Budidarma*. 5(2): 524-531.

Fausett, L. 1994. *Fundamentals of Neural Network (Archetectures, Algoruthms, and Applications).* New-Jersey: Prentice-Hall.

Fauziyah, T.A. & Utomo, A.P. 2023. *Angka Kemiskinan Brebes Turun 1,38 Perse, Wagub Taj Yasin: Gencarkan Lagi*. Kompas. Semarang. 13 Januari.

Gulli, Antonio & Pal, Sujit. 2017. *Deep Learning with Keras*. Birmingham: Packt Publishing.

Hafizah, Tugiono & Pane, D.K. 2023. *JST for Beginner (Memulai Belajar Konsep Dasar Jaringan Syaraf Tiruan)*. Jambi: PT. Sonpedia Publishing Indonesia.

Hamid, *et al*. 2011. Accelerating Learning Performance of Back Propagation Algorithm by Using Adaptive Gain Together with Adaptive Momentum and Adaptive Learning Rate on Classification Problems*. Internastional Journal of Software Engineeringand Its Applications.* 5(4): 559-570.

Hardati, Puji. 2013. Pertumbuhan Penduduk dan Struktur Lapangan Pekerjaan di Jawa Tengah. *Forum Ilmu Sosial.* 40 (2): 219-229.

Hardiyanti, A.F. & Firianah, D. 2021. Perbandingan Algoritma C4.5 dan Multilayer Perceptron untuk Klasifikasi Kelas Rumah Sakit di DKI Jakarta. *Jurnal Telekomunikasi dan Komputer*. 11 (3): 198-209.

Hermawan, Nanang. 2014. *Aplikasi Model Recurrrent Neural Network dan Recurrent Neuro Fuzzy untuk Peramalan Banyaknya Penumpang Kereta Api Jabodetabek*. Skripsi. Yogyakarta: Universitas Negeri Yogyakarta.

Houari, M.B.E., *et al*. 2015. Prediction of Air Temperature Using Multi-layer Perceptrons with Levenberg-Marquardt Training Algorithm. *International Research Journal of Engineering and Technology (IRJET).* 02 (08): 26-32.

Jaya, H., *et al*. 2018. *Kecerdasan Buatan*. Makassar: Fakultas MIPA Universitas Negeri Makassar.

Khoirudin, *et al*. 2018. Prediksi Penerimaan Mahasiswa Baru dengan Multilayer Perceptron. *Jurnal Pengembangan Rekayasa dan Teknologi*. 14 (1): 1-4.

Kholis, 2015. Analisis Variasi Parameter Backpropagation Artificial Neural Network terhadap Pengenalan Pola Data Iris. *Jurnal Teknik dan Ilmu Komputer Ukrida*. 1-10.

Kusnanti, E.A., *et al*. 2022. Predicting Velocity and Direction of Ocean Surface Currents using Elman Recurrent Neural Network Method. *Journal of Information Systems Engineering and Business Intelligence.* 8 (1): 21-30.

Kusumadewi, S. 2004. *Membangun Jaringan Syaraf Tiruan: Menggunakan MATLAB & Excel Link*. Yogyakarta: Graha Ilmu.

Lesnussa, Y.A., *et al*. 2017. Aplikasi Jaringan Saraf Tiruan Backpropagation untuk Penyebaran Penyakit Demam

Berdarah Dengue (DBD) di Kota Ambon. *Jurnal Matematika Integratif*. 13 (2): 63-72.

Marisya, A., *et al*. 2020. Konsep Model Discovery Learning pada Pembelajaran Tematik Terpadu di Sekolah Dasar menurut Pandangan para Ahli. *Jurnal Pendidikan Tambusai*. 4 (3): 2189-2198.

Moreno, J. J. M., *et al*. 2013. Using the R-MAPE index as a resistant measure of forecast accuracy. *Psicothema*. 25 (4): 500-506.

Muharrom, M. 2023. Analisis Komparassi Algoritma Data Mining Naïve Bayes, K-Nearest Neighbors dan Regresi Linier dalam Prediksi Harga Emas. *Bulletin of Information Technology*. 4 (4): 430-438.

Muis, Saludin. 2017. *Jaringan Syaraf Tiruan; Sistem Kecerdasan Tiruan dengan Kemampuan Belajar dan Adaptasi*. Edisi pertama. Yogyakarta: Teknosain.

Mulyatiningsih, Endang. 2011. *Riset Terapan Bidang Pendidikan dan Teknik*. Yogyakarta: Universitas Negeri Yogyakarta.

Musashi, J. P. 2014. *Prediksi Harga High Nilai Tukar EUR-USD menggunakan Recurrent Neural Network (RNN).* Skripsi. Malang: Universitas Brawijaya.

Mustafidah, H., Budiastanto, M.H., dan Suwarsito. 2019. Kinerja Algoritma Pelatihan Levenberg-Marquardt dalam Variasi Banyaknya Neuron pada Lapisan Tersembunyi. *JUITA: Jurnal Informatika*. 7 (2): 115-123.

Nasution & Hakim, A. 2008. *Perencanaan dan Pengendalian Produksi*. Edisi Kedua. Surabaya: Prima Printing.

Nooriansyah, S., *et al*. 2018. Analisis Kinerja Metode Artificial Neural Network dan Support Vector Regression untuk Prediksi Significant Wave Height. *Jurnal Teknologi Informasi dan Komunikasi.* 13 (1): 17-24.

Nurfita, R. D. 2018. *Implementasi Deep Learning Berbasis Tensorflow untuk Pengenalan Sidik Jari*. Skripsi. Surakarta: Universitas Muhammadiyah.

Olawoyin, A. & Chen, Y. 2018. Predicting the Future with Artificial Neural Network. *Procedia Computer Science*. 140 (2018): 383-392.

Pakaja, F., Naba, A., & Purwanto. 2012. Peramalan Penjualan Mobil Menggunakan Jaringan Syaraf Tiruan dan Certainty Factor*. Jurnal EECCIS*. 6 (1): 23-28.

Pamungkas I., *et al*. 2022. Studi Komparasi Fungsi Aktivasi Sigmoid Biner, Sigmoid Bipolar dan Linear pada Jaringan Saraf Tiruan dalam Menentukan Warna RGB Menggunakan Matlab. *Serambi Engineering*. 7 (4): 3749-3758.

Purwadi, Ramadhan, dan Safitri. 2019. Penerapan Data *Mining* untuk Mengestimasi Laju Pertumbuhan Penduduk Menggunakan Metode Regresi Berganda pada BPS Deli Serdang. *Sains dan Komputer (SAINTIKOM).* 18 (1): 55-61.

Putri, M. 2021. *Perbandingan Metode Extreme Learning Machine dan Multilayer Perceptron dalam Peramalan Saham Nippon Paint*. Skripsi. Yogyakarta: Universitas Islam Indonesia.

Putri, Maulidya Ayu. 2021. *Perbandingan Metode Extreme Learning Machine dan Multilayer Perceptron Dalam Peramalan Saham Nippon Paint*. Skripsi. Yogyakarta: Universitas Islam Indonesia Yogyakarta.

Rahmawaty, Resa. 2023. Prediksi Laju Pertumbuhan Penduduk dengan Program Matlab menggunakan Metode Jaringan Syaraf Tiruan. *Jurnal Publikasi Teknik Informatika*. 2 (1): 1-5.

Retnoningsih, E. & Pramudita, R. 2020. Mengenal Machine Learning dengan Teknik Supervised dan Unsupervised Learning Menggunakan Python. *Bina Insani ICT Journal*. 7 (2): 156-165.

Rifaldo, Muhammad. *et al*. 2021. Peramalan Kedatangan Wisatawan Mancanegara ke Indonesia menurut Kebangsaan Perbulannya menggunakan Metode *Multilayer Perceptron*. *Jurnal Computer Science and Information Technology (CoSiTech).* 2 (2): 113-119.

Ryandhi, Rizky. 2017. *Penerapan Metode Artificial Neural Network (ANN) untuk Peramalan Inflasi di Indonesia*. Skripsi. Surabaya: Institut Teknologi Sepuluh November.

Santoso, W. *et al*. 2023. Prediksi Volume Sampah di TPSA Banyuurip Menggunakan Metode Backpropagation Neural Network. *Jurnal Media Informatika Budidarma*. 7 (1). 464-472.

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 Informasi*. 12 (1): 61-69.

Tarkus. E.D., *et al*. 2020. Implementasi Metode *Recurrent Neural Network* pada Pengklasifikasian Kualitas Telur Puyuh. *Jurnal Teknik Informatika*. 15 (2): 137-144.

Varcellis, C. 2009. *Business Intelligence: Data Mining and Optimization for Decision Making*. Milano, Italy.

Wang, J., *et al*. 2016. Financial Time Series Prediction Using Elman Recurrent Random Neural Networks. *Computational Intelligemce and Neuroscience*.

Wardani, Wulan. 2021. *Prediksi Harga Saham Syariah Menggunakan Metode Recurrent Neural Network-Long Short Term Memory*. Skripsi. Surabaya: Universitas Islam Negeri Sunan Ampel.

Warsito, Budi. 2009. *Kapita Selekta Statistika Neural Network*. Semarang: Fakultas MIPA Universitas Diponegoro.

Wibawa, A.P., *et al*. 2020. Multilayer Perceptron untuk Prediksi Sessions pada Sebuah Website *Journal Elektronik. Indonesian Journal of Data and Science*. 1(3): 57-67.

Windarto, A.P., *et al*. 2020. *Jaringan Syaraf Tiruan: Algoritma Prediksi dan Implementasi*. Yayasan Kita Menulis.

World Bank. 2022. Population, total. https://data.worldbank.org/indicator/SP.POP.TOTL?most_recent_value_desc=true

Yeung, Daniel S., *et al*. 2010. *Sensitivity Analysis for Neural Network*. Berlin: Springer.

Zakariah, M. A., *et al*. 2020. *Metodologi Penelitian Kualitatif, Kuantitatif, Action Research, Research and Development (RnD).* Kolaka: Yayasan Pondok Pesantren Al Mawaddah Warrahmah.

## LAMPIRAN

**Lampiran 1** Data Jumlah Penduduk Kabupaten Brebes Tahun 1991-2022

| No | Tahun | Jumlah Penduduk (Jiwa) |
|---|---|---|
| 1 | 1991 | 1.536.534 |
| 2 | 1992 | 1.542.775 |
| 3 | 1993 | 1.548.928 |
| 4 | 1994 | 1.555.424 |
| 5 | 1995 | 1.561.329 |
| 6 | 1996 | 1.567.044 |
| 7 | 1997 | 1.572.878 |
| 8 | 1998 | 1.577.631 |
| 9 | 1999 | 1.583.426 |
| 10 | 2000 | 1.698.635 |
| 11 | 2001 | 1.705.433 |
| 12 | 2002 | 1.711.657 |
| 13 | 2003 | 1.717.103 |
| 14 | 2004 | 1.722.306 |
| 15 | 2005 | 1.727.708 |
| 16 | 2006 | 1.736.401 |
| 17 | 2007 | 1.743.195 |
| 18 | 2008 | 1.747.430 |
| 19 | 2009 | 1.752.128 |
| 20 | 2010 | 1.736.331 |
| 21 | 2011 | 1.742.511 |

| No | Tahun | Jumlah Penduduk (Jiwa) |
|---|---|---|
| 22 | 2012 | 1.748.510 |
| 23 | 2013 | 1.764.648 |
| 24 | 2014 | 1.773.379 |
| 25 | 2015 | 1.781.379 |
| 26 | 2016 | 1.788.880 |
| 27 | 2017 | 1.796.004 |
| 28 | 2018 | 1.802.829 |
| 29 | 2019 | 1.809.096 |
| 30 | 2020 | 1.978.759 |
| 31 | 2021 | 1.992.685 |
| 32 | 2022 | 2.010.617 |

**Lampiran 2** Data *Time Series* Jumlah Penduduk

| No | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | Y |
|---|---|---|---|---|---|---|
| 1 | 1.536.534 | 1.542.775 | 1.548.928 | 1.555.424 | 1.561.329 | 1.567.044 |
| 2 | 1.542.775 | 1.548.928 | 1.555.424 | 1.561.329 | 1.567.044 | 1.572.878 |
| 3 | 1.548.928 | 1.555.424 | 1.561.329 | 1.567.044 | 1.572.878 | 1.577.631 |
| 4 | 1.555.424 | 1.561.329 | 1.567.044 | 1.572.878 | 1.577.631 | 1.583.426 |
| 5 | 1.561.329 | 1.567.044 | 1.572.878 | 1.577.631 | 1.583.426 | 1.698.635 |
| 6 | 1.567.044 | 1.572.878 | 1.577.631 | 1.583.426 | 1.698.635 | 1.705.433 |
| 7 | 1.572.878 | 1.577.631 | 1.583.426 | 1.698.635 | 1.705.433 | 1.711.657 |
| 8 | 1.577.631 | 1.583.426 | 1.698.635 | 1.705.433 | 1.711.657 | 1.717.103 |
| 9 | 1.583.426 | 1.698.635 | 1.705.433 | 1.711.657 | 1.717.103 | 1.722.306 |
| 10 | 1.698.635 | 1.705.433 | 1.711.657 | 1.717.103 | 1.722.306 | 1.727.708 |
| 11 | 1.705.433 | 1.711.657 | 1.717.103 | 1.722.306 | 1.727.708 | 1.736.401 |
| 12 | 1.711.657 | 1.717.103 | 1.722.306 | 1.727.708 | 1.736.401 | 1.743.195 |
| 13 | 1.717.103 | 1.722.306 | 1.727.708 | 1.736.401 | 1.743.195 | 1.747.430 |
| 14 | 1.722.306 | 1.727.708 | 1.736.401 | 1.743.195 | 1.747.430 | 1.752.128 |
| 15 | 1.727.708 | 1.736.401 | 1.743.195 | 1.747.430 | 1.752.128 | 1.736.331 |
| 16 | 1.736.401 | 1.743.195 | 1.747.430 | 1.752.128 | 1.736.331 | 1.742.511 |
| 17 | 1.743.195 | 1.747.430 | 1.752.128 | 1.736.331 | 1.742.511 | 1.748.510 |
| 18 | 1.747.430 | 1.752.128 | 1.736.331 | 1.742.511 | 1.748.510 | 1.764.648 |
| 19 | 1.752.128 | 1.736.331 | 1.742.511 | 1.748.510 | 1.764.648 | 1.773.379 |
| 20 | 1.736.331 | 1.742.511 | 1.748.510 | 1.764.648 | 1.773.379 | 1.781.379 |
| 21 | 1.742.511 | 1.748.510 | 1.764.648 | 1.773.379 | 1.781.379 | 1.788.880 |
| 22 | 1.748.510 | 1.764.648 | 1.773.379 | 1.781.379 | 1.788.880 | 1.796.004 |
| 23 | 1.764.648 | 1.773.379 | 1.781.379 | 1.788.880 | 1.796.004 | 1.802.829 |
| 24 | 1.773.379 | 1.781.379 | 1.788.880 | 1.796.004 | 1.802.829 | 1.809.096 |
| 25 | 1.781.379 | 1.788.880 | 1.796.004 | 1.802.829 | 1.809.096 | 1.978.759 |
| 26 | 1.788.880 | 1.796.004 | 1.802.829 | 1.809.096 | 1.978.759 | 1.992.685 |
| 27 | 1.796.004 | 1.802.829 | 1.809.096 | 1.978.759 | 1.992.685 | 2.010.617 |

**Lampiran 3** Normalisasi Data

| No | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | Y |
|---|---|---|---|---|---|---|
| 1 | 0,1 | 0,11053 | 0,12091 | 0,13188 | 0,14184 | 0,15148 |
| 2 | 0,11053 | 0,12091 | 0,13188 | 0,14184 | 0,15148 | 0,16133 |
| 3 | 0,12091 | 0,13188 | 0,14184 | 0,15148 | 0,16133 | 0,16935 |
| 4 | 0,13188 | 0,14184 | 0,15148 | 0,16133 | 0,16935 | 0,17913 |
| 5 | 0,14184 | 0,15148 | 0,16133 | 0,16935 | 0,17913 | 0,37354 |
| 6 | 0,15148 | 0,16133 | 0,16935 | 0,17913 | 0,37354 | 0,38501 |
| 7 | 0,16133 | 0,16935 | 0,17913 | 0,37354 | 0,38501 | 0,39551 |
| 8 | 0,16935 | 0,17913 | 0,37354 | 0,38501 | 0,39551 | 0,4047 |
| 9 | 0,17913 | 0,37354 | 0,38501 | 0,39551 | 0,4047 | 0,41348 |
| 10 | 0,37354 | 0,38501 | 0,39551 | 0,4047 | 0,41348 | 0,4226 |
| 11 | 0,38501 | 0,39551 | 0,4047 | 0,41348 | 0,4226 | 0,43727 |
| 12 | 0,39551 | 0,4047 | 0,41348 | 0,4226 | 0,43727 | 0,44873 |
| 13 | 0,4047 | 0,41348 | 0,4226 | 0,43727 | 0,44873 | 0,45588 |
| 14 | 0,41348 | 0,4226 | 0,43727 | 0,44873 | 0,45588 | 0,46381 |
| 15 | 0,4226 | 0,43727 | 0,44873 | 0,45588 | 0,46381 | 0,43715 |
| 16 | 0,43727 | 0,44873 | 0,45588 | 0,46381 | 0,43715 | 0,44758 |
| 17 | 0,44873 | 0,45588 | 0,46381 | 0,43715 | 0,44758 | 0,4577 |
| 18 | 0,45588 | 0,46381 | 0,43715 | 0,44758 | 0,4577 | 0,48494 |
| 19 | 0,46381 | 0,43715 | 0,44758 | 0,4577 | 0,48494 | 0,49967 |
| 20 | 0,43715 | 0,44758 | 0,4577 | 0,48494 | 0,49967 | 0,51317 |
| 21 | 0,44758 | 0,4577 | 0,48494 | 0,49967 | 0,51317 | 0,52583 |
| 22 | 0,4577 | 0,48494 | 0,49967 | 0,51317 | 0,52583 | 0,53785 |
| 23 | 0,48494 | 0,49967 | 0,51317 | 0,52583 | 0,53785 | 0,54936 |
| 24 | 0,49967 | 0,51317 | 0,52583 | 0,53785 | 0,54936 | 0,55994 |
| 25 | 0,51317 | 0,52583 | 0,53785 | 0,54936 | 0,55994 | 0,84624 |
| 26 | 0,52583 | 0,53785 | 0,54936 | 0,55994 | 0,84624 | 0,86974 |
| 27 | 0,53785 | 0,54936 | 0,55994 | 0,84624 | 0,86974 | 0,9 |

Keterangan:

Biru = *training*

Hijau = *testing*

**Lampiran 4** Data Transpos

| | | | | | | |
|---|---|---|---|---|---|---|
| X1 | 0,1 | 0,11053 | 0,12091 | 0,13188 | 0,14184 | 0,15148 |
| X2 | 0,11053 | 0,12091 | 0,13188 | 0,14184 | 0,15148 | 0,16133 |
| X3 | 0,12091 | 0,13188 | 0,14184 | 0,15148 | 0,16133 | 0,16935 |
| X4 | 0,13188 | 0,14184 | 0,15148 | 0,16133 | 0,16935 | 0,17913 |
| X5 | 0,14184 | 0,15148 | 0,16133 | 0,16935 | 0,17913 | 0,37354 |
| Y | 0,15148 | 0,16133 | 0,16935 | 0,17913 | 0,37354 | 0,38501 |

Lanjutan

| | | | | | | |
|---|---|---|---|---|---|---|
| X1 | 0,16133 | 0,16935 | 0,17913 | 0,37354 | 0,38501 | 0,39551 |
| X2 | 0,16935 | 0,17913 | 0,37354 | 0,38501 | 0,39551 | 0,4047 |
| X3 | 0,17913 | 0,37354 | 0,38501 | 0,39551 | 0,4047 | 0,41348 |
| X4 | 0,37354 | 0,38501 | 0,39551 | 0,4047 | 0,41348 | 0,4226 |
| X5 | 0,38501 | 0,39551 | 0,4047 | 0,41348 | 0,4226 | 0,43727 |
| Y | 0,39551 | 0,4047 | 0,41348 | 0,4226 | 0,43727 | 0,44873 |

Lanjutan

| | | | | | | |
|---|---|---|---|---|---|---|
| X1 | 0,4047 | 0,41348 | 0,4226 | 0,43727 | 0,44873 | 0,45588 |
| X2 | 0,41348 | 0,4226 | 0,43727 | 0,44873 | 0,45588 | 0,46381 |
| X3 | 0,4226 | 0,43727 | 0,44873 | 0,45588 | 0,46381 | 0,43715 |
| X4 | 0,43727 | 0,44873 | 0,45588 | 0,46381 | 0,43715 | 0,44758 |
| X5 | 0,44873 | 0,45588 | 0,46381 | 0,43715 | 0,44758 | 0,4577 |
| Y | 0,45588 | 0,46381 | 0,43715 | 0,44758 | 0,4577 | 0,48494 |

Lanjutan

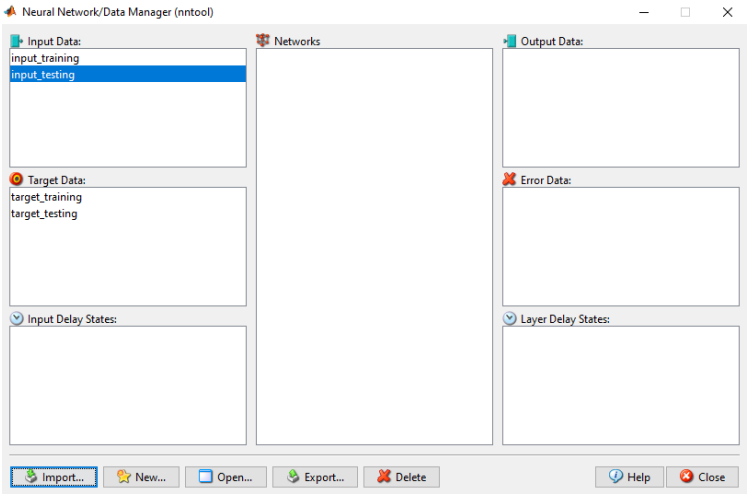
| | | | | | | |
|---|---|---|---|---|---|---|
| X1 | 0,46381 | 0,43715 | 0,44758 | 0,4577 | 0,48494 | 0,49967 |
| X2 | 0,43715 | 0,44758 | 0,4577 | 0,48494 | 0,49967 | 0,51317 |
| X3 | 0,44758 | 0,4577 | 0,48494 | 0,49967 | 0,51317 | 0,52583 |
| X4 | 0,4577 | 0,48494 | 0,49967 | 0,51317 | 0,52583 | 0,53785 |
| X5 | 0,48494 | 0,49967 | 0,51317 | 0,52583 | 0,53785 | 0,54936 |
| Y | 0,49967 | 0,51317 | 0,52583 | 0,53785 | 0,54936 | 0,55994 |

Lanjutan

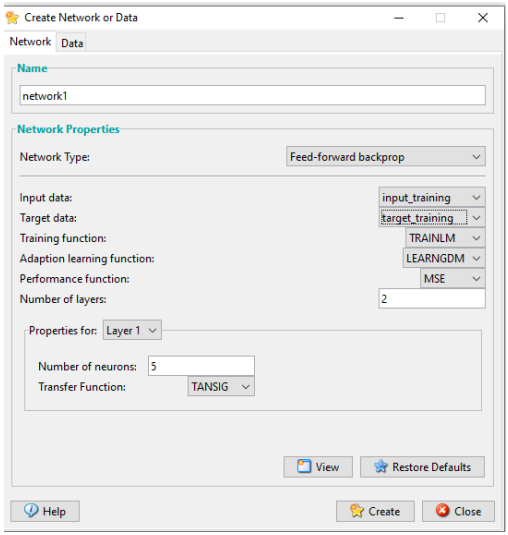
| | | | |
|---|---|---|---|
| X1 | 0,51317 | 0,52583 | 0,53785 |
| X2 | 0,52583 | 0,53785 | 0,54936 |
| X3 | 0,53785 | 0,54936 | 0,55994 |
| X4 | 0,54936 | 0,55994 | 0,84624 |
| X5 | 0,55994 | 0,84624 | 0,86974 |
| Y | 0,84624 | 0,86974 | 0,9 |

**Lampiran 5** Proses Pelatihan MLP menggunakan MATLAB R2013a

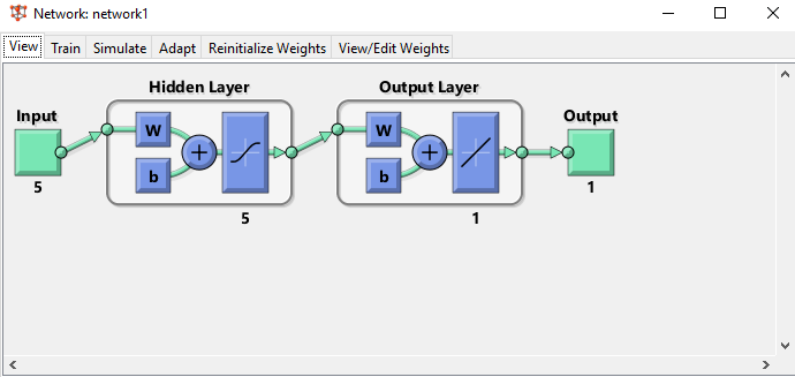
1. *Neural Network Tool* (*nntool*)



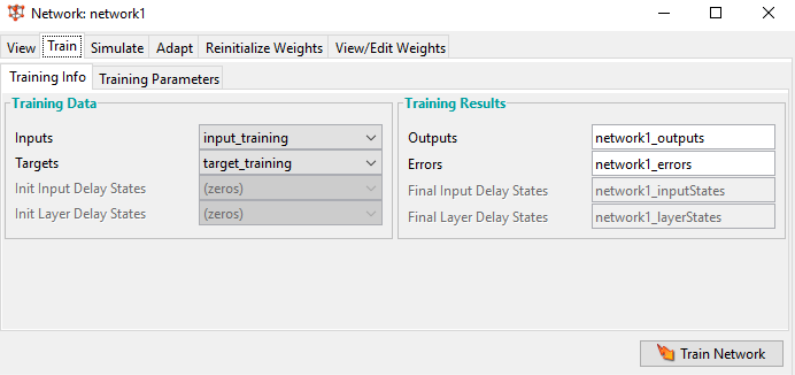

2. *Create a Network*



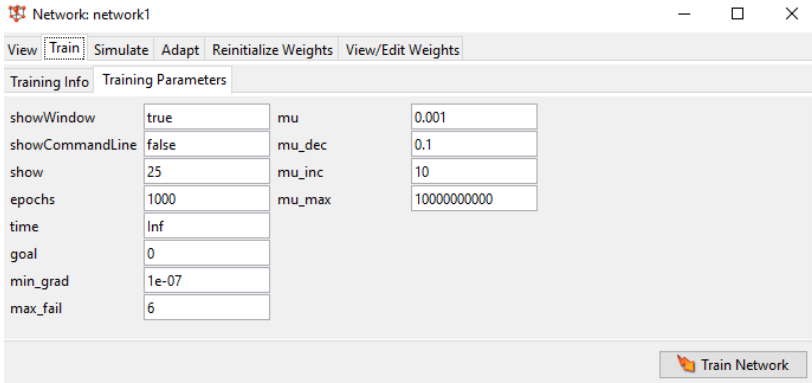
3. Arsitektur Jaringan



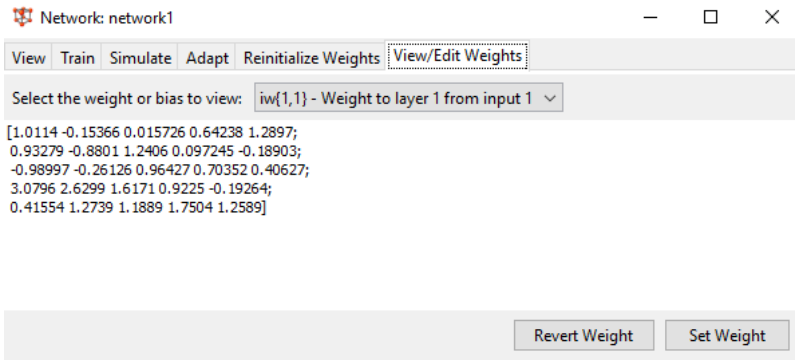

4. *Training Info*



## 5. *Training Parameters*

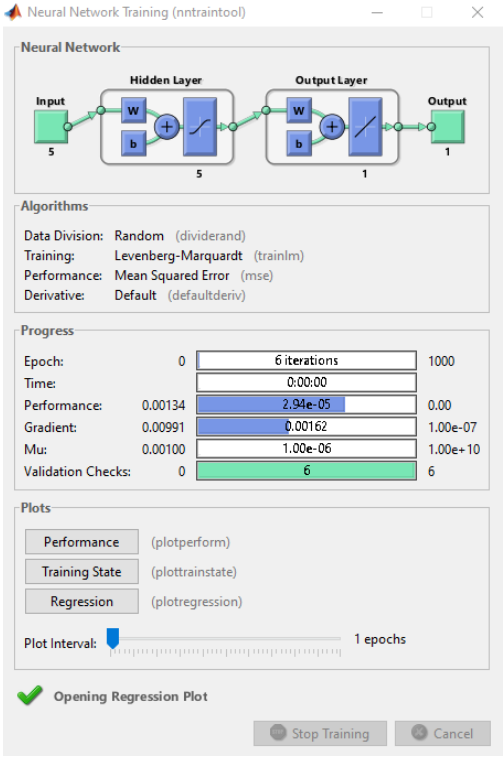



## 6. *View Weigts*

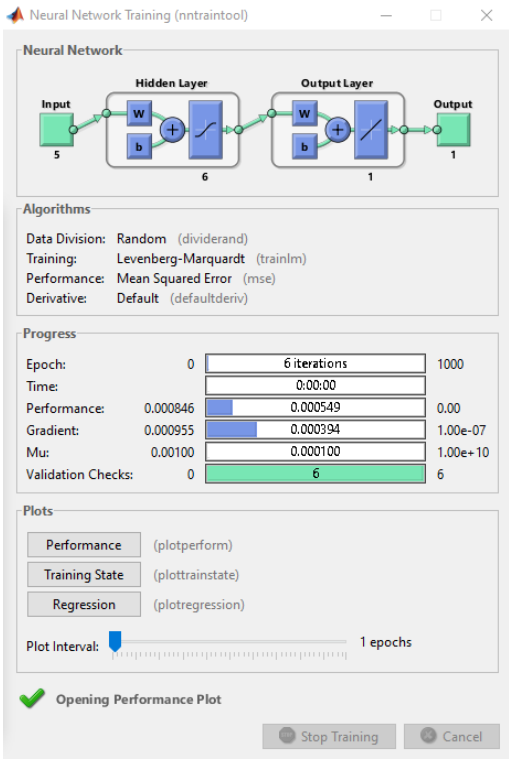


7. *Neural Network Training* (*nntraintool*)

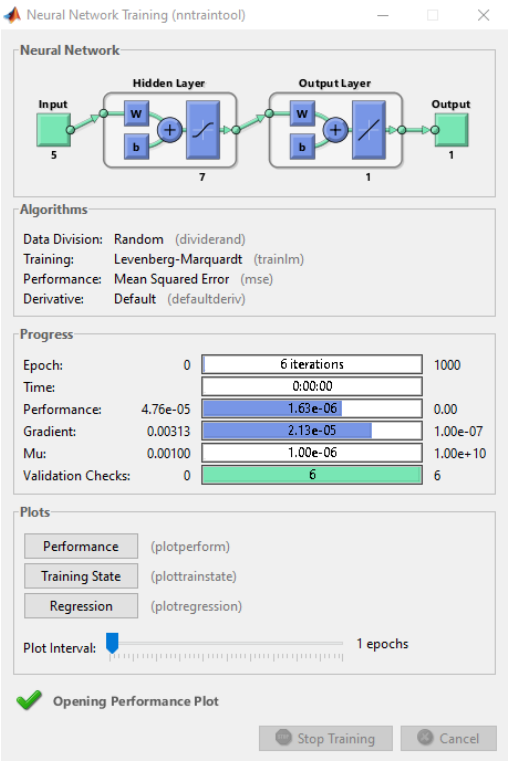
   a) Arsitektur 5-5-1



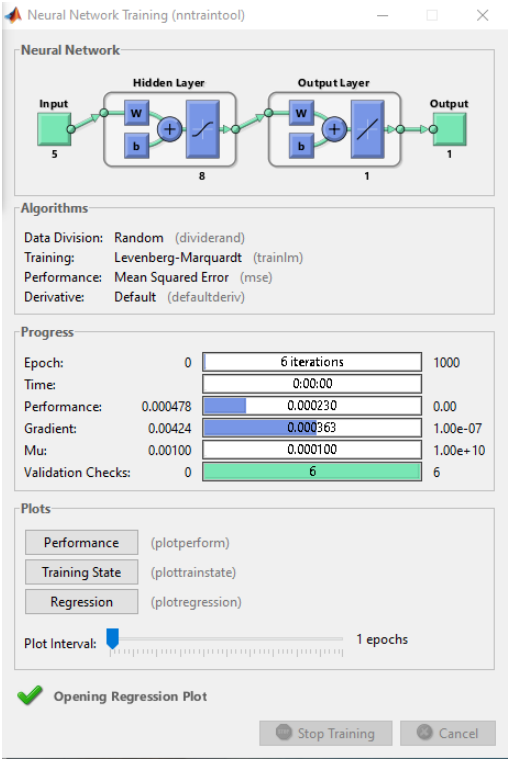
b) Model 5-6-1


Neural Network Training (nntraintool)
Neural Network
Input
5
Hidden Layer
W
b
6
Output Layer
W
b
1
Output
1
Algorithms
Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Derivative: Default (defaultderiv)
Progress
Epoch: 0 6 iterations 1000
Time: 0:00:00
Performance: 0.000846 0.000549 0.00
Gradient: 0.000955 0.000394 1.00e-07
Mu: 0.00100 0.000100 1.00e+10
Validation Checks: 0 6 6
Plots
Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)
Plot Interval: 1 epochs
Opening Performance Plot
Stop Training
Cancel

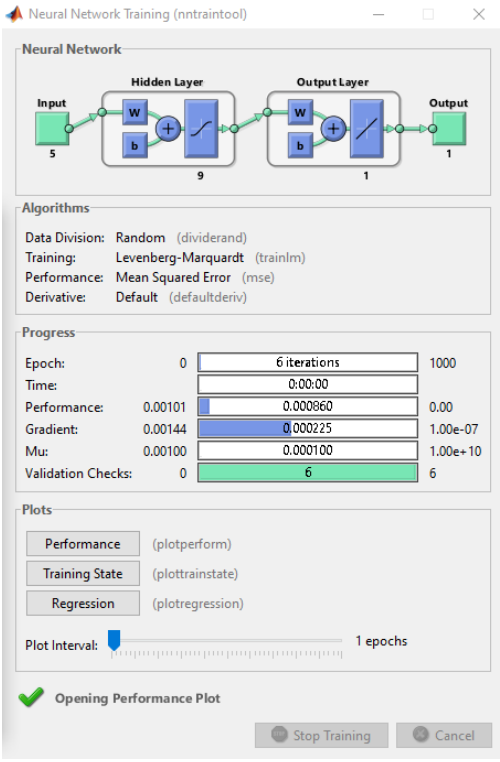
c) Model 5-7-1


Neural Network Training (nntraintool)
Neural Network
Input
5
Hidden Layer
W
b
7
Output Layer
W
b
1
Output
1
Algorithms
Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Derivative: Default (defaultderiv)
Progress
Epoch: 0 6 iterations 1000
Time: 0:00:00
Performance: 4.76e-05 1.63e-06 0.00
Gradient: 0.00313 2.13e-05 1.00e-07
Mu: 0.00100 1.00e-06 1.00e+10
Validation Checks: 0 6 6
Plots
Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)
Plot Interval: 1 epochs
Opening Performance Plot
Stop Training
Cancel

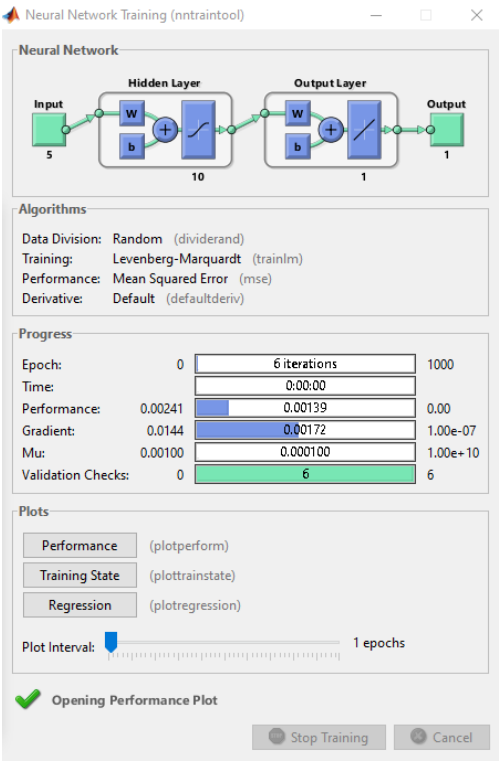
d) Model 5-8-1


Neural Network Training (nntraintool)
Neural Network
Input
5
Hidden Layer
W
b
8
Output Layer
W
b
1
Output
1
Algorithms
Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Derivative: Default (defaultderiv)
Progress
Epoch: 0 6 iterations 1000
Time: 0:00:00
Performance: 0.000478 0.000230 0.00
Gradient: 0.00424 0.000363 1.00e-07
Mu: 0.00100 0.000100 1.00e+10
Validation Checks: 0 6 6
Plots
Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)
Plot Interval: 1 epochs
Opening Regression Plot
Stop Training
Cancel

e) Model 5-9-1

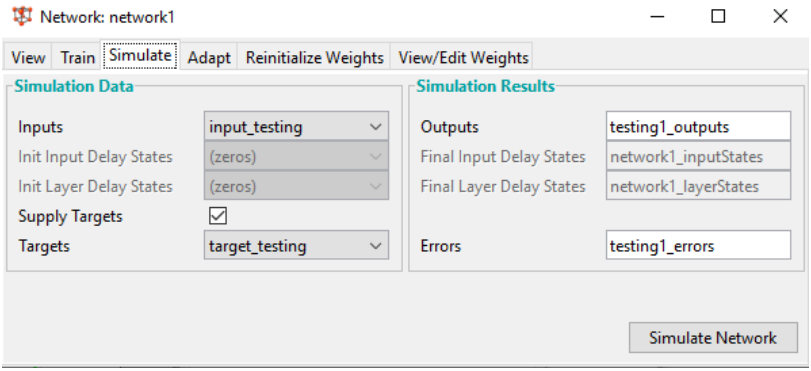

Neural Network Training (nntraintool)
Neural Network
Input
5
Hidden Layer
W
b
9
Output Layer
W
b
1
Output
1
Algorithms
Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Derivative: Default (defaultderiv)
Progress
Epoch: 0 6 iterations 1000
Time: 0:00:00
Performance: 0.00101 0.000860 0.00
Gradient: 0.00144 0.000225 1.00e-07
Mu: 0.00100 0.000100 1.00e+10
Validation Checks: 0 6 6
Plots
Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)
Plot Interval: 1 epochs
Opening Performance Plot
Stop Training
Cancel

f) Model 5-10-1


Neural Network Training (nntraintool)
Neural Network
Input
5
Hidden Layer
W
b
10
Output Layer
W
b
1
Output
1
Algorithms
Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Derivative: Default (defaultderiv)
Progress
Epoch: 0 6 iterations 1000
Time: 0:00:00
Performance: 0.00241 0.00139 0.00
Gradient: 0.0144 0.00172 1.00e-07
Mu: 0.00100 0.000100 1.00e+10
Validation Checks: 0 6 6
Plots
Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)
Plot Interval: 1 epochs
Opening Performance Plot
Stop Training
Cancel

**Lampiran 6** Proses Pengujian MLP menggunakan MATLAB R2013a

*Simulate Network*



**Lampiran 7** Hasil Pengujian MLP

1. Arsitektur 5-5-1

| No | Target | Hasil Prediksi | *Error* | *Error*^2 | %*Error* |
|---|---|---|---|---|---|
| 1 | 0,54936 | 0,58578 | -0,03642 | 0,001326416 | 6,629532547 |
| 2 | 0,55994 | 0,61441 | -0,05447 | 0,002966981 | 9,727827982 |
| 3 | 0,84624 | 0,63927 | 0,20697 | 0,042836581 | 24,45760068 |
| 4 | 0,86974 | 0,77802 | 0,09172 | 0,008412558 | 10,54568032 |
| 5 | 0,9 | 0,78233 | 0,11767 | 0,013846229 | 13,07444444 |
| TOTAL | | | | 0,069388766 | 64,43508597 |
| MSE | | | | 0,013877753 | |
| MAPE | | | | | 12,88701719 |

2. Arsitektur 5-6-1

| No | Target | Hasil Prediksi | *Error* | *Error*^2 | %*Error* |
|---|---|---|---|---|---|
| 1 | 0,54936 | 0,54806 | 0,0013 | 1,69E-06 | 0,236638998 |
| 2 | 0,55994 | 0,55362 | 0,00632 | 3,99424E-05 | 1,12869236 |
| 3 | 0,84624 | 0,55646 | 0,28978 | 0,083972448 | 34,24324069 |
| 4 | 0,86974 | 0,59401 | 0,27573 | 0,076027033 | 31,70257778 |
| 5 | 0,9 | 0,59327 | 0,30673 | 0,094083293 | 34,08111111 |
| TOTAL | | | | 0,254124407 | 101,3922609 |
| MSE | | | | 0,050824881 | |
| MAPE | | | | | 20,27845219 |

3. Arsitektur 5-7-1

| No | Target | Hasil Prediksi | *Error* | *Error*^2 | %*Error* |
|---|---|---|---|---|---|
| 1 | 0,54936 | 0,57927 | -0,02991 | 0,000894608 | 5,444517256 |
| 2 | 0,55994 | 0,61918 | -0,05924 | 0,003509378 | 10,57970497 |
| 3 | 0,84624 | 0,66062 | 0,18562 | 0,034454784 | 21,93467574 |
| 4 | 0,86974 | 0,71935 | 0,15039 | 0,022617152 | 17,29137443 |
| 5 | 0,9 | 0,61146 | 0,28854 | 0,083255332 | 32,06 |
| TOTAL | | | | 0,144731254 | 87,3102724 |
| MSE | | | | 0,028946251 | |
| MAPE | | | | | 17,46205448 |

4. Arsitektur 5-8-1

| No | Target | Hasil Prediksi | *Error* | *Error*^2 | %*Error* |
|---|---|---|---|---|---|
| 1 | 0,54936 | 0,57043 | -0,02107 | 0,000443945 | 3,835372069 |
| 2 | 0,55994 | 0,5851 | -0,02516 | 0,000633026 | 4,493338572 |
| 3 | 0,84624 | 0,59416 | 0,25208 | 0,063544326 | 29,78823974 |
| 4 | 0,86974 | 0,59036 | 0,27938 | 0,078053184 | 32,12224343 |
| 5 | 0,9 | 0,83949 | 0,06051 | 0,00366146 | 6,723333333 |
| TOTAL | | | | 0,146335941 | 76,96252715 |
| MSE | | | | 0,029267188 | |
| MAPE | | | | | 15,39250543 |

5. Arsitektur 5-9-1

| No | Target | Hasil Prediksi | *Error* | *Error*^2 | %*Error* |
|---|---|---|---|---|---|
| 1 | 0,54936 | 0,57164 | -0,02228 | 0,000496398 | 4,055628368 |
| 2 | 0,55994 | 0,59966 | -0,03972 | 0,001577678 | 7,093617173 |
| 3 | 0,84624 | 0,62784 | 0,2184 | 0,04769856 | 25,80828134 |
| 4 | 0,86974 | 0,86161 | 0,00813 | 6,60969E-05 | 0,934762113 |
| 5 | 0,9 | 0,97234 | -0,07234 | 0,005233076 | 8,037777778 |
| TOTAL | | | | 0,055071809 | 45,93006677 |
| MSE | | | | 0,011014362 | |
| MAPE | | | | | 9,186013354 |

6. Arsitektur 5-10-1

| No | Target | Hasil Prediksi | *Error* | *Error*^2 | %*Error* |
|---|---|---|---|---|---|
| 1 | 0,54936 | 0,55903 | -0,00967 | 9,35089E-05 | 1,760230086 |
| 2 | 0,55994 | 0,5743 | -0,01436 | 0,00020621 | 2,564560489 |
| 3 | 0,84624 | 0,58621 | 0,26003 | 0,067615601 | 30,72768954 |
| 4 | 0,86974 | 0,53647 | 0,33327 | 0,111068893 | 38,31834801 |
| 5 | 0,9 | 0,5676 | 0,3324 | 0,11048976 | 36,93333333 |
| TOTAL | | | | 0,289473972 | 110,3041615 |
| MSE | | | | 0,057894794 | |
| MAPE | | | | | 22,06083229 |

**Lampiran 8** *Syntax* Program MLP

*Plot* Perbandingan Nilai Prediksi dan Nilai Asli Tahun 1991-2022

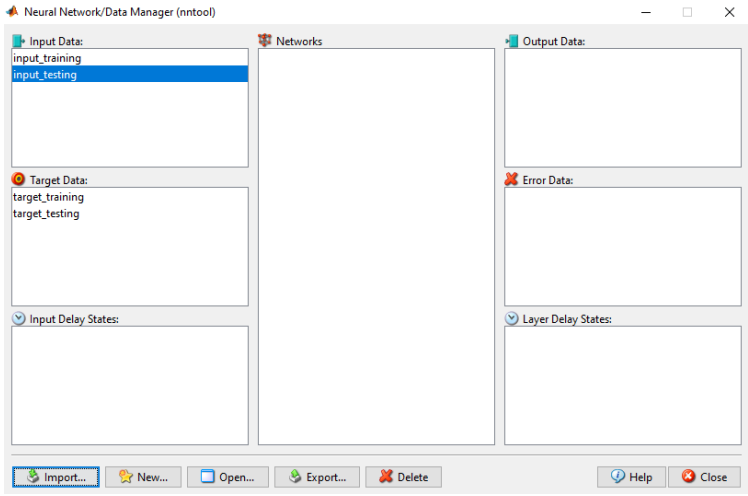
```
plot (nilai_prediksi,'bo-','LineWidth',2)
hold on
plot (nilai_asli,'ro-','LineWidth',2)
grid on
title (['Grafik Perbandingan Nilai Prediksi dengan Nilai Asli Jumlah Penduduk Tahun 1996-2022 Menggunakan Metode MLP'])
xlabel ('Tahun 1996-2022')
ylabel ('Jumlah Penduduk')
legend ('Nilai Prediksi','Nilai Asli')
```

Prediksi Tahun 2023-2026

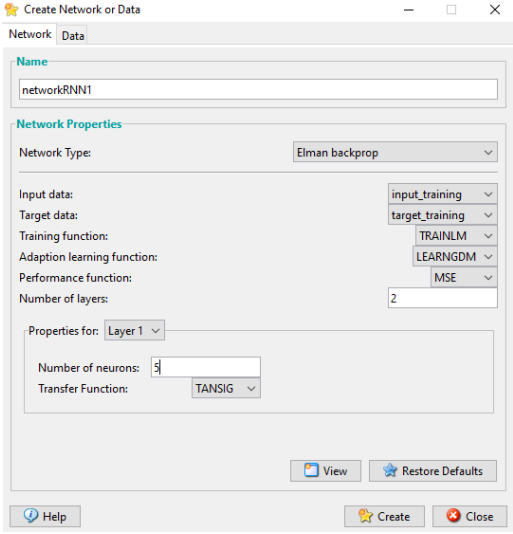
```
>> prediksi_2023=sim(network5,hasil_testing)

prediksi_2023 =

    0.9875

>> prediksi_2024=sim(network5,hasil_testing)

prediksi_2024 =

    1.0064

>> prediksi_2025=sim(network5,hasil_testing)

prediksi_2025 =

    0.9816

>> prediksi_2026=sim(network5,hasil_testing)

prediksi_2026 =

    0.9991
```

**Lampiran 9** Proses Pelatihan RNN menggunakan MATLAB R2013a

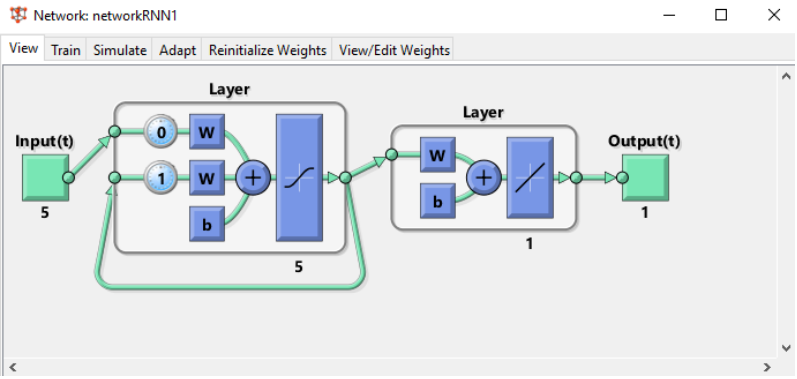
1. *Neural Network Tool* (*nntool*)



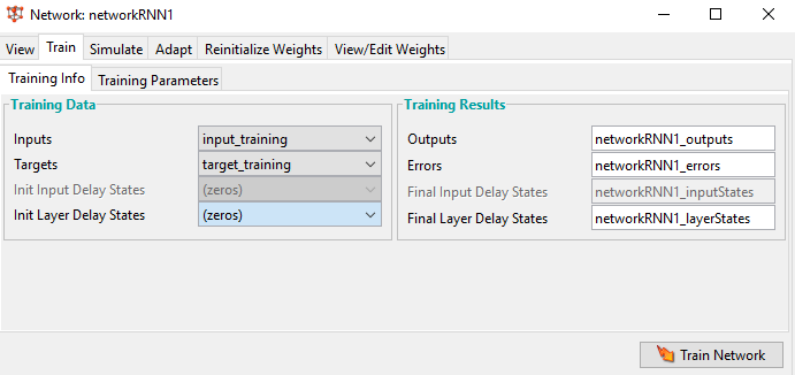

2. *Create a Network*



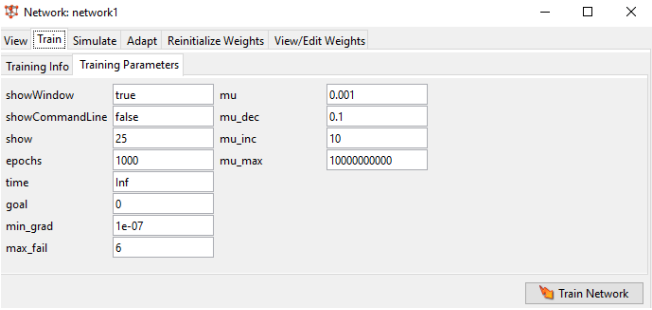
*3.* Arsitektur Jaringan




*4. Training Info*

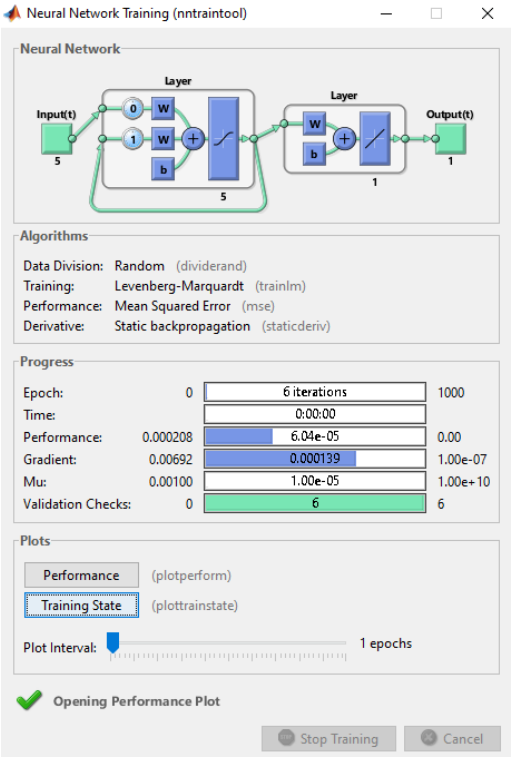


5. *Training Parameters*

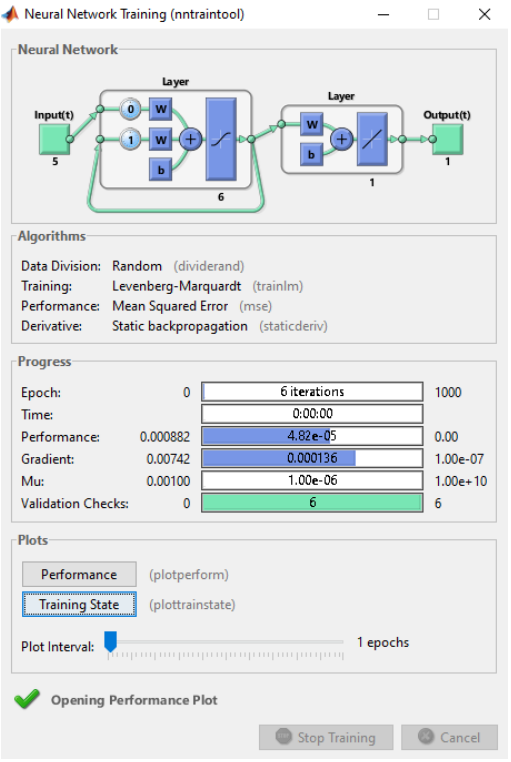



6. *Neural Network Training* (*nntraintool*)

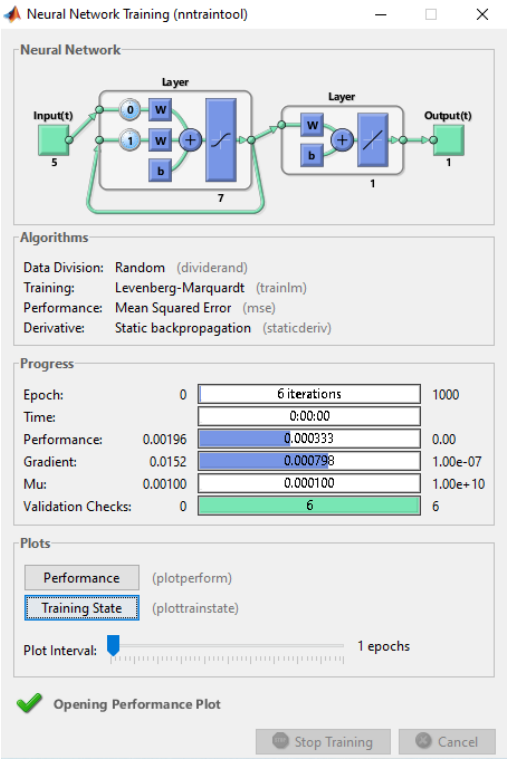
a) Arsitektur 5-5-1



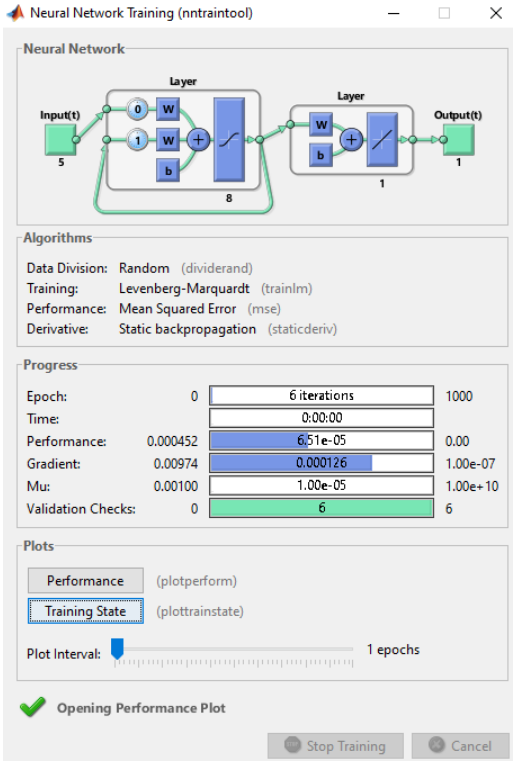
b) Model 5-6-1


Neural Network Training (nntraintool)
Neural Network
Input(t)
5
Layer
0
1
W
W
b
6
Layer
W
b
1
Output(t)
1
Algorithms
Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Derivative: Static backpropagation (staticderiv)
Progress
Epoch: 0 6 iterations 1000
Time: 0:00:00
Performance: 0.000882 4.82e-05 0.00
Gradient: 0.00742 0.000136 1.00e-07
Mu: 0.00100 1.00e-06 1.00e+10
Validation Checks: 0 6 6
Plots
Performance (plotperform)
Training State (plottrainstate)
Plot Interval: 1 epochs
Opening Performance Plot
Stop Training
Cancel

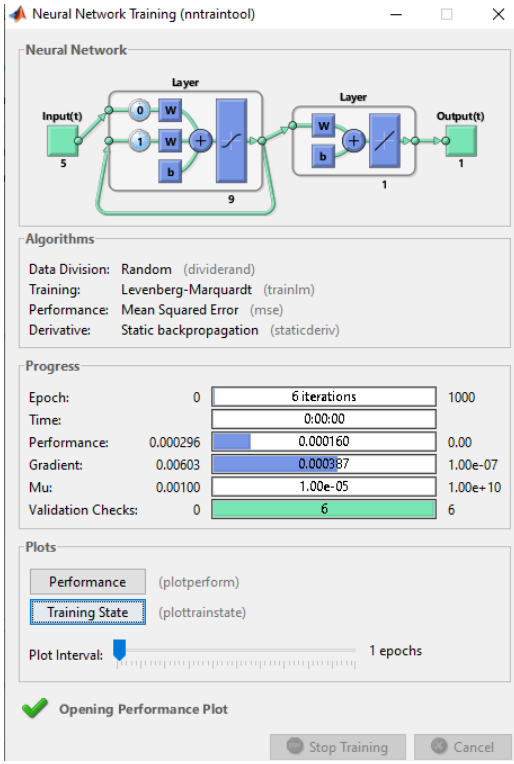
c) Model 5-7-1


Neural Network Training (nntraintool)
Neural Network
Input(t)
5
Layer
0
1
W
W
b
7
Layer
W
b
1
Output(t)
1
Algorithms
Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Derivative: Static backpropagation (staticderiv)
Progress
Epoch: 0 6 iterations 1000
Time: 0:00:00
Performance: 0.00196 0.000333 0.00
Gradient: 0.0152 0.000798 1.00e-07
Mu: 0.00100 0.000100 1.00e+10
Validation Checks: 0 6 6
Plots
Performance (plotperform)
Training State (plottrainstate)
Plot Interval: 1 epochs
Opening Performance Plot
Stop Training
Cancel

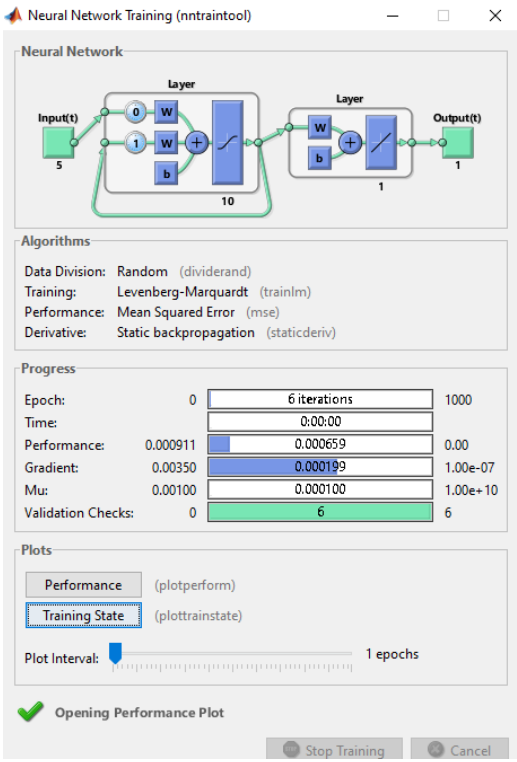
d) Model 5-8-1


Neural Network Training (nntraintool)
Neural Network
Layer
Input(t)
5
0
1
W
W
b
8
Layer
W
b
1
Output(t)
1
Algorithms
Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Derivative: Static backpropagation (staticderiv)
Progress
Epoch: 0 6 iterations 1000
Time: 0:00:00
Performance: 0.000452 6.51e-05 0.00
Gradient: 0.00974 0.000126 1.00e-07
Mu: 0.00100 1.00e-05 1.00e+10
Validation Checks: 0 6 6
Plots
Performance (plotperform)
Training State (plottrainstate)
Plot Interval: 1 epochs
Opening Performance Plot
Stop Training
Cancel

e) Model 5-9-1

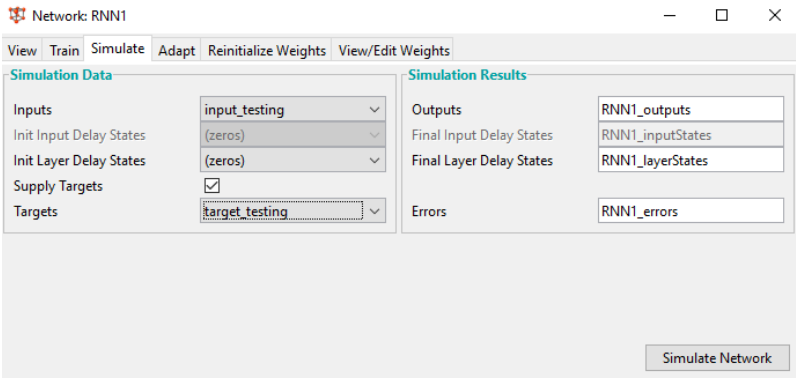

Neural Network Training (nntraintool)
Neural Network
Input(t)
5
Layer
0
1
W
W
b
9
Layer
W
b
1
Output(t)
1
Algorithms
Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Derivative: Static backpropagation (staticderiv)
Progress
Epoch: 0 6 iterations 1000
Time: 0:00:00
Performance: 0.000296 0.000160 0.00
Gradient: 0.00603 0.000387 1.00e-07
Mu: 0.00100 1.00e-05 1.00e+10
Validation Checks: 0 6 6
Plots
Performance (plotperform)
Training State (plottrainstate)
Plot Interval: 1 epochs
Opening Performance Plot
Stop Training
Cancel

f) Model 5-10-1


Neural Network Training (nntraintool)
Neural Network
Input(t)
5
Layer
10
Layer
1
Output(t)
1
Algorithms
Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Derivative: Static backpropagation (staticderiv)
Progress
Epoch: 0 6 iterations 1000
Time: 0:00:00
Performance: 0.000911 0.000659 0.00
Gradient: 0.00350 0.000199 1.00e-07
Mu: 0.00100 0.000100 1.00e+10
Validation Checks: 0 6 6
Plots
Performance (plotperform)
Training State (plottrainstate)
Plot Interval: 1 epochs
Opening Performance Plot
Stop Training
Cancel

**Lampiran 10** Proses Pengujian RNN menggunakan MATLAB R2013a

*Simulate Network*



**Lampiran 11** Hasil Pengujian RNN

1. Arsitektur 5-5-1

| No | Target | Hasil Prediksi | *Error* | *Error*^2 | %*Error* |
|---|---|---|---|---|---|
| 1 | 0,54936 | 0,54406 | 0,0053 | 2,809E-05 | 0,964758992 |
| 2 | 0,55994 | 0,55049 | 0,00945 | 8,93025E-05 | 1,687680823 |
| 3 | 0,84624 | 0,55517 | 0,29107 | 0,084721745 | 34,39567971 |
| 4 | 0,86974 | 0,71276 | 0,15698 | 0,02464272 | 18,04907214 |
| 5 | 0,9 | 0,69969 | 0,20031 | 0,040124096 | 22,25666667 |
| TOTAL | | | | 0,149605954 | 77,35385833 |
| MSE | | | | 0,029921191 | |
| MAPE | | | | | 15,47077167 |

2. Arsitektur 5-6-1

| No | Target | Hasil Prediksi | *Error* | *Error*^2 | %*Error* |
|---|---|---|---|---|---|
| 1 | 0,54936 | 0,55531 | -0,00595 | 3,54025E-05 | 1,083078491 |
| 2 | 0,55994 | 0,56929 | -0,00935 | 8,74225E-05 | 1,669821767 |
| 3 | 0,84624 | 0,58133 | 0,26491 | 0,070177308 | 31,3043581 |
| 4 | 0,86974 | 0,76134 | 0,1084 | 0,01175056 | 12,46349484 |
| 5 | 0,9 | 0,72765 | 0,17235 | 0,029704523 | 19,15 |
| TOTAL | | | | 0,111755216 | 65,6707532 |
| MSE | | | | 0,022351043 | |
| MAPE | | | | | 13,13415064 |

3. Arsitektur 5-7-1

| No | Target | Hasil Prediksi | *Error* | *Error*^2 | %*Error* |
|---|---|---|---|---|---|
| 1 | 0,54936 | 0,55383 | -0,00447 | 1,99809E-05 | 0,813674093 |
| 2 | 0,55994 | 0,56919 | -0,00925 | 8,55625E-05 | 1,65196271 |
| 3 | 0,84624 | 0,58259 | 0,26365 | 0,069511323 | 31,15546417 |
| 4 | 0,86974 | 0,92351 | -0,05377 | 0,002891213 | 6,182307356 |
| 5 | 0,9 | 1,0065 | -0,1065 | 0,01134225 | 11,83333333 |
| TOTAL | | | | 0,083850329 | 51,63674166 |
| MSE | | | | 0,016770066 | |
| MAPE | | | | | 10,32734833 |

4. Arsitektur 5-8-1

| No | Target | Hasil Prediksi | *Error* | *Error*^2 | %*Error* |
|---|---|---|---|---|---|
| 1 | 0,54936 | 0,57536 | -0,026 | 0,000676 | 4,732779962 |
| 2 | 0,55994 | 0,5994 | -0,03946 | 0,001557092 | 7,047183627 |
| 3 | 0,84624 | 0,62193 | 0,22431 | 0,050314976 | 26,50666478 |
| 4 | 0,86974 | 0,50614 | 0,3636 | 0,13220496 | 41,80559707 |
| 5 | 0,9 | 0,45841 | 0,44159 | 0,195001728 | 49,06555556 |
| TOTAL | | | | 0,379754756 | 129,157781 |
| MSE | | | | 0,075950951 | |
| MAPE | | | | | 25,8315562 |

5. Arsitektur 5-9-1

| No | Target | Hasil Prediksi | *Error* | *Error*^2 | %*Error* |
|---|---|---|---|---|---|
| 1 | 0,54936 | 0,5774 | -0,02804 | 0,000786242 | 5,104121159 |
| 2 | 0,55994 | 0,60291 | -0,04297 | 0,001846421 | 7,674036504 |
| 3 | 0,84624 | 0,62657 | 0,21967 | 0,048254909 | 25,95835697 |
| 4 | 0,86974 | 0,85436 | 0,01538 | 0,000236544 | 1,768344563 |
| 5 | 0,9 | 0,92107 | -0,02107 | 0,000443945 | 2,341111111 |
| TOTAL | | | | 0,051568061 | 42,8459703 |
| MSE | | | | 0,010313612 | |
| MAPE | | | | | 8,569194061 |

6. Arsitektur 5-10-1

| No | Target | Hasil Prediksi | *Error* | *Error*^2 | %*Error* |
|---|---|---|---|---|---|
| 1 | 0,54936 | 0,55321 | -0,00385 | 1,48225E-05 | 0,700815494 |
| 2 | 0,55994 | 0,56055 | -0,00061 | 3,721E-07 | 0,108940244 |
| 3 | 0,84624 | 0,56476 | 0,28148 | 0,07923099 | 33,26243146 |
| 4 | 0,86974 | 0,66462 | 0,20512 | 0,042074214 | 23,5840596 |
| 5 | 0,9 | 0,54902 | 0,35098 | 0,12318696 | 38,99777778 |
| TOTAL | | | | 0,24450736 | 96,65402458 |
| MSE | | | | 0,048901472 | |
| MAPE | | | | | 19,33080492 |

**Lampiran 12** *Syntax* Prediksi Program RNN

*Plot* Perbandingan Nilai Prediksi dan Nilai Asli Tahun 1991-2022

```
plot (nilai_prediksi,'bo-','LineWidth',2)
hold on
plot (nilai_asli,'ro-','LineWidth',2)
grid on
title (['Grafik Perbandingan Nilai Prediksi dengan Nilai Asli Jumlah Penduduk Tahun 1996-2022 Menggunakan Metode RNN'])
xlabel ('Tahun 1996-2022')
ylabel ('Jumlah Penduduk')
legend ('Nilai Prediksi','Nilai Asli')
```

Prediksi Tahun 2023-2026

```
>> prediksi2024=sim(networkRNN5,hasil_testingRNN)

prediksi2024 =

    0.9278

>> prediksi2025=sim(networkRNN5,hasil_testingRNN)

prediksi2025 =

    0.9196

>> prediksi2026=sim(networkRNN5,hasil_testingRNN)

prediksi2026 =

    0.9217

>>
```

**Lampiran 13** Perhitungan Manual *Recurrent Neural Network*

Berdasarkan pelatihan jaringan RNN tipe Elman algoritma *backpropagation* yang dihitung menggunakan MATLAB R2013a didapatkan nilai-nilai bobot sebagai berikut.

```
>> networkRNN5.iw{1,1}

ans =

   -0.2180    0.6477    0.5311    0.4438    0.4725
   -0.4975   -0.2475    0.2651    0.5369   -0.1251
   -0.3885   -0.1805    0.0049    0.0542   -0.9508
   -0.4830    0.2092    0.3540   -0.6044   -0.5641
   -0.1486    0.5084   -0.8196   -0.7127   -0.3724
    0.1070    0.0922   -0.4879   -0.3356   -0.5839
    0.2579    0.3163    1.2073    1.1808    0.7105
    0.5163   -0.2824   -0.4581   -0.4753   -0.3845
   -3.0134   -1.6108   -2.4109   -0.2458   -0.0491

>> networkRNN5.lw{1,1}

ans =

    0.1288    0.1305   -0.2906    0.2150    0.7273   -0.5375    0.7103    0.2604   -0.7741
   -0.3783   -0.5978   -0.5160   -0.7109    0.0478   -0.3338   -0.3952    0.5249   -0.0572
   -0.3889    0.7541   -0.5962   -0.8003   -0.3245   -0.1112    0.3269   -0.2887   -0.1404
    0.2293    0.7435   -0.1510   -0.3533   -0.7724    0.0532   -0.4131    0.5494   -0.0765
   -0.4336    0.5870   -0.7495    0.0570    0.2050   -0.0786    0.3174    0.3239    0.4990
    0.5200   -0.3836    0.1579    0.2476    0.4469    0.6017    0.3128   -0.7908   -0.2846
    0.6538    0.1278   -0.0394   -0.1251   -0.1037    0.0245   -0.5852    0.1384    0.3857
    0.3214   -0.6665    0.2735    0.4466   -0.5711    0.6192   -0.3423   -0.1580   -0.0400
   -0.2283   -0.1093    0.2923    0.3193   -0.3415    0.2014   -0.4035    0.6083   -0.6789

>> networkRNN5.lw{2,1}

ans =

   -0.5509    0.4360   -0.7978   -0.9917    0.6006    1.3393   -1.8139   -0.0920   -4.0542
```

```
>> networkRNN5.b{1}

ans =

    1.4710
    1.3895
    1.4687
    1.3316
    0.0689
   -0.4909
    0.7184
    1.3026
   -5.0508

>> networkRNN5.b{2}

ans =

    1.0366
```

Berikut merupakan perhitungan manual pelatihan jaringan RNN tipe Elman arsitektur 5-9-1.

Fase Perambatan Maju

Setelah didapatkan nilai bobot, substitusikan nilai bobot ke dalam persamaan 2.11 diperoleh:

1. Proses keluaran dari *input layer* 1 menuju *hidden layer*

$$z_{in_{(1)1}} = b_1 + x_1 w_{11(x)} + x_2 w_{21(x)} + x_3 w_{31(x)} + x_4 w_{41(x)} + x_5 w_{51(x)}$$

$$= 1{,}4710 + (0{,}48494)(0{,}218) + (0{,}49967)(0{,}6477) + (0{,}51317)(0{,}5311) + (0{,}52583)(0{,}4438) + (0{,}53785)(0{,}4725) = 2{,}4489$$

kemudian sinyal $z_{in_{(1)1}}$ dilakukan aktivasi dengan fungsi sigmoid bipolar (Persamaan 2.3) yaitu sebagai berikut.

$$z(1)_1 = f\left(z_{in(1)_1}\right) = \frac{1 - e^{-(2{,}4489)}}{1 + e^{-(2{,}4489)}} = 0{,}8409$$

2. Proses keluaran dari *input layer* 2 menuju *hidden layer*

$$z_{in_{(1)2}} = b_2 + x_1 w_{12(x)} + x_2 w_{22(x)} + x_3 w_{32(x)} + x_4 w_{42(x)} + x_5 w_{52(x)}$$
$$= 1{,}3895 + (0{,}48494)(-0{,}4975) + (0{,}49967)(-0{,}2475) + (0{,}51317)(0{,}2651) + (0{,}52583)(0{,}5369) + (0{,}53785)(-0{,}1251) = 1{,}3756$$
$$z(1)_2 = f\left(z_{in(1)_2}\right) = \frac{1 - e^{-(1{,}3756)}}{1 + e^{-(1{,}3756)}} = 0{,}5966$$

3. Proses keluaran dari *input layer* 3 menuju *hidden layer*

$$z_{in_{(1)3}} = b_3 + x_1 w_{13(x)} + x_2 w_{23(x)} + x_3 w_{33(x)} + x_4 w_{43(x)} + x_5 w_{53(x)}$$
$$= 1{,}4687 + (0{,}48494)(-0{,}3885) + (0{,}49967)(-0{,}1805) + (0{,}51317)(0{,}0049) + (0{,}52583)(0{,}0542) + (0{,}53785)(-0{,}9508) = 0{,}7097$$
$$z(1)_3 = f\left(z_{in(1)_3}\right) = \frac{1 - e^{-(0{,}7097)}}{1 + e^{-(0{,}7097)}} = 0{,}3407$$

4. Proses keluaran dari *input layer* 4 menuju *hidden layer*

$$z_{in_{(1)4}} = b_4 + x_1 w_{14(x)} + x_2 w_{24(x)} + x_3 w_{34(x)} + x_4 w_{44(x)} + x_5 w_{54(x)}$$
$$= 1{,}3316 + (0{,}48494)(-0{,}483) + (0{,}49967)(0{,}2092) + (0{,}51317)(0{,}354) + (0{,}52583)(-0{,}6044) + (0{,}53785)(-0{,}5641) = 0{,}7623$$
$$z(1)_4 = f\left(z_{in(1)_4}\right) = \frac{1 - e^{-(0{,}7623)}}{1 + e^{-(0{,}7623)}} = 0{,}3637$$

5. Proses keluaran dari *input layer* 5 menuju *hidden layer*

$$z_{in_{(1)5}} = b_5 + x_1 w_{15(x)} + x_2 w_{25(x)} + x_3 w_{35(x)} + x_4 w_{45(x)} + x_5 w_{55(x)}$$
$$= 0{,}0689 + (0{,}48494)(-0{,}1486) + (0{,}49967)(-0{,}5084) + (0{,}51317)(0{,}8197) + (0{,}52583)(-0{,}7127) + (0{,}53785)(-0{,}3724) = -0{,}7449$$
$$z(1)_5 = f\left(z_{in(1)_5}\right) = \frac{1 - e^{-(-0{,}7449)}}{1 + e^{-(-0{,}7449)}} = -0{,}3561$$

6. Proses keluaran dari *input layer* 6 menuju *hidden layer*

$$z_{in_{(1)6}} = b_6 + x_1 w_{16(x)} + x_2 w_{26(x)} + x_3 w_{36(x)} + x_4 w_{46(x)} + x_5 w_{56(x)}$$
$$= -0{,}4909 + (0{,}48494)(0{,}107) + (0{,}49967)(0{,}0922) + (0{,}51317)(-0{,}4879) + (0{,}52583)(-0{,}3356) + (0{,}53785)(-0{,}5839) = -1{,}1338$$
$$z(1)_6 = f\left(z_{in(1)_6}\right) = \frac{1 - e^{-(-1{,}1338)}}{1 + e^{-(-1{,}1338)}} = -0{,}5131$$

7. Proses keluaran dari *input layer* 7 menuju *hidden layer*

$$z_{in_{(1)7}} = b_7 + x_1 w_{17(x)} + x_2 w_{27(x)} + x_3 w_{37(x)} + x_4 w_{47(x)} + x_5 w_{57(x)}$$
$$= 0{,}7184 + (0{,}48494)(0{,}2579) + (0{,}49967)(0{,}3163) + (0{,}51317)(1{,}2073) + (0{,}52583)(1{,}1808) + (0{,}53785)(0{,}7105) = 2{,}6241$$
$$z(1)_7 = f\left(z_{in(1)_7}\right) = \frac{1 - e^{-(2{,}6241)}}{1 + e^{-(2{,}6241)}} = 0{,}8648$$

8. Proses keluaran dari *input layer* 8 menuju *hidden layer*

$$z_{in_{(1)8}} = b_8 + x_1 w_{18(x)} + x_2 w_{28(x)} + x_3 w_{38(x)} + x_4 w_{48(x)} + x_5 w_{58(x)}$$
$$= 1{,}3026 + (0{,}48494)(0{,}5163) + (0{,}49967)(-0{,}2824) + (0{,}51317)(-0{,}4581) + (0{,}52583)(-0{,}4753) + (0{,}53785)(-0{,}3845) = 0{,}7201$$
$$z(1)_8 = f\left(z_{in(1)_8}\right) = \frac{1 - e^{-(0{,}7201)}}{1 + e^{-(0{,}7201)}} = 0{,}3452$$

9. Proses keluaran dari *input layer* 9 menuju *hidden layer*

$$z_{in_{(1)9}} = b_9 + x_1 w_{19(x)} + x_2 w_{29(x)} + x_3 w_{39(x)} + x_4 w_{49(x)} + x_5 w_{59(x)}$$
$$= -5{,}0508 + (0{,}48494)(-3{,}0133) + (0{,}49967)(-1{,}6108) + (0{,}51317)(-2{,}4109) + (0{,}52583)(-0{,}2458) + (0{,}53785)(-0{,}0491) = -8{,}7098$$
$$z(1)_9 = f\left(z_{in(1)_9}\right) = \frac{1 - e^{-(-8{,}7098)}}{1 + e^{-(-8{,}7098)}} = -0{,}9997$$

Selanjutnya, sesuai persamaan 2.16 didapatkan:

1. Proses keluaran neuron 1 pada *hidden layer* dengan neuron tambahan ke lapisan *output*

$$z_{in_1} = b_1 + x_1 w_{11(x)} + x_2 w_{21(x)} + x_3 w_{31(x)} + x_4 w_{41(x)} + x_5 w_{51(x)} + u_1 w_{11(u)} + u_2 w_{21(u)} + u_3 w_{31(u)} + u_4 w_{41(u)} + u_5 w_{51(u)} + u_6 w_{61(u)} + u_7 w_{71(u)} + u_8 w_{81(u)} + u_9 w_{91(u)}$$

disederhanakan menjadi:

$$z_{in_1} = z_{in(1)_1} + u_1 w_{11(u)} + u_2 w_{21(u)} + u_3 w_{31(u)} + u_4 w_{41(u)} + u_5 w_{51(u)} + u_6 w_{61(u)} + u_7 w_{71(u)} + u_8 w_{81(u)} + u_9 w_{91(u)}$$

$$= 2{,}4489 + (0{,}8409)(0{,}1288) + (0{,}5966)(0{,}1305) + (0{,}3407)(-0{,}2906) + (0{,}3637)(0{,}215) + (-0{,}3561)(0{,}7273) + (-0{,}5131)(-0{,}5375) + (0{,}8648)(0{,}7103) + (0{,}3452)(0{,}2604) + (-0{,}9997)(-0{,}7741) = 4{,}1091$$

$$z_1 = f(z_{in_1}) = \frac{1 - e^{-(4{,}1091)}}{1 + e^{-(4{,}1091)}} = 0{,}9677$$

2. Proses keluaran neuron 2 pada *hidden layer* dengan neuron tambahan ke lapisan *output*

$$z_{in_2} = z_{in(1)_2} + u_1 w_{12(u)} + u_2 w_{22(u)} + u_3 w_{32(u)} + u_4 w_{42(u)} + u_5 w_{52(u)} + u_6 w_{62(u)} + u_7 w_{72(u)} + u_8 w_{82(u)} + u_9 w_{92(u)}$$

$$= 1{,}3756 + (0{,}8409)(-0{,}3783) + (0{,}5966)(-0{,}5978) + (0{,}3407)(-0{,}516) + (0{,}3637)(-0{,}7109) + (-0{,}3561)(0{,}0478) + (-0{,}5131)(-0{,}3338) + (0{,}8648)(-0{,}3952) + (0{,}3452)(0{,}5249) + (-0{,}9997)(-0{,}0572) = 0{,}3174$$

$$z_2 = f(z_{in_2}) = \frac{1 - e^{-(0{,}3174)}}{1 + e^{-(0{,}3174)}} = 0{,}1574$$

3. Proses keluaran neuron 3 pada *hidden layer* dengan neuron tambahan ke lapisan *output*

$$z_{in_3} = z_{in(1)_3} + u_1 w_{13(u)} + u_2 w_{23(u)} + u_3 w_{33(u)} + u_4 w_{43(u)} + u_5 w_{53(u)} + u_6 w_{63(u)} + u_7 w_{73(u)} + u_8 w_{83(u)} + u_9 w_{93(u)}$$

$$= 0{,}7097 + (0{,}8409)(-0{,}3889) + (0{,}5966)(0{,}7541) + (0{,}3407)(-0{,}5962) + (0{,}3637)(-0{,}8003) + (-0{,}3561)(-0{,}3245) + (-0{,}5131)(-0{,}1112) + (0{,}8648)(0{,}3269) + (0{,}3452)(-0{,}2887) + (-0{,}9997)(-0{,}1404) = 0{,}8343$$

$$z_3 = f(z_{in_3}) = \frac{1 - e^{-(0{,}8343)}}{1 + e^{-(0{,}8343)}} = 0{,}3945$$

4. Proses keluaran neuron 4 pada *hidden layer* dengan neuron tambahan ke lapisan *output*

$$z_{in_4} = z_{in(1)_4} + u_1 w_{14(u)} + u_2 w_{24(u)} + u_3 w_{34(u)} + u_4 w_{44(u)} + u_5 w_{54(u)} + u_6 w_{64(u)} + u_7 w_{74(u)} + u_8 w_{84(u)} + u_9 w_{94(u)}$$

$$= 0{,}7623 + (0{,}8409)(0{,}2293) + (0{,}5966)(0{,}7435) + (0{,}3407)(-0{,}151) + (0{,}3637)(-0{,}3533) + (-0{,}3561)(-0{,}7724) + (-0{,}5131)(0{,}0532) + (0{,}8648)(-0{,}4131) + (0{,}3452)(0{,}5494) + (-0{,}9997)(-0{,}0765) = 1{,}3754$$

$$z_4 = f(z_{in_4}) = \frac{1 - e^{-(1{,}3754)}}{1 + e^{-(1{,}3754)}} = 0{,}5965$$

5. Proses keluaran neuron 5 pada *hidden layer* dengan neuron tambahan ke lapisan *output*

$$z_{in_5} = z_{in(1)_5} + u_1 w_{15(u)} + u_2 w_{25(u)} + u_3 w_{35(u)} + u_4 w_{45(u)} + u_5 w_{55(u)} + u_6 w_{65(u)} + u_7 w_{75(u)} + u_8 w_{85(u)} + u_9 w_{95(u)}$$

$$= -0{,}7449 + (0{,}8409)(-0{,}4336) + (0{,}5966)(0{,}587) + (0{,}3407)(-0{,}7495) + (0{,}3637)(0{,}057) + (-0{,}3561)(0{,}205) + (-0{,}5131)(-0{,}0786) + (0{,}8648)(0{,}3174) + (0{,}3452)(0{,}3239) + (-0{,}9997)(0{,}499) = -1{,}1391$$

$$z_5 = f(z_{in_5}) = \frac{1 - e^{-(-1{,}1391)}}{1 + e^{-(-1{,}1391)}} = -0{,}515$$

6. Proses keluaran neuron 6 pada *hidden layer* dengan neuron tambahan ke lapisan *output*

$$z_{in_6} = z_{in(1)_6} + u_1 w_{16(u)} + u_2 w_{26(u)} + u_3 w_{36(u)} + u_4 w_{46(u)} + u_5 w_{56(u)} + u_6 w_{66(u)} + u_7 w_{76(u)} + u_8 w_{86(u)} + u_9 w_{96(u)}$$

$$= -1{,}1338 + (0{,}8409)(0{,}52) + (0{,}5966)(-0{,}3836) + (0{,}3407)(0{,}1579) + (0{,}3637)(0{,}2476) + (-0{,}3561)(0{,}4469) + (-0{,}5131)(0{,}6017) + (0{,}8648)(0{,}3128) + (0{,}3452)(-0{,}7908) + (-0{,}9997)(-0{,}2846) = -0{,}9673$$

$$z_6 = f(z_{in_6}) = \frac{1 - e^{-(-0{,}9763)}}{1 + e^{-(-0{,}9763)}} = -0{,}4492$$

7. Proses keluaran neuron 7 pada *hidden layer* dengan neuron tambahan ke lapisan *output*

$$z_{in_7} = z_{in(1)_7} + u_1 w_{17(u)} + u_2 w_{27(u)} + u_3 w_{37(u)} + u_4 w_{47(u)} + u_5 w_{57(u)} + u_6 w_{67(u)} + u_7 w_{77(u)} + u_8 w_{87(u)} + u_9 w_{97(u)}$$

$$= 2{,}6241 + (0{,}8409)(0{,}6538) + (0{,}5966)(0{,}1278) + (0{,}3407)(-0{,}0394) + (0{,}3637)(-0{,}1251) + (-0{,}3561)(-0{,}1037) + (-0{,}5131)(0{,}0245) + (0{,}8648)(-0{,}5852) + (0{,}3452)(0{,}1384) + (-0{,}9997)(0{,}3857) = 2{,}3717$$

$$z_7 = f(z_{in_7}) = \frac{1 - e^{-(2{,}3717)}}{1 + e^{-(2{,}3717)}} = 0{,}8293$$

8. Proses keluaran neuron 8 pada *hidden layer* dengan neuron tambahan ke lapisan *output*

$$z_{in_8} = z_{in(1)_8} + u_1 w_{18(u)} + u_2 w_{28(u)} + u_3 w_{38(u)} + u_4 w_{48(u)} + u_5 w_{58(u)} + u_6 w_{68(u)} + u_7 w_{78(u)} + u_8 w_{88(u)} + u_9 w_{98(u)}$$

$$= 0{,}7201 + (0{,}8409)(0{,}3214) + (0{,}5966)(-0{,}6665) + (0{,}3407)(0{,}2735) + (0{,}3637)(0{,}4466) + (-0{,}3561)(-0{,}5711) + (-0{,}5131)(0{,}6192) + (0{,}8648)(-0{,}3423) + (0{,}3452)(-0{,}158) + (-0{,}9997)(-0{,}04) = 0{,}4234$$

$$z_8 = f(z_{in_8}) = \frac{1 - e^{-(0{,}4234)}}{1 + e^{-(0{,}4234)}} = 0{,}2086$$

9. Proses keluaran neuron 9 pada *hidden layer* dengan neuron tambahan ke lapisan *output*

$$z_{in_9} = z_{in(1)_9} + u_1 w_{19(u)} + u_2 w_{29(u)} + u_3 w_{39(u)} + u_4 w_{49(u)} + u_5 w_{59(u)} + u_6 w_{69(u)} + u_7 w_{79(u)} + u_8 w_{89(u)} + u_9 w_{99(u)}$$

$$= -8{,}7098 + (0{,}8409)(-0{,}22883) + (0{,}5966)(-0{,}1093) + (0{,}3407)(0{,}2923) + (0{,}3637)(0{,}3193) + (-0{,}3561)(-0{,}3415) + (-0{,}5131)(0{,}2014) + (0{,}8648)(-0{,}4035) + (0{,}3452)(0{,}6083) + (-0{,}9997)(-0{,}6789) = -8{,}1933$$

$$z_9 = f(z_{in_9}) = \frac{1 - e^{-(-8{,}1933)}}{1 + e^{-(-8{,}1933)}} = -0{,}999$$

Untuk mendapatkan proses dari lapisan *output* menggunakan persamaan 2.18. *Output* yang didapatkan, digunakan sebagai input untuk proses perhitungan selanjutnya

## RIWAYAT HIDUP

### A. Identitas Diri

| | |
|---|---|
| Nama Lengkap | : Izzatul Yazidah |
| Tempat, Tanggal Lahir | : Brebes, 22 Agustus 2002 |
| Alamat | : Dk. Watujaya RT 008/RW 001, Kaliwadas, Bumiayu, Brebes |
| No. HP | : 083863865396 |
| Email | : izzatuly998@gmail.com |

### B. Riwayat Pendidikan

1. SD Negeri Kaliwadas 01 (Lulus 2014)
2. SMP Negeri 1 Bumiayu (Lulus 2017)
3. MA Negeri 2 Brebes (Lulus 2020)

Semarang, 27 Juni 2024

Izzatul Yazidah
NIM: 2008046001