

**APLIKASI METODE *HYBRID SINGULAR SPECTRUM*  
*ANALYSIS* DAN *NEURAL NETWORK* PADA  
PERAMALAN CURAH HUJAN KABUPATEN  
PANGKAJENE DAN KEPULAUAN**

**SKRIPSI**



**ALFIYAH SALSA DILA SABIR**

**H051191040**

**DEPARTEMEN STATISTIKA PROGRAM STUDI STATISTIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**DESEMBER 2023**

**LEMBAR PERNYATAAN KEOTENTIKAN**

Saya yang bertanda tangan di bawah ini menyatakan dengan sungguh-sungguh bahwa skripsi ini saya buat dengan judul:

***Aplikasi Metode Singular Spectrum Analysis dan Neural Network pada Peramalan Curah Hujan Kabupaten Pangkajene dan Kepulauan***

adalah benar hasil karya saya sendiri, bukan hasil plagiat dan belum pernah dipublikasikan dalam bentuk apapun

**Makassar, 6 Desember 2023**



**Alfiyah Salsa Dila Sabir**

**NIM H051191040**

**APLIKASI METODE *HYBRID SINGULAR SPECTRUM*  
*ANALYSIS* DAN *NEURAL NETWORK* PADA  
PERAMALAN CURAH HUJAN KABUPATEN  
PANGKAJENE DAN KEPULAUAN**

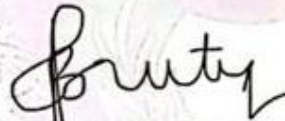
Disetujui Oleh:

Pembimbing Utama

Pembimbing Pertama



Sitti Sahriman, S.Si., M.Si.  
NIP. 19881018 201504 2 002



Sri Astuti Thamrin, S.Si., M.Stat., Ph.D  
NIP. 19740713 199903 2 001

Ketua Program Studi



Dr. Anna Islamiyati, S.Si., M.Si.  
NIP. 19770808 200501 2 002

Pada 6 Desember 2023

**HALAMAN PENGESAHAN**

Skripsi ini diajukan oleh:

Nama : Alfiyah Salsa Dila Sabir  
NIM : H051191040  
Program Studi : Statistika  
Judul Skripsi : Aplikasi Metode *Hybrid Singular Spectrum Analysis* dan *Neural Network* pada Peramalan Curah Hujan Kabupaten Pangkajene dan Kepulauan

Telah berhasil dipertahankan dihadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Sains pada Program Studi Statistika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

**DEWAN PENGUJI**

1. Ketua : Sitti Sahrinan, S.Si., M.Si. (.....)
2. Sekretaris : Sri Astuti Thamrin, S.Si., M.Stat., Ph.D (.....)
3. Anggota : Drs. Raupong, M.Si. (.....)
4. Anggota : Dr. Erna Tri Herdiani, S.Si., M.Si. (.....)

Ditetapkan di : Makassar

Tanggal : 6 Desember 2023

## KATA PENGANTAR

### *Assalamu'alaikum Warahmatullahi Wabarakatuh*

Puji syukur kehadiran Allah *Subhanahu Wa Ta'ala* atas segala limpahan rahmat, nikmat, dan hidayah-Nya yang telah diberikan sampai titik ini sehingga penulis mampu menyelesaikan penyusunan tugas akhir ini. Shalawat serta salam senantiasa tercurahkan kepada baginda Rasulullah *Shallallahu 'Alaihi Wa Sallam* beserta para keluarga, sahabat, dan pengikut beliau hingga akhir zaman kelak. *Alhamdulillah robbil'amin*, berkat nikmat kemudahan dan rahmat yang diberikan oleh Allah *Subhanahu Wa Ta'ala*, penulis dapat menyelesaikan tugas akhir yang berjudul “**Aplikasi Metode *Hybrid Singular Spectrum Analysis* dan *Neural Network* pada Peramalan Curah Hujan Kabupaten Pangkajene dan Kepulauan**” yang disusun sebagai salah satu syarat akademik memperoleh gelar sarjana pada Program Studi Statistika Departemen Statistika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

Penulis menyadari bahwasannya dengan segala keterbatasan pengetahuan dan kemampuan yang dimiliki penyelesaian tugas akhir ini tidak terlepas dari dukungan dan dorongan dari berbagai pihak yang senantiasa turut memberikan bantuan baik secara moril maupun materil sehingga penulis dapat menyelesaikan tugas akhir ini. Oleh karena itu, penulis senantiasa mengucapkan terima kasih yang setulus-tulusnya serta penghargaan yang setinggi-tingginya untuk orang tua penulis, Ayahanda **Muhammad Sabir** dan Ibunda **Jahrawaty** yang telah membesarkan dan mendidik penulis, memberikan dukungan penuh, pengorbanan, kesabaran hati, limpahan cinta dan kasih sayang tanpa batas, pemenuhan materi dengan sangat cukup serta dengan ikhlas telah mengiringi setiap langkah penulis dengan doa dan restu mulianya. Ucapan terima kasih juga penulis haturkan kepada adik-adik tersayang penulis **Abrar Gailan Mubarak** dan **Arkan Fayyadh Mufid** karena telah menjadi adik-adik yang sangat baik dan senantiasa memberikan semangat maupun doa terbaiknya untuk penulis dalam menyelesaikan tugas akhir ini, serta keluarga besar penulis, terima kasih atas dukungan dan doa mulianya selama ini.

Penghargaan yang tulus dan ucapan terima kasih dengan penuh keikhlasan juga penulis ucapkan kepada:

1. **Bapak Prof. Dr. Ir. Jamaluddin Jompa, M.Sc.**, selaku Rektor Universitas Hasanuddin beserta seluruh jajarannya.
2. **Bapak Dr. Eng. Amiruddin**, selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin beserta seluruh jajarannya.
3. **Ibu Dr. Anna Islamiyati, S.Si., M.Si.**, selaku Ketua Departemen Statistika, segenap Dosen Pengajar dan Staf yang telah membekali ilmu dan kemudahan kepada penulis dalam berbagai hal selama menempuh Pendidikan sarjana di Departemen Statistika.
4. **Ibu Sitti Sahriman, S.Si., M.Si.**, selaku Pembimbing Utama yang dengan penuh kesabaran telah meluangkan waktu dan pemikirannya di tengah berbagai kesibukan dan prioritasnya untuk senantiasa memberikan arahan, dorongan semangat, dan motivasi kepada penulis mulai dari awal hingga selesainya penulisan skripsi ini.
5. **Ibu Sri Astuti Thamrin, S.Si., M.Stat., Ph.D**, selaku Pembimbing Pertama sekaligus Penasehat Akademik penulis yang senantiasa meluangkan waktu memberikan bantuan, nasehat, serta motivasi kepada penulis selama menjadi mahasiswa di Departemen Statistika.
6. **Bapak Drs Raupong, M.Si.** dan **Ibu Dr. Erna Tri Herdiani, S.Si., M.Si.**, selaku Tim Penguji yang telah meluangkan waktunya kepada penulis dengan senantiasa memberikan saran dan kritikan yang membangun kepada penulis dalam penyempurnaan tugas akhir ini.
7. Sahabat karib penulis sejak di bangku SD, **Finka Ayunita, A.Md.T.** yang senantiasa hadir sebagai penghibur di kala jenuh dengan skripsi dan menjadi tempat ternyaman untuk berkeluh kesah perihal apapun. Terima kasih telah tumbuh bersama dan masih menjadi teman terbaik dari kecil hingga dewasa ini.
8. Sahabat penulis di bangku SMP dan SMA, **Affah Aulia Ananda Arif, A.Md.T., A. Tenri Ola Batari Nuneng, Nur Khaliza, S.S**, serta keluarga besar **SMAN 13 Pangkep**. Terima kasih atas kebahagiaan, kebersamaan, kerukunan yang sampai saat ini masih penulis rasakan.
9. Keponakan *online*, **Rayyanza Malik Ahmad** yang selalu menebar kebahagiaan secara *online* dengan tingkah lucunya dan menjadi *moodboster* penulis selama penyusunan skripsi.

10. Seluruh teman-teman penulis di **Statistika 2019** yang tidak dapat disebutkan namanya satu per satu. Terima kasih untuk seluruh warna dalam pertemanan yang diberikan selama kurang lebih empat tahun lamanya.
11. Seluruh teman-teman **KKN PPM Pangkep Gelombang 108 Kecamatan Anrong Appaka**, yang bersedia menjadi keluarga selama kurang lebih dua bulan terkhusus kepada **Andi Annisa Hermansyah, Kintan Amara Dhiemma, dan Aurora Istijoany**, terima kasih karena menjadi orang yang paling meluangkan waktunya dan menjadi peluk paling hangat penulis selama di lokasi KKN.
12. Teman-teman **UKM Liga Film Mahasiswa Universitas Hasanuddin**, terima kasih untuk pengalaman tak ternilai yang diberikan kepada penulis selama di bangku perkuliahan pada beberapa projek produksi film.
13. Semua pihak yang terlibat dalam proses perkuliahan penulis yang tidak dapat penulis sebutkan satu per satu. Pastinya tak henti-henti penulis sampaikan semoga amal baik semua pihak mendapat balasan yang berlipat ganda dari sang pencipta yang pengasih dan penyayang Allah SWT. Amin.

Sebagai manusia biasa penulis menyadari penyusunan skripsi ini jauh dari kata sempurna karena keterbatasan kemampuan dan ilmu pengetahuan yang dimiliki oleh penulis. Oleh karenanya atas kesalahan dan kekurangan dalam penulisan skripsi ini, penulis memohon maaf dan bersedia menerima kritikan yang membangun.

Terakhir, harapan penulis, semoga skripsi ini dapat memberikan manfaat bagi siapa saja yang membacanya.

**Makassar, 6 Desember 2023**



**Alfiyah Salsa Dila Sabir**

**PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK  
KEPENTINGAN AKADEMIK**

---

Sebagai civitas akademik Universitas Hasanuddin, saya yang bertanda tangan di bawah ini:

Nama : Alfiyah Salsa Dila Sabir  
NIM : H051191040  
Program Studi : Statistika  
Departemen : Statistika  
Fakultas : Matematika dan Ilmu Pengetahuan Alam  
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Bebas Royalti Non-eksklusif (*Non-exclusive Royalty-Free Right*)** atas tugas akhir saya yang berjudul:

**“Aplikasi Metode *Singular Spectrum Analysis* dan *Neural Network* pada Peramalan Curah Hujan Kabupaten Pangkajene dan Kepulauan”**

Beserta perangkat yang ada (jika diperlukan). Terkait dengan hal di atas, maka pihak universitas berhak menyimpan, mengalih-media/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar tanggal 6 Desember 2023.

Yang menyatakan,



(Alfiyah Salsa Dila Sabir)



**ABSTRAK**

Kabupaten Pangkajene dan Kepulauan merupakan kabupaten di Sulawesi Selatan dengan curah hujan bulanan mencapai 200 mm hingga 560 mm/bulannya. Dampak dari intensitas curah hujan yang tinggi dapat menyebabkan berbagai efek negatif terhadap lingkungan seperti bencana banjir dan longsor. Akibat yang ditimbulkan dari bencana alam yang terlambat diantisipasi mengakibatkan biaya, tenaga, dan waktu yang dibutuhkan lebih besar. Penelitian ini bertujuan memperoleh hasil peramalan curah hujan yang nantinya dapat digunakan sebagai dasar-dasar perencanaan, pengawasan, dan pengambilan keputusan serta memberi informasi terkait besarnya intensitas curah hujan mendatang menggunakan metode *hybrid SSA-NN*. Metode SSA adalah metode yang mampu mendekomposisi suatu deret menjadi beberapa komponen, seperti komponen *trend*, musiman, dan *noise*, kemudian hasil dekomposisi pola-pola yang terbentuk oleh SSA selanjutnya digunakan untuk peramalan menggunakan metode NN. Hasil penelitian menunjukkan bahwa metode terbaik adalah gabungan metode SSA dengan *window length 23* dan metode NN dengan arsitektur model 6-4-1 (6 *neuron input* – 4 *neuron hidden* – 1 *neuron output*) untuk komponen *trend*, 6-3-1 (6 *neuron input* – 3 *neuron hidden* – 1 *neuron output*) untuk komponen musiman, serta 6-10-1 (6 *neuron input* – 10 *neuron hidden* – 1 *neuron output*) untuk komponen *noise*. Tingkat kesalahan prediksi berdasarkan nilai MAPE mencapai 7.24%. Dan dengan tingkat korelasi sebesar 0.946.

**Kata Kunci:** *Hybrid SSA-NN*, Peramalan, Curah Hujan

**ABSTRACT**

*Pangkajene and Islands Regency is a district in South Sulawesi with monthly rainfall reaching 200 mm to 560 mm/month. The impact of high rainfall intensity can cause various negative effects on the environment such as floods and landslides. The consequences of natural disasters that are anticipated too late result in greater costs, energy and time required. This research aims to obtain rainfall forecasting results which can later be used as a basis for planning, monitoring and decision making as well as providing information regarding the magnitude of future rainfall intensity using the hybrid SSA-NN method. The SSA method is a method that can decompose a series into several components, such as trend, seasonal and noise components, then the results of the decomposition of the patterns formed by SSA are then used for forecasting using the NN method. The research results show that the best method is a combination of the SSA method with a window length of 23 and the NN method with a 6-4-1 model architecture (6 input neurons – 4 hidden neurons – 1 output neuron) for the trend component, 6-3-1 (6 input neurons – 3 hidden neurons – 1 output neuron) for the seasonal component, and 6-10-1 (6 input neurons – 10 hidden neurons – 1 output neuron) for the noise component. The prediction error rate based on the MAPE value reached 7.24%. And with a correlation level of 0.946.*

**Keywords:** *Hybrid SSA-NN, Forecasting, Rainfall*

## DAFTAR ISI

<b>HALAMAN SAMPUL</b> .....	<b>i</b>
<b>HALAMAN SAMPUL</b> .....	Error! Bookmark not defined.
<b>LEMBAR PERNYATAAN KEOTENTIKAN</b> .....	<b>ii</b>
<b>HALAMAN PENGESAHAN</b> .....	Error! Bookmark not defined.
<b>KATA PENGANTAR</b> .....	<b>v</b>
<b>PERNYATAAN PERSETUJUAN</b> .....	Error! Bookmark not defined.
<b>ABSTRAK</b> .....	<b>viii</b>
<b>ABSTRACT</b> .....	<b>x</b>
<b>DAFTAR ISI</b> .....	<b>xi</b>
<b>DAFTAR GAMBAR</b> .....	<b>xiii</b>
<b>DAFTAR TABEL</b> .....	<b>xiv</b>
<b>DAFTAR LAMPIRAN</b> .....	<b>xv</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1    Latar Belakang .....	1
1.2    Rumusan Masalah .....	3
1.3    Tujuan Penelitian.....	3
1.4    Batasan Masalah.....	3
1.5    Manfaat Penelitian.....	4
<b>BAB II TINJAUAN PUSTAKA</b> .....	<b>5</b>
2.1 <i>Singular Spectrum Analysis</i> .....	5
2.1.1 <i>Embedding</i> .....	5
2.1.2 <i>Singular Value Decomposition</i> .....	6
2.1.3 <i>Grouping</i> .....	6
2.1.4 <i>Diagonal Averaging</i> .....	7
2.2 <i>Neural Network</i> .....	8
2.2.1    Arsitektur Jaringan <i>Neural Network</i> .....	9
2.3    Algoritma <i>Backpropagation</i> .....	11
2.4    Fungsi Aktivasi.....	15
2.5    Pengukuran Akurasi .....	15
2.6    Curah Hujan .....	16
<b>BAB III METODOLOGI PENELITIAN</b> .....	<b>18</b>
3.1    Sumber Data .....	18
3.2    Struktur Data .....	18
3.3    Tahapan Analisis .....	18

<b>BAB IV HASIL DAN PEMBAHASAN .....</b>	<b>22</b>
4.1 Statistik Deskriptif Data .....	22
4.2 <i>Hybrid Singular Spectrum Analysis dan Neural Network</i> .....	23
4.2.1 <i>Singular Spectrum Analysis</i> .....	24
4.2.1.1 <i>Embedding</i> .....	24
4.2.1.2 <i>Singular Value Decomposition</i> .....	25
4.2.1.3 <i>Grouping</i> .....	27
4.2.1.4 <i>Diagonal Averaging</i> .....	29
4.2.2 <i>Neural Network</i> .....	30
4.2.2.1 <i>Feedforward</i> .....	33
4.2.2.2 <i>Backpropagation</i> .....	35
4.2.2.3 <i>Perubahan Bobot dan Bias</i> .....	37
4.3 <i>Prediksi Curah Hujan Kabupaten Pangkajene dan Kepulauan</i> .....	42
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>45</b>
5.1 Kesimpulan.....	45
5.2 Saran .....	45
<b>DAFTAR PUSTAKA .....</b>	<b>46</b>
<b>LAMPIRAN.....</b>	<b>51</b>

## DAFTAR GAMBAR

<b>Gambar 2.1</b> Arsitektur Neural Network .....	9
<b>Gambar 2.2</b> Jaringan Saraf Lapisan Tunggal .....	10
<b>Gambar 2.3</b> Jaringan Saraf Lapisan Banyak .....	10
<b>Gambar 2.4</b> Jaringan Saraf Lapisan Kompetitif .....	11
<b>Gambar 4.1</b> Plot Deret Waktu Curah Hujan Kabupaten Pangkajene dan Kepulauan Pos Hujan Bungoro .....	22
<b>Gambar 4.2</b> Grafik <i>Singular Value</i> .....	27
<b>Gambar 4.3</b> Plot <i>Eigenvector</i> .....	28
<b>Gambar 4.4</b> Plot Arsitektur <i>Neural Network</i> Komponen <i>Trend</i> .....	40
<b>Gambar 4.5</b> Plot Arsitektur <i>Neural Network</i> Komponen Musiman .....	41
<b>Gambar 4.6</b> Arsitektur <i>Neural Network</i> Komponen <i>Noise</i> .....	42
<b>Gambar 4.7</b> Perbandingan Data Aktual dengan Hasil Prediksi SSA-NN .....	44

## DAFTAR TABEL

<b>Tabel 3.1</b> Struktur Data Penelitian .....	18
<b>Tabel 4.1</b> Analisis Deskriptif Curah Hujan Kabupaten Pangkajene dan Kepulauan Pos Hujan Bungoro.....	23
<b>Tabel 4.2</b> Hasil MAPE berdasarkan nilai $L$ .....	24
<b>Tabel 4.3</b> Hasil Tracking MAPE berdasarkan nilai $L$ .....	24
<b>Tabel 4.4</b> Eigenvalues dan Singular Values .....	26
<b>Tabel 4.5</b> Eigenvector.....	26
<b>Tabel 4.6</b> Principal Component.....	27
<b>Tabel 4.7</b> Eigentriple dan Komponen yang Terkait .....	29
<b>Tabel 4.8</b> Hasil Rekonstruksi .....	29
<b>Tabel 4.9</b> Variabel Input dan Target.....	30
<b>Tabel 4.10</b> Hasil MAPE berdasarkan Neuron .....	31
<b>Tabel 4.11</b> Bobot Awal dan Bias Input Layer ke Hidden Layer pada Komponen Trend.....	32
<b>Tabel 4.12</b> Bobot Awal dan Bias Hidden Layer ke Output Layer pada Komponen Trend.....	32
<b>Tabel 4.13</b> Hasil Normalisasi .....	33
<b>Tabel 4.14</b> Nilai Kesalahan Bobot .....	37
<b>Tabel 4.15</b> Bobot Akhir dan Bias Input Layer ke Hidden Layer (1 kali iterasi)..	39
<b>Tabel 4.16</b> Bobot Akhir dan Bias Hidden Layer ke Output Layer (1 kali iterasi)	39
<b>Tabel 4.17</b> Hasil Prediksi dan Akurasi Hasil Prediksi .....	43
<b>Tabel 4.18</b> Perbandingan MAPE dan Korelasi.....	43

**DAFTAR LAMPIRAN**

**Lampiran 1.** Data Curah Hujan Bulanan Kabupaten Pangkajene dan Kepulauan pada Pos Hujan Bungoro Periode 2018-2022..... 52

**Lampiran 2.** *Output* Hasil Rekonstruksi menggunakan *software R Studio* ..... 53

**Lampiran 3.** *Output* Hasil Rekonstruksi menggunakan *software R Studio*..... 54

**Lampiran 4.** *Output* Bobot dan Bias Awal dengan Nilai *Random* ..... 55

**Lampiran 5.** Data Hasil Normalisasi untuk *Input Neural Network*..... 56

**Lampiran 6.** Hasil Prediksi Metode *Hybrid SSA-NN* pada Data Uji..... 58

**Lampiran 7.** *Output* Perbandingan menggunakan MAPE dan Korelasi *Pearson* 59

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Indonesia merupakan negara beriklim tropis yang dilalui oleh garis khatulistiwa serta dikelilingi oleh dua benua dan dua samudra yang menjadikan Indonesia rentan terhadap perubahan iklim atau cuaca (Suryani dkk., 2018). Cuaca dan iklim memiliki hubungan yang saling mempengaruhi terhadap unsur-unsur yang lain seperti kelembapan udara, temperatur udara, dan curah hujan (Desvina dan Ratnawati, 2014). Curah hujan merupakan salah satu unsur yang memiliki pengaruh pada aktivitas kehidupan (Syaifullah, 2014). Badan meteorologi, klimatologi, dan geofisika (BMKG) mengelompokkan curah hujan bulanan menjadi empat kategori yaitu kategori rendah (0-100 mm), kategori sedang (100-300 mm), kategori tinggi (300-500 mm) dan kategori sangat tinggi (>500 mm) (Supriyati dkk., 2018).

Kabupaten Pangkajene dan Kepulauan merupakan salah satu kabupaten di Sulawesi Selatan dengan curah hujan bulanan 200 mm/bulannya namun dengan bulan terbasah mencapai lebih dari 560 mm/bulan (PangkepKab). Dampak dari curah hujan yang tergolong tinggi dapat menyebabkan bencana, seperti bencana banjir, longsor, dan efek negatif terhadap lingkungan. Bersumber dari detikSulsel (2023) banjir berhasil merendam 7 kecamatan di Kabupaten Pangkajene dan Kepulauan salah satunya adalah Kecamatan Bungoro, banjir terjadi akibat intensitas hujan yang cukup tinggi sejak pagi hingga tengah malam. Akibat yang ditimbulkan dari bencana alam yang terlambat diantisipasi akan mengakibatkan biaya, tenaga, dan waktu yang dibutuhkan lebih besar sehingga perlu dilakukan peramalan untuk memperkirakan besarnya curah hujan mendatang.

Analisis runtun waktu merupakan salah satu prosedur statistik yang umum digunakan untuk memodelkan kondisi curah hujan (Fauziah dkk., 2016). Peramalan data runtun waktu dapat dianalisis menggunakan metode yang beragam. Pemilihan metode bergantung pada berbagai aspek yang mempengaruhi seperti aspek waktu, pola data, tipe model sistem yang diamati, hingga tingkat keakuratan peramalan. Selain itu dalam menerapkan suatu metode, data harus memenuhi asumsi seperti stasioneritas dan normalitas (Christienova dkk., 2018). Saat ini telah berkembang metode non-linier yang dianggap mewakili kondisi data, karena terkadang



ditemukan data yang tidak memenuhi asumsi-asumsi pada model linier (Eksiandayani, 2016). Data curah hujan umumnya dapat dimodelkan secara linier namun pada saat sekarang karakteristik dan pola curah hujan terkadang sulit untuk diidentifikasi sehingga model linier tidak mampu menjelaskan karakteristik non-linier dari data dengan baik (Sinay dkk., 2020).

Beberapa pendekatan *non-linier* yang dapat digunakan dalam peramalan diantaranya, *support vector machine* (SVM), *long short-term memory* (LSTM), *k-nearest neighbors* (KNN) dan *singular spectrum analysis* (SSA). SSA merupakan salah satu pendekatan yang mempunyai tujuan utama yaitu mendekomposisikan deret waktu asli menjadi beberapa komponen yang dipisahkan seperti komponen *trend*, musiman, dan *noise* (Golyandina dkk., 2001). Berdasarkan tujuan tersebut, beberapa data deret waktu memerlukan dekomposisi agar pola diketahui secara terpisah. Golyandina dkk (2001) menjelaskan bahwa kurang optimalnya hasil dekomposisi dapat menjadi salah satu kekurangan SSA. Namun dalam prosesnya SSA dapat mengurangi *noise* atau unsur yang tidak relevan pada data, yang dapat memberikan nilai eror lebih kecil dan hasil yang stabil dalam peramalan.

Metode non-linier yang juga banyak digunakan dalam peramalan adalah *neural network* (NN) karena memiliki kemampuan untuk menangani data kompleks (Syamsiah dan Purwandani, 2019). NN dengan proses pembelajaran jaringan saraf, dianggap memiliki proses fungsi pendekatan khusus dan solusi terhadap masalah yang diberikan umumnya kurang akurat, sehingga kemampuan fungsi pendekatan dan ketepatan solusi perlu dipertimbangkan (Suryani dan Wahono, 2015). Namun metode NN memiliki kemampuan untuk mempelajari pola data yang diberikan pada tahap pelatihan dan juga memiliki toleransi kesalahan yang tinggi serta mampu menjalankan perhitungan secara paralel yang membuat proses menjadi singkat (Diwanda dkk., 2021). Oleh karena itu, untuk mendapatkan hasil ramalan yang lebih optimal dan akurat dilakukan *hybrid* antara metode *singular spectrum analysis* dan *neural network* yang menggabungkan kelebihan dari kedua metode. Secara umum juga telah disetujui dalam literatur peramalan bahwa belum terdapat satu metode yang mampu untuk di semua situasi (Eksiandayani, 2016).

Beberapa penelitian terdahulu dalam meramalkan curah hujan diantaranya Purnama (2022) mempelajari model peramalan curah hujan di Provinsi Gorontalo

dan melakukan peramalan untuk tahun 2022-2023 menggunakan metode *singular spectrum analysis*. Penelitian terkait lainnya dilakukan oleh Fitriyanti (2022) yang menggunakan *neural network* sebagai sistem prediksi dalam prakiraan curah hujan bulanan di Kabupaten Wajo, Sulawesi Selatan. Sementara Setyowati dkk (2019) telah melakukan penelitian menggunakan metode *hybrid singular spectrum analysis* dan *neural network* (SSA-NN) terkait peramalan *inflow* dan *outflow* mata uang Bank Indonesia.

Berdasarkan uraian tersebut, maka penelitian ini difokuskan dalam melakukan peramalan terkait data curah hujan di Kabupaten Pangkajene dan Kepulauan menggunakan metode *hybrid singular spectrum analysis* dan *neural network*. Pada penelitian ini menggunakan algoritma pembelajaran *neural network* yaitu *backpropagation* dengan fungsi aktivasi *sigmoid biner*. Hasil dari penelitian ini diharapkan mampu membantu sebagai dasar-dasar perencanaan, pengawasan, dan pengambilan keputusan serta memberi informasi terkait besarnya intensitas curah hujan bulanan kedepannya.

## **1.2 Rumusan Masalah**

Berdasarkan uraian pada latar belakang, maka rumusan masalah pada penelitian ini adalah bagaimana hasil peramalan pada data curah hujan bulanan Kabupaten Pangkajene dan Kepulauan menggunakan metode *hybrid singular spectrum analysis* dan *neural network*?

## **1.3 Tujuan Penelitian**

Berdasarkan perumusan masalah, penelitian ini bertujuan untuk memperoleh hasil peramalan pada data curah hujan bulanan Kabupaten Pangkajene dan Kepulauan menggunakan metode *hybrid singular spectrum analysis* dan *neural network*.

## **1.4 Batasan Masalah**

Batasan masalah pada penelitian ini adalah sebagai berikut:

1. Data yang digunakan adalah data curah hujan bulanan periode 2018-2022.
2. Fungsi aktivasi dalam algoritma pembelajaran yang digunakan adalah *sigmoid biner*.

3. Pengujian kinerja peramalan menggunakan *mean absolute percentage error* (MAPE) dan korelasi.

### **1.5 Manfaat Penelitian**

Manfaat yang diharapkan dari hasil penelitian ini adalah sebagai berikut:

1. Menambah wawasan dan pengetahuan mengenai peramalan dengan metode *hybrid singular spectrum analysis* dan *neural network*.
2. Memberikan inspirasi bagi pembaca dalam pengembangan metode dalam analisis runtun waktu.
3. Memberikan informasi kedepannya untuk kepentingan-kepentingan pihak yang terkait dengan keadaan hujan atau curah hujan di Kabupaten Pangkajene dan Kepulauan.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 *Singular Spectrum Analysis*

*Singular spectrum analysis* (SSA) adalah teknik analisis deret waktu yang mendekomposisi deret waktu asli menjadi suatu penjumlahan dari sejumlah kecil komponen yang independen dan dapat diinterpretasikan seperti komponen *trend*, musiman, dan *noise*. Analisis deret waktu menjadi salah satu prosedur dalam statistika yang diterapkan untuk meramalkan peluang keadaan mendatang dalam rangka pengambilan keputusan. Dasar pemikiran deret waktu adalah pengamatan sekarang ( $Z_t$ ) dipengaruhi oleh satu atau beberapa pengamatan sebelumnya ( $Z_{t-n}$ );  $n = 1, 2, 3 \dots$ . Berdasarkan dengan hal itu, maka model deret waktu dibuat karena secara statistik ada korelasi antar deret pengamatan (Adiputra dkk., 2021). SSA merupakan gabungan dari dua istilah *singular* dan *spectrum*. Istilah *singular* didefinisikan dekomposisi spektral atau dekomposisi eigen dari matriks lintasan dan istilah *spectrum* didefinisikan untuk penjumlahan satu set nilai eigen setelah dekomposisi spektral (Sakinah, 2018). SSA merupakan model yang terhindar dari banyaknya syarat seperti independensi dan normalitas residual. SSA mampu memodelkan data musiman dengan multi periode dan musiman kompleks (Chrisantama dkk., 2021).

Algoritma dasar SSA adalah membagi deret waktu ke dalam data deret waktu yang baru. Pada dasarnya SSA terdiri dari dua tahap, yaitu dekomposisi dan rekonstruksi. Tahap dekomposisi terdiri dari tahapan *embedding* dan *singular value decomposition* sedangkan tahap rekonstruksi terdiri dari tahapan *grouping* dan *diagonal averaging* (Khaeri dkk., 2018). Metode SSA terdiri dari 4 langkah yang terdiri dari 2 tahap yang saling melengkapi (Golyandina dkk., 2001), yaitu:

##### 2.1.1 *Embedding*

Pada *embedding* mengubah data runtun waktu ke dalam bentuk matriks lintasan (*trajectory matrix*) dengan mentransformasikan data yang berdimensi satu (vektor) menjadi data multidimensional (matriks) (Siringoringo dkk., 2022). Data deret waktu awal diubah menjadi matriks *trajectory* yang memiliki dimensi  $L \times K$ , diketahui bahwa  $L$  merupakan *window length* yang menjadi baris matriks, sedangkan  $K = N - L + 1$  yang menjadi kolom matriks (Satriani dkk., 2020).

Diasumsikan data deret waktu dengan panjang  $N$ , tanpa data hilang (*missing data*) dinyatakan dengan  $X = \{x_1, x_2, \dots, x_n\}$ , dan dengan rentang pemilihan nilai  $L$  adalah  $2 < L < \frac{N}{2}$ . Belum ada metode khusus untuk menentukan nilai  $L$  secara pasti, sehingga untuk menentukan nilai  $L$  dilakukan dengan proses coba-coba (*trial and error*) (Idrus dkk., 2022). Matriks *trajectory* dapat dituliskan sebagai berikut:

$$X = \begin{bmatrix} x_1 & x_2 & \cdots & x_K \\ x_2 & x_3 & \cdots & x_{K+1} \\ \vdots & \cdots & \ddots & \vdots \\ x_L & x_{L+1} & \cdots & x_N \end{bmatrix} \quad (2.1)$$

### 2.1.2 Singular Value Decomposition

*Singular value decomposition* (SVD) atau dekomposisi nilai singular bertujuan untuk memperoleh pemisahan komponen dalam dekomposisi dari deret waktu untuk mempermudah pengolahan data (Satriani dkk., 2020). Misalkan matriks  $S = XX^T$ , dengan menggunakan matriks  $S$  kemudian dihitung *eigenvalue*  $(\lambda_1, \lambda_2, \dots, \lambda_L)$  dengan urutan secara menurun  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_L \geq 0$ . Selain itu  $U_1, U_2, \dots, U_L$  adalah *eigenvector* dari masing-masing *eigenvalue* (Sitohang dan Darmawan, 2018). Rank dari matriks  $X$  dapat ditunjukkan dengan  $d = \max\{i, \lambda_i \geq 0\}$ . Jika dinotasikan  $V_i = \frac{X^T U_i}{\sqrt{\lambda_i}}$  untuk  $i = 1, 2, \dots, d$ , maka SVD dari matriks lintasan adalah sebagai berikut (Idrus dkk., 2022):

$$X = X_1 + X_2 + \cdots + X_d \quad (2.2)$$

dengan  $X_i = \sqrt{\lambda_i} U_i V_i^T$  memiliki rank 1 (Isnawati, 2018).

Matriks  $X$  terbentuk dari *eigenvector*  $U_i$ , *singular value*  $\sqrt{\lambda_i}$  dan *principal component*  $V_i$ . Ketiga elemen pembentuk SVD ini disebut dengan *eigen triple*. Konsep dasar SVD adalah mendapatkan barisan matriks dari matriks  $S$  yang pada masing-masing matriks dalam barisan tersebut mengandung *eigenvector*  $U_i$ , *singular value*  $\sqrt{\lambda_i}$  dan *principal component*  $V_i$  yang menggambarkan karakteristik pada masing-masing matriks dalam barisan (Christienova dkk., 2018).

### 2.1.3 Grouping

*Grouping* adalah tahap penguraian matriks lintasan  $L \times K$  menjadi beberapa komponen yaitu *trend*, musiman, dan *noise* yang terdapat pada data deret waktu

(Ningsih, 2021). Parameter yang digunakan pada tahap ini adalah *grouping effect* ( $r$ ). *Grouping effect* digunakan untuk membatasi jumlah *eigentriple* yang akan mengidentifikasi komponen *trend* dan musiman. Pengelompokan berhubungan erat dengan pemecahan matriks  $X_i$  menjadi beberapa kelompok dan menjumlahkan matriks dalam masing-masing kelompok (Christienova dkk., 2018). Proses *grouping* dilakukan dengan cara mengelompokkan set-set indeks  $i = \{1, 2, \dots, d\}$  menjadi  $m$  subset (Wicaksono dkk., 2019). Matriks  $X_i$  dipartisi ke  $m$  *disjoint subset*  $I_1, I_2, \dots, I_m$ . Jika  $I = \{i_1, i_2, \dots, i_p\}$  maka yang dihasilkan dari matriks  $X_i$  sesuai dengan kelompok  $I$  yang didefinisikan sebagai  $X_I = X_{i_1} + X_{i_2} + \dots + X_{i_p}$ . Matriks yang dihasilkan dihitung untuk pengelompokan  $I = I_1, I_2, \dots, I_m$  maka persamaan yang terbentuk sebagai berikut:

$$X_{I_x} = X_{I_1} + X_{I_2} + \dots + X_{I_m} \quad (2.3)$$

Prosedur pemilihan set  $I_1, I_2, \dots, I_m$  disebut *eigentriple grouping* (Shafira dkk., 2020).

#### 2.1.4 Diagonal Averaging

Tahapan *diagonal averaging* adalah tahapan merekonstruksi masing-masing matriks yang terdapat pada matriks  $X_{I_x}$  menjadi data deret waktu yang baru dengan panjang  $N$  (Satriani dkk., 2020). Langkah dasar untuk melakukan tahap ini adalah mengubah bentuk dari matriks  $X_{I_x}$  dari tahap rekonstruksi menjadi matriks Hankel. Cara optimal untuk melakukannya adalah dengan mencari rata-rata semua elemen pada beberapa antidiagonal (Utami dkk., 2019). Dimisalkan matriks  $Y$  sebagai berikut:

$$Y = \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1K} \\ y_{21} & y_{22} & \dots & y_{2K} \\ \vdots & \dots & \ddots & \vdots \\ y_{L1} & y_{L2} & \dots & y_{LK} \end{bmatrix} \quad (2.4)$$

Jika  $Y$  adalah matriks  $L \times K$  dengan elemen  $y_{ij}, 1 \leq i \leq L, 1 \leq j \leq K, L^* = \min(L, K), K^* = \max(L, K)$ , dan  $N = L + K - 1$ . Rata-rata *diagonal averaging* diperoleh dengan rumus sebagai berikut (Asrof dkk., 2017):

$$g_k = \begin{cases} \frac{1}{k+1} \sum_{m=1}^{k+1} y_{m,k-m+2}^*; 0 \leq k < L^* - 1 \\ \frac{1}{L^*} \sum_{m=1}^{L^*} y_{m,k-m+2}^*; L^* - 1 \leq k < K^* \\ \frac{1}{N-k} \sum_{m=k-K^*+2}^{N-K^*+1} y_{m,k-m+2}^*; K^* \leq k < N \end{cases} \quad (2.5)$$

Berdasarkan persamaan (2.4) dan (2.5), misalnya dipilih  $k = 1$  dihasilkan  $g_1 = y_{11}$ ,  $k = 2$  dihasilkan  $g_2 = \frac{y_{12}+y_{21}}{2}$ , dan seterusnya (Satriani dkk., 2020).

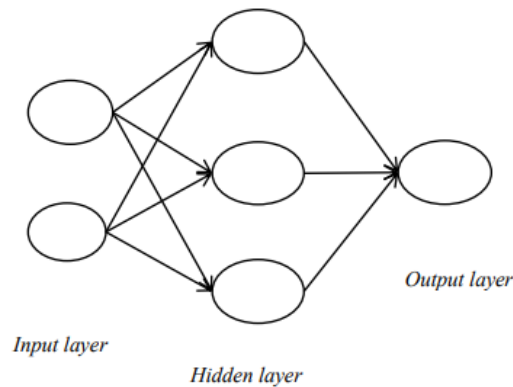
## 2.2 Neural Network

Jaringan saraf tiruan, dalam bahasa Inggris disebutkan *artificial neural network*, umumnya hanya disebut *neural network* (NN) merupakan sistem pemrosesan informasi yang memiliki karakteristik mirip dengan jaringan saraf biologi (Ma'rufah dkk., 2013). Kemampuan otak manusia seperti mengingat, menghitung, menggeneralisasi, dan adaptasi, diharapkan *neural network* dapat meniru kemampuan tersebut. NN berguna untuk memecahkan persoalan yang berkaitan dengan pengenalan pola, klasifikasi, prediksi dan data mining (Badrul, 2016). Jaringan saraf tiruan memiliki karakteristik sebagai berikut (Fausett, 1994):

1. Pola koneksi antara *neuron* yang disebut arsitektur.
2. Metode penentuan bobot pada koneksi (*training, learning, algoritma*).
3. Fungsi aktivasi yang dijalankan masing-masing *neuron* pada *input* jaringan untuk menentukan *output* (pembangkit sinyal).

*Neural network* terdiri atas elemen-elemen untuk pemrosesan informasi yang disebut dengan unit (*neuron, sel atau node*). Setiap *neuron* dihubungkan dengan *neuron* lainnya dengan suatu *connection link*, yang direpresentasikan dengan *weight/bobot*. Setiap *neuron* menggunakan fungsi aktivasi pada net *input* untuk menentukan prediksi *output*. *Neuron-neuron* dalam *neural network* disusun dalam grup, yang disebut *layer* (lapisan). Susunan *neuron-neuron* dalam lapis dan pola koneksi di dalam dan antar lapis disebut dengan arsitektur jaringan. Arsitektur jaringan ini merupakan salah satu karakteristik penting yang membedakan *neural network* (Kusumadewi, 2014). Secara umum lapisan pada arsitektur *neural network* terdapat tiga lapis yaitu *input layer, hidden layer, dan output layer*. Setiap lapisan

tersebut memiliki jumlah *neuron* yang berbeda-beda. Arsitektur *neural network* digambarkan seperti pada Gambar 2.1 sebagai berikut (Sholikhah, 2021):



**Gambar 2.1** Arsitektur *Neural Network*

Berdasarkan Gambar 2.1 lapisan arsitektur *neural network* terdiri dari:

- a. *Input layer* yaitu lapisan dari beberapa *neuron* yang menerima sinyal dari luar kemudian diteruskan ke *neuron* lain yang ada pada jaringan yang menggambarkan suatu permasalahan. Banyak *node* atau *neuron* dalam lapis *input* tergantung banyaknya *input* dalam model dan setiap *input* menentukan satu *neuron* (Sonata, 2021).
- b. *Hidden layer* merupakan *node-node* yang ada di dalam lapisan tersembunyi. *Output* dari lapisan ini tidak secara langsung dapat diamati (Achmalia, 2019). *Hidden layer* terletak di antara lapis *input* dan lapis *output*, yang terdiri dari atas beberapa lapis tersembunyi (Sonata, 2021).
- c. *Output layer* merupakan lapisan yang menghasilkan solusi *neural network* terhadap suatu permasalahan. Setelah melalui proses *training*, *network* merespon *input* baru untuk menghasilkan *output* yang merupakan hasil peramalan (Kusumadewi, 2014).

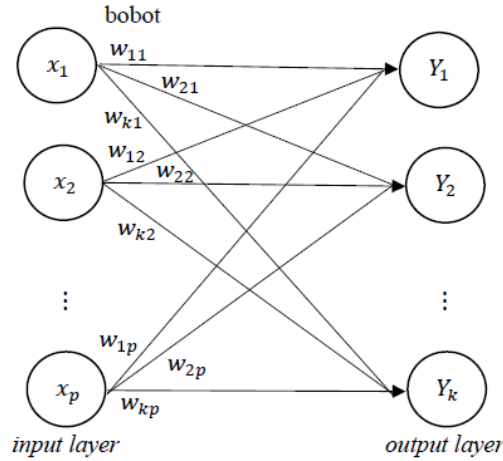
### 2.2.1 Arsitektur Jaringan *Neural Network*

Di dalam jaringan saraf tiruan, *neuron-neuron* dikelompokkan dalam lapisan-lapisan (*layers*). Arsitektur jaringan NN dibagi menjadi 3 macam, yaitu sebagai berikut (Amalina, 2016):

1. Jaringan saraf dengan lapisan tunggal (*single layer net*), lapisan ini hanya terdapat satu lapisan dengan bobot yang terhubung. Dapat dikatakan ciri-ciri lapisan ini adalah memiliki satu lapisan *input* dan satu lapisan *output*. Lapisan

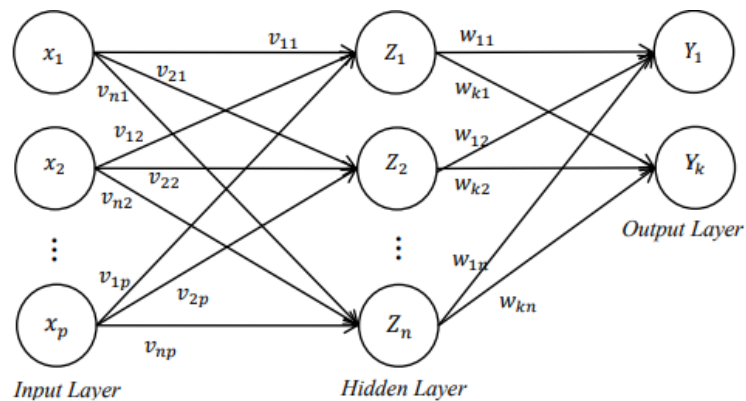


*input* sepenuhnya terkoneksi ke lapisan *output* tetapi tidak terkoneksi dengan unit yang berada pada lapisan yang sama. *Single layer* digambarkan pada Gambar 2.2 sebagai berikut (Sholikhah, 2021).



**Gambar 2.2** Jaringan Saraf Lapisan Tunggal

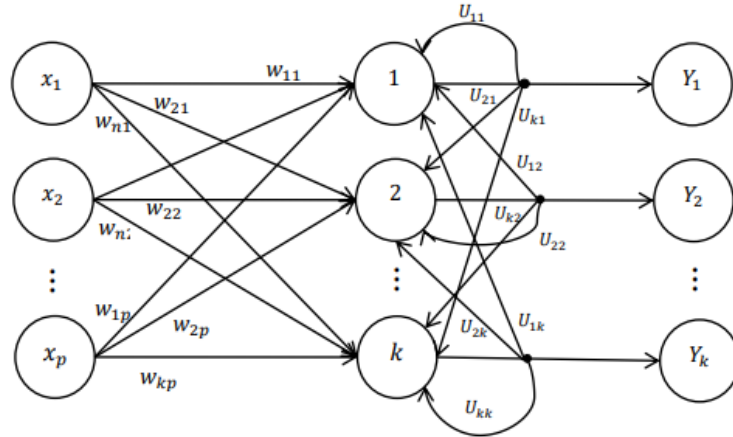
- Jaringan saraf dengan banyak lapisan (*multi-layer net*), jaringan ini memiliki banyak lapisan dan satu atau lebih lapisan yang terletak di antara lapisan *input* dan lapisan *output*. Jaringan ini menyelesaikan permasalahan yang lebih rumit dibandingkan dengan lapisan tunggal. Terdapat sebuah lapisan dari bobot diantara dua tingkat yang berdekatan dari unit (*input*, *hidden* atau *output*). *Multi-layer* digambarkan pada Gambar 2.3 sebagai berikut (Sholikhah, 2021).



**Gambar 2.3** Jaringan Saraf Lapisan Banyak

- Jaringan saraf dengan lapisan kompetitif (*competitive layer*), lapisan ini memiliki perbedaan dengan lapisan lain yang mana antar *neuron* saling dihubungkan. Jaringan ini disebut dengan *feedback loop* dikarenakan *output-*

nya ada yang memberikan informasi terhadap tiap *input*-nya. Jaringan ini memiliki satu perulangan (*loop*) umpan balik. *Competitive layer* digambarkan pada Gambar 2.4 sebagai berikut (Sholikhah, 2021).



**Gambar 2.4** Jaringan Saraf Lapisan Kompetitif

### 2.3 Algoritma *Backpropagation*

Algoritma *backpropagation* pertama kali diperkenalkan pada tahun 1970-an, tetapi mulai berkembang secara luas pada tahun 1986 oleh David Rumelhart, Geoffrey Hinton, dan Ronald Williams. Pada perkembangannya *backpropagation* bekerja jauh lebih cepat daripada pendekatan lainnya, sehingga memungkinkannya digunakan untuk memecahkan masalah yang sebelumnya tidak dapat dipecahkan (Nielsen, 2015). *Backpropagation* merupakan sebuah algoritma pembelajaran dalam *artificial neural network* yang telah banyak digunakan untuk memecahkan kasus-kasus rumit. Di dalam algoritma ini dilakukan dua tahap perhitungan, yaitu perhitungan maju yang dilakukan untuk menghitung error antara *output* NN dengan target yang diinginkan. Dan yang berikutnya adalah perhitungan mundur yang menggunakan error yang telah didapatkan untuk memperbaiki bobot pada semua *neuron* yang ada (Ryandhi, 2017).

Tujuan algoritma *backpropagation* yang digunakan untuk menemukan fungsi yang bergerak secara berulang sehingga didapatkan fungsi maksimal atau minimal. *Backpropagation* banyak digunakan di beberapa aplikasi pengendalian karena proses *training*-nya didasarkan dengan interkoneksi yang sederhana, yaitu apabila *output* memberikan hasil yang salah maka bobot dikoreksi agar error dapat diperkecil dan tanggapan NN selanjutnya diharapkan mendekati nilai yang benar

sehingga menjadikan *backpropagation* juga berkemampuan untuk memperbaiki bobot pada *hidden layer* (Abhishek dkk., 2012).

Fausett (1994) menyatakan bahwa algoritma *backpropagation* dalam melakukan *training* terhadap suatu jaringan terdapat 3 tahapan yaitu fase *feedforward*, fase *backpropagation* dan fase perubahan atau penyesuaian bobot. Secara rinci langkah dalam algoritma *backpropagation* diuraikan sebagai berikut (Sholikhah, 2021):

1. Fase *feedforward* yaitu fase ketika setiap *input* dipropagasi (dihitung maju) menuju *hidden layer* hingga *output layer* dengan menggunakan fungsi aktivasi yang ditentukan.
  - a. Setiap unit pada *input layer* ( $x_i, i = 1, 2, \dots, d$ ) menerima sinyal *input*  $x_i$  dan diteruskan ke *hidden layer*.
  - b. Jumlahkan tiap *hidden layer* ( $Z_j, j = 1, 2, \dots, n$ ) dengan bobot sinyal *input*.

$$Z_{net_j} = v_{0j} + \sum_{i=1}^d x_i v_{ij} \quad (2.6)$$

dengan:

$Z_{net_j}$  : nilai *output* untuk unit  $Z_j$

$x_i$  : nilai *input* ke- $i$

$v_{ij}$  : bobot antara *input layer* ke- $i$  dan *hidden layer* ke- $j$

$v_{0j}$  : bias menuju *hidden layer* ke- $j$

Mengaplikasikan fungsi aktivasi untuk menghitung sinyal *output* dan kemudian mengirim sinyal ke semua unit pada *output layer*.

$$z_j = f(Z_{net_j}) = \frac{1}{1 + e^{-Z_{net_j}}} \quad (2.7)$$

dengan:

$z_j$  : nilai aktivasi dari unit  $Z_j$

- c. Setiap unit pada *output layer* ( $Y_l, l = 1, 2, \dots, k$ ) menjumlahkan bobot sinyal *input*.

$$Y_{net_l} = w_{0k} + \sum_{j=1}^n z_j w_{jk} \quad (2.8)$$

dengan:

$Y_{net_l}$  : nilai *output* untuk unit  $Y_l$

$w_{0k}$  : bias menuju *output layer* ke- $k$

$w_{jk}$  : bobot antara *hidden layer* ke- $j$  dan *output layer* ke- $k$

Mengaplikasikan fungsi aktivasinya untuk menghitung sinyal *output*.

$$y_l = f(Y_{net_l}) = \frac{1}{1 + e^{-Y_{net_l}}} \quad (2.9)$$

dengan:

$y_l$  : nilai *output*

2. Fase *backpropagation* yaitu fase ketika kesalahan (selisih antara *output* dengan target yang diinginkan) yang terjadi dipropagasi mundur dari garis yang berhubungan langsung dengan unit-unit di *output layer*.

a. Setiap unit *output layer* ( $Y_l, l = 1, 2, \dots, k$ ) menerima pola target sesuai dengan pola *training input*, dengan menghitung informasi error.

$$\delta_l = (t_l - y_l)f'(Y_{net_l}) \quad (2.10)$$

dengan:

$\delta_l$  : faktor koreksi error bobot  $w_{jk}$

$t_l$  : nilai target

Menghitung bobot barunya (digunakan untuk memperbaharui  $w_{jk}$ ).

$$\Delta w_{jk} = \alpha \delta_l z_j \quad (2.11)$$

dengan:

$\Delta w_{jk}$  : nilai koreksi error bobot  $w_{jk}$

$\alpha$  : nilai *learning rate*

$z_j$  : nilai aktivasi dari unit  $Z_j$

Menghitung korelasi bias (digunakan untuk memperbaharui  $w_{0k}$ ).

$$\Delta w_{0k} = \alpha \delta_l \quad (2.12)$$

dengan:

$\Delta w_{0k}$  : nilai koreksi error bias

Mengirim ke unit *layer* sebelumnya.

b. Setiap *hidden* ( $Z_j, j = 1, 2, \dots, n$ ) menjumlahkan delta *input*.

$$\delta_{net_l} = \sum_{l=1}^k \delta_l w_{jk} \quad (2.13)$$

dengan:

$\delta_{net_l}$  : nilai untuk menghitung kesalahan *hidden layer*

$w_{jk}$  : bobot antara *hidden layer* ke- $j$  dan *output layer* ke- $k$

Turunan dari fungsi aktivasi dikalikan dengan nilai kesalahan pada *hidden layer* untuk menghitung informasi error

$$\delta_l = \delta_{net_l} f'(Z_{net_j}) = \delta_{net_l} z_j (1 - z_j) \quad (2.14)$$

dengan:

$\delta_l$  : faktor koreksi error bobot  $v_{ij}$

Menghitung koreksi bobot (digunakan untuk memperbaharui  $v_{ij}$ ).

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (2.15)$$

dengan:

$\Delta v_{ij}$  : nilai koreksi error bobot  $v_{ij}$

$x_i$  : nilai *input* ke- $i$

Menghitung koreksi bias (digunakan untuk memperbaharui  $v_{0j}$ ).

$$\Delta v_{0j} = \alpha \delta_j \quad (2.16)$$

dengan:

$\Delta v_{0j}$  : nilai koreksi error bias

3. Fase perubahan bobot dan bias yaitu fase dilakukannya modifikasi bobot untuk menurunkan kesalahan yang terjadi.

Setiap unit *output*  $Y_l$  memperbaharui bias dan bobot ( $l = 1, 2, \dots, k$ )

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (2.17)$$

$$w_{0k}(\text{baru}) = w_{0k}(\text{lama}) + \Delta w_{0k} \quad (2.18)$$

Setiap *hidden* unit  $Z_j$  memperbaharui bobot dan bias ( $j = 1, 2, \dots, n$ )

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (2.19)$$

$$v_{0j}(\text{baru}) = v_{0j}(\text{lama}) + \Delta v_{0j} \quad (2.20)$$

## 2.4 Fungsi Aktivasi

Fungsi aktivasi adalah *output* yang didapatkan pada sebuah *neuron* dengan menggunakan langkah aktivasi tertentu berdasarkan *output* dari penggabungan *linier* (Saraswati dkk., 2021). Dalam jaringan saraf tiruan, fungsi aktivasi dipakai untuk menentukan keluaran suatu *neuron* (Rahayu dkk., 2018). Fungsi aktivasi ini memiliki peranan sangat penting dalam suatu jaringan yang penggunaannya sesuai kebutuhan dan target yang diinginkan. Fungsi aktivasi ini menentukan besarnya bobot (Sholikhah, 2021).

Fungsi aktivasi *sigmoid biner* biasanya digunakan dalam NN untuk pelatihan data dengan metode *backpropagation*. Fungsi aktivasi *sigmoid biner* memiliki *range* dari 0 hingga 1. Namun, fungsi ini bisa juga digunakan oleh NN yang nilai *output*-nya 0 atau 1. Rumus fungsi aktivasi *sigmoid biner* sebagai berikut (Achmalia, 2019):

$$y = f(x) = \frac{1}{1 + e^{-x}} \quad (2.21)$$

dengan  $f'(x) = f(x)[1 - f(x)]$ .

## 2.5 Pengukuran Akurasi

Metode prediksi dibutuhkan tingkat akurasi yang tinggi, maka untuk mengetahuinya diperlukan menghitung tingkat kesalahan dalam prediksi. Semakin kecil tingkat kesalahan yang dihasilkan dari hasil prediksi maka semakin baik prediksi (Sinaga dan Irawati, 2018). Metode atau teknik yang digunakan untuk melakukan evaluasi atau validasi adalah *mean absolute percentage error* (MAPE) dan korelasi *pearson*. MAPE merupakan metode yang menghitung kesalahan (*residu*) menggunakan kesalahan pada tiap periode dibagi dengan nilai observasi untuk periode itu. Kemudian, dihitung rata-rat (*mean*) dari kesalahan persentase tersebut (Ryandhi, 2017). MAPE dapat dihitung dengan rumus berikut:

$$MAPE = \frac{\sum_{t=1}^n \left| \frac{x_t - \hat{x}_t}{x_t} \right|}{n} \times 100\% \quad (2.22)$$

Korelasi *pearson* merupakan salah satu pengujian korelasi yang bertujuan untuk mengetahui seberapa erat hubungan dari dua variabel. Nilai positif adalah nilai 1, nilai -1 merupakan nilai negatif, dan 0 merupakan nilai yang tidak terdapat korelasi.

Apabila terdapat nilai korelasi yang mendekati satu maka korelasi antara dua variabel akan semakin kuat. Namun sebaliknya apabila nilai korelasi mendekati nol maka korelasi antara dua variabel lemah (Windarto, 2020). Korelasi *pearson* dapat dihitung dengan rumus berikut

$$r = \frac{\frac{1}{n} \sum_{t=1}^n (x_t - \bar{x})(\hat{x}_t - \bar{\hat{x}})}{\left[ \left( \frac{1}{n} \sum_{t=1}^n (x_t - \bar{x})^2 \right) \left( \frac{1}{n} \sum_{t=1}^n (\hat{x}_t - \bar{\hat{x}})^2 \right) \right]^{\frac{1}{2}}} \quad (2.23)$$

dengan:

$\hat{x}_t$  : nilai prediksi ke- $t$

$x_t$  : nilai aktual ke- $t$

$n$  : banyaknya data

## 2.6 Curah Hujan

Hujan merupakan gejala meteorologi dan unsur klimatologi. Hujan yang sampai ke permukaan tanah dapat diukur dengan mengukur tinggi hujan tersebut berdasarkan volume air hujan per satuan luas. Hasil pengukuran tersebut dinamakan curah hujan (Habinuddin dkk., 2019). Curah hujan merupakan endapan berupa cairan yang jatuh ke permukaan bumi dengan variabilitas yang besar dalam ruang dan waktu. Berdasarkan skala ruang, variabilitasnya sangat dipengaruhi oleh letak geografi, topografi, ketinggian tempat, arah angin umum, dan letak lintang (Faulina, 2014). Curah hujan biasanya disajikan dalam satuan milimeter (mm). Rata-rata curah hujan di Indonesia cukup tinggi karena wilayahnya yang terletak di daerah tropis dan dikelilingi lautan luas (Utami dkk., 2021).

Atmosfer di atas wilayah Indonesia memainkan peranan penting dan unik dalam dinamika atmosfer global. Pada wilayah Indonesia 70% adalah perairan, maka jumlah uap air yang dapat diendapkan sangat besar, sehingga pembentukan awannya unik dan jumlah curah hujannya berfluktuasi dari bulan ke bulan, dari musim ke musim, atau dari tahun ke tahun. Kerumitan dinamika atmosfer ekuator dan keunikan atmosfer benua maritim menyebabkan kesulitan untuk melakukan prediksi cuaca dengan tingkat ketelitian yang tinggi (Tjasyono, 2009).

Pengamatan terhadap curah hujan umumnya dilakukan secara konvensional. Pengamatan dilakukan dengan menakar jumlah curah hujan yang jatuh di

permukaan bumi menggunakan alat penakar hujan. Untuk mendapatkan curah hujan, maka diperlukan stasiun pengamatan cuaca/iklim sebagai tempat observasi di darat (Swarinoto dan Husain, 2012). Curah hujan dapat diukur dengan alat pengukur curah hujan otomatis atau yang manual. Alat-alat pengukur tersebut harus diletakkan pada daerah yang masih alamiah, sehingga curah hujan yang terukur dapat mewakili wilayah yang luas. Salah satu tipe pengukur hujan manual yang banyak dipakai adalah tipe observatorium (obs) atau biasa disebut ombrometer. Curah hujan dari pengukuran alat ini dihitung dari volume air hujan per luas mulut penakar. Alat tipe observatorium ini merupakan alat baku dengan mulut penakar seluas  $100 \text{ cm}^2$  dan dipasang dengan ketinggian mulut penakar 1.2 meter dari permukaan tanah (Aini, 2017).



## BAB III METODOLOGI PENELITIAN

### 3.1 Sumber Data

Data pada penelitian ini merupakan data sekunder yang diperoleh dari badan meteorologi, klimatologi, dan geofisika (BMKG) Makassar. Data yang digunakan merupakan data curah hujan bulanan dalam satuan mm. Adapun data berupa data curah hujan Kabupaten Pangkajene dan Kepulauan pada Pos Hujan Bungoro Sulawesi Selatan periode 2018-2022 dari stasiun BMKG wilayah IV Makassar dengan jumlah data sebanyak 60 data.

### 3.2 Struktur Data

Struktur data yang digunakan dalam penelitian ini terdiri dari bulan, tahun, dan data curah hujan (mm). Struktur dalam penelitian ini ditunjukkan pada Tabel 3.1.

**Tabel 3.1** Struktur Data Penelitian

Tahun	Bulan	Curah Hujan (mm)
2018	Januari	$x_1$
2018	Februari	$x_2$
2018	Maret	$x_3$
$\vdots$	$\vdots$	$\vdots$
2022	Oktober	$x_{58}$
2022	November	$x_{59}$
2022	Desember	$x_{60}$

### 3.3 Tahapan Analisis

Tahapan analisis data yang dilakukan dalam penelitian ini yaitu sebagai berikut:

1. Mendeskripsikan data curah hujan dengan statistika deskriptif dan plot data deret waktu untuk mengetahui karakteristik dan pola data.
2. Melakukan tahap *embedding*, pada tahap ini dilakukan pemilihan terhadap parameter tunggal dekomposisi yaitu *window length* ( $L$ ). *Window length* digunakan untuk membentuk matriks lintasan  $X$  seperti pada Persamaan (2.1).

$$X = \begin{bmatrix} x_1 & x_2 & \cdots & x_K \\ x_2 & x_3 & \cdots & x_{K+1} \\ \vdots & \cdots & \ddots & \vdots \\ x_L & x_{L+1} & \cdots & x_N \end{bmatrix} \quad (2.1)$$

3. Melakukan tahap *singular value decomposition* (SVD) dari matriks lintasan untuk mendapatkan *eigen triple*. Sehingga SVD dari matriks lintasan  $X$  dapat dituliskan seperti pada Persamaan (2.2).

$$X = X_1 + X_2 + \cdots + X_d \quad (2.2)$$

4. Melakukan tahap *grouping*, pada tahap ini berhubungan dengan pemecahan matriks beberapa hasil dekomposisi dengan mengelompokkan *eigen triple* berdasarkan karakteristik setiap komponen.
5. Melakukan tahap *diagonal averaging*, pada tahap ini dilakukan rekonstruksi pada masing-masing matriks menjadi data deret waktu baru dengan panjang  $N$  menurut karakteristik yang dimiliki.
6. Melakukan pemodelan terhadap hasil dekomposisi dengan menggunakan metode *neural network* dengan langkah-langkah sebagai berikut:
- Membagi data menjadi data latih dan data uji dengan perbandingan 80:20. Data latih mencakup data Januari 2018 hingga Desember 2021 kemudian, data uji mencakup data periode Januari 2022 hingga Desember 2022.
  - Mendefinisikan *input* dan target. Berdasarkan penelitian yang dilakukan oleh Amalina (2016) mendefinisikan *input* dengan cara variabel pertama memuat keseluruhan data latih kemudian pada variabel selanjutnya berkurang satu data dan seterusnya dilakukan pengurangan data pada setiap variabel. Kemudian untuk variabel target didefinisikan berdasarkan data bulan selanjutnya pada tiap data variabel *input*.
  - Menentukan banyaknya *neuron* pada *hidden layer*.
  - Inisialisasi bobot dan bias awal dengan nilai acak dengan bantuan komputasi *software* Rstudio.
  - Melakukan proses pelatihan *backpropagation* menggunakan data latih dengan tiga tahapan proses sebagai berikut:

- 1) Fase *feedforward* memiliki langkah awal dengan menghitung nilai sinyal *input* ke *hidden layer* dengan bobot dan biasanya menggunakan Persamaan (2.6).

$$Z_{net_j} = v_{0j} + \sum_{i=1}^d x_i v_{ij} \quad (2.6)$$

- a) Menghitung sinyal *output* pada *hidden layer* menggunakan fungsi aktivasi dengan Persamaan (2.9).

$$z_j = f(Z_{net_j}) = \frac{1}{1 + e^{-Z_{net_j}}} \quad (2.7)$$

- b) Menjumlahkan bobot serta bias dengan sinyal *output* pada *hidden layer* yang menjadi *input* pada *output layer* menggunakan Persamaan (2.10).

$$Y_{net_l} = w_{0k} + \sum_{j=1}^n z_j w_{jk} \quad (2.8)$$

- c) Mengaktifkan sinyal *output* pada *output layer* menggunakan fungsi aktivasi dengan Persamaan (2.11).

$$y_l = f(Y_{net_l}) = \frac{1}{1 + e^{-Y_{net_l}}} \quad (2.9)$$

- 2) Fase *backpropagation* melakukan perhitungan faktor kesalahan menggunakan Persamaan (2.12).

$$\delta_l = (t_l - y_l) f'(y_{net_l}) \quad (2.10)$$

- a) Memperbaiki bobot ( $w_{jk}$ ) dan bias ( $w_{0k}$ ) dengan faktor kesalahan menggunakan Persamaan (2.13) dan (2.14).

$$\Delta w_{jk} = \alpha \delta_l z_j \quad (2.11)$$

$$\Delta w_{0k} = \alpha \delta_l \quad (2.12)$$

- b) Menghitung faktor kesalahan  $\delta_{net_l}$  di setiap *hidden layer* menggunakan Persamaan (2.15).

$$\delta_{net_l} = \sum_{l=1}^k \delta_l w_{jk} \quad (2.13)$$

- c) Mengaktifkan faktor kesalahan menggunakan fungsi aktivasi dengan Persamaan (2.16).

$$\delta_l = \delta_{net_l} f'(z_{net_j}) = \delta_{net_l} z_j (1 - z_j) \quad (2.14)$$

- d) Memperbaiki bobot ( $v_{ij}$ ) dan bias ( $v_{0j}$ ) dengan faktor kesalahan menggunakan Persamaan (2.15) dan (2.16)

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (2.15)$$

$$\Delta v_{0j} = \alpha \delta_j \quad (2.16)$$

- 3) Fase perubahan bobot dan bias dengan cara memodifikasi bobot dan bias menggunakan Persamaan (2.17) hingga (2.20).

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (2.17)$$

$$w_{0k}(\text{baru}) = w_{0k}(\text{lama}) + \Delta w_{0k} \quad (2.18)$$

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (2.19)$$

$$v_{0j}(\text{baru}) = v_{0j}(\text{lama}) + \Delta v_{0j} \quad (2.20)$$

7. Melakukan pengukuran akurasi dengan menggunakan nilai MAPE dan korelasi menggunakan Persamaan (2.22) dan Persamaan (2.23).

$$MAPE = \frac{\sum_{t=1}^n \left| \frac{x_t - \hat{x}_t}{x_t} \right|}{n} \times 100\% \quad (2.22)$$

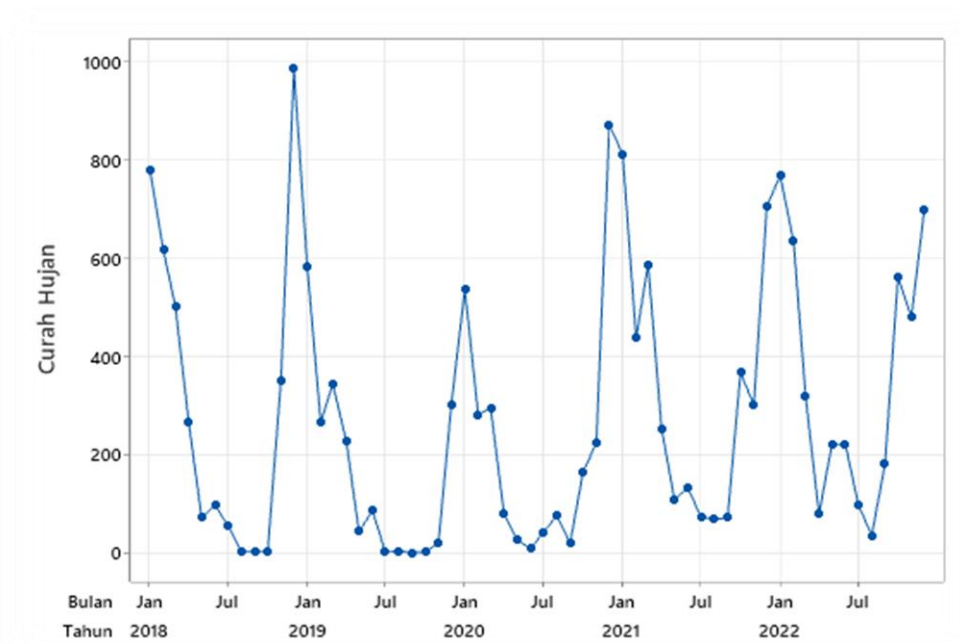
$$r = \frac{\frac{1}{n} \sum_{t=1}^n (x_t - \bar{x})(\hat{x}_t - \bar{\hat{x}})}{\left[ \left( \frac{1}{n} \sum_{t=1}^n (x_t - \bar{x})^2 \right) \left( \frac{1}{n} \sum_{t=1}^n (\hat{x}_t - \bar{\hat{x}})^2 \right) \right]^{\frac{1}{2}}} \quad (2.23)$$

8. Membuat kesimpulan

## BAB IV HASIL DAN PEMBAHASAN

### 4.1 Statistik Deskriptif Data

Data yang dianalisis dalam penelitian ini adalah data curah hujan bulanan di Kabupaten Pangkajene dan Kepulauan yang diambil di pos hujan Bungoro dalam satuan milimeter (mm). Data diperoleh dari stasiun BMKG wilayah IV Makassar tahun 2018 hingga tahun 2020 yang disajikan pada Gambar 4.1.



**Gambar 4.1** Plot Deret Waktu Curah Hujan Kabupaten Pangkajene dan Kepulauan Pos Hujan Bungoro

Gambar 4.1 menunjukkan plot curah hujan Kabupaten Pangkajene dan Kepulauan yang menggambarkan variasi curah hujan setiap tahunnya. Curah hujan memiliki intensitas tinggi pada awal dan akhir tahun, khususnya di bulan Januari dan Desember, serta rendah pada pertengahan tahun, sekitar bulan Juni dan Juli. Dan plot menunjukkan bahwa setiap tahun, periode Januari, Februari bahkan hingga Mei membentuk pola *trend* turun, sementara periode Oktober hingga Desember membentuk pola *trend* naik. Pola yang sama berulang setiap tahun, sehingga mengindikasikan adanya pola musiman dalam data. Pola berulang ini menandakan bahwa data memiliki bentuk teratur yang tidak dapat dijelaskan oleh pendekatan *linier* karena hubungan curah hujan dengan waktu (periode) tidak bersifat *linier*. Oleh karena itu, digunakan pendekatan *non-linier* dalam penanganan

data ini. Karena secara deskriptif terdapat pola *trend* dan musiman, metode yang dapat digunakan adalah SSA yang secara umum mampu untuk mendekomposisi suatu deret data menjadi pola *trend*, musiman dan *noise*. Kemudian dilakukan peramalan dengan menggunakan salah satu metode pendekatan *non-linier* yaitu NN. Sehingga, metode yang digunakan adalah pendekatan *hybrid* antara SSA dan NN. Selanjutnya, data curah hujan dianalisis secara deskriptif dan hasilnya ditampilkan dalam Tabel 4.1.

**Tabel 4.1** Analisis Deskriptif Curah Hujan Kabupaten Pangkajene dan Kepulauan Pos Hujan Bungoro

Deskripsi	2018	2019	2020	2021	2022	Total
Minimum	1	0	8	68	35	0
Maximum	984	581	870	810	767	984
Range	983	581	862	742	732	984
Mean	310.666	156.083	218.083	325.667	357.25	273.55
Standar Deviasi	338.660	187.309	257.666	260.271	258.474	266.60

Tabel 4.1 menunjukkan curah hujan Kabupaten Pangkajene dan Kepulauan terendah terjadi di Tahun 2019 dengan curah hujan 0 dan tertinggi di Tahun 2018 mencapai 984 mm (sangat tinggi). Kemudian data dengan variasi terbesar pada Tahun 2018 dan variasi terkecil pada Tahun 2019. Adapun nilai rata-rata tertinggi berada di tahun 2022 dan terendah berada di tahun 2019. Diperoleh nilai keragaman paling besar di Tahun 2018 sebesar 338.660 dan paling kecil di Tahun 2019 sebesar 187.309. Sejak Januari 2018 hingga Desember 2022 rata-rata curah hujan mencapai 273.55 dengan nilai keragaman 266.60 dan variasi data yang cukup besar yaitu 984.

#### 4.2 *Hybrid Singular Spectrum Analysis dan Neural Network*

Selanjutnya, analisis untuk mendapatkan peramalan curah hujan. Karena pola data menunjukkan bahwa data tidak dapat dijelaskan dengan pendekatan *linier*, maka digunakan metode dengan pendekatan *non-linier*, yaitu pendekatan *hybrid* antara metode SSA dan NN. Metode SSA adalah metode yang mampu mendekomposisi suatu deret menjadi beberapa komponen, seperti komponen *trend*, musiman, dan *noise*. Pada analisis ini, SSA digunakan sebagai tahapan pra-pemrosesan data. Hasil dekomposisi pola-pola yang terbentuk oleh SSA selanjutnya digunakan untuk peramalan menggunakan metode NN. NN didesain untuk menjadi sistem pemrosesan yang menirukan cara kerja otak

manusia dalam menyelesaikan masalah. Proses pelatihan (*training*) adalah bagian terpenting dalam NN, yang bertujuan untuk mengatur bobot-bobot pada NN agar diperoleh bobot akhir yang tepat sesuai dengan pola data yang dilatih.

#### 4.2.1 Singular Spectrum Analysis

Tahap *singular spectrum analysis* terdiri dari 4 tahap diantaranya:

##### 4.2.1.1 Embedding

Tahap *embedding* melibatkan proses analisis yang mengubah data satu dimensi menjadi data multidimensi. Langkah awal adalah membentuk 60 data yaitu data dari Januari 2018 hingga Desember 2022, menjadi susunan data sebagai berikut:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{59} \\ x_{60} \end{bmatrix} = \begin{bmatrix} 779 \\ 617 \\ \vdots \\ 479 \\ 697 \end{bmatrix}$$

Selanjutnya, mengubah data menjadi data multidimensi yang disebut sebagai matriks lintasan dengan dimensi  $L \times K$ . Penentuan nilai parameter *window length* ( $L$ ) didasarkan pada nilai MAPE dengan nilai  $L$  yang dianalisis berada di rentang  $2 < L < 30$  ( $2 < L < \frac{N}{2}$ ). Proses penentuan nilai  $L$  dilakukan dengan *trial and error*, penentuan yang pertama dilakukan pada nilai  $L$  10, 20, dan 30. Hasil *trial and error* nilai  $L$  dapat dilihat pada Tabel 4.2.

**Tabel 4.2** Hasil MAPE berdasarkan nilai  $L$

L	10	<b>20</b>	30
MAPE	60.13	<b>19.91</b>	67.36

Tabel 4.2 menunjukkan nilai MAPE berdasarkan pemilihan nilai  $L$  dari proses *trial and error* pertama. Diperoleh nilai  $L$  dengan MAPE minimum yaitu 20. Kemudian, *trial and error* selanjutnya nilai  $L$  di *tracking* di sekitar nilai 20 untuk mendapatkan nilai  $L$  yang lebih tepat. Hasil *tracking* nilai  $L$  disajikan pada Tabel 4.3.

**Tabel 4.3** Hasil *Tracking* MAPE berdasarkan nilai  $L$

L	16	17	18	19	20	21	22	<b>23</b>	24
MAPE	30.34	19.61	27.63	40.99	19.91	32.50	22.65	<b>19.19</b>	32.68

Hasil *tracking* disekitar nilai 20 ditunjukkan pada Tabel 4.3 dan hasil prediksi setiap nilai  $L$  ditampilkan pada Lampiran 2. Diperoleh nilai dengan MAPE minimum sebesar 19.19% pada nilai  $L = 23$ . Dari nilai  $L$  diperoleh nilai  $K = 38$  ( $K = N - L + 1$ ) dengan nilai  $N = 60$ . Selanjutnya, dengan nilai  $L = 23$  dan  $K = 38$  yang diperoleh dapat dibentuk matriks lintasan  $X$ , yang dituliskan sebagai berikut:

$$X_{23 \times 38} = \begin{bmatrix} 779 & 617 & 502 & \dots & 436 \\ 617 & 502 & 267 & \dots & 586 \\ 502 & 267 & 71 & \dots & 253 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 20 & 302 & 536 & \dots & 697 \end{bmatrix}$$

#### 4.2.1.2 Singular Value Decomposition

Setelah membentuk matriks lintasan kemudian akan dihitung *singular value decomposition*. Dalam tahap ini matriks lintasan di dekomposisi menjadi *eigen triple* yang terdiri atas *singular value*, *eigenvector*, dan *principal component*. Langkah awal yaitu membentuk matriks simetris, matriks simetris didapatkan dari hasil kali matriks  $X$  dengan matriks  $X^T$ , kemudian dihitung nilai-nilai *eigen triple*. Matriks simetris yang terbentuk akan berukuran  $L \times L$  yang dituliskan sebagai berikut:

$$\begin{aligned} S &= X_{(23 \times 38)} \cdot X_{(38 \times 23)}^T \\ S_{(23 \times 23)} &= \begin{bmatrix} 779 & 617 & 502 & \dots & 436 \\ 617 & 502 & 267 & \dots & 586 \\ 502 & 267 & 71 & \dots & 253 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 20 & 302 & 536 & \dots & 697 \end{bmatrix} \cdot \begin{bmatrix} 779 & 617 & 502 & \dots & 20 \\ 617 & 502 & 267 & \dots & 302 \\ 502 & 267 & 71 & \dots & 536 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 436 & 586 & 253 & \dots & 697 \end{bmatrix} \\ &= \begin{bmatrix} 5245278 & 4189559 & 2973612 & \dots & 3538038 \\ 4189559 & 4981833 & 3857174 & \dots & 2978819 \\ 2973612 & 3857174 & 4665153 & \dots & 1955461 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 3538038 & 2978819 & 1955461 & \dots & 5676932 \end{bmatrix}_{(23 \times 23)} \end{aligned}$$

##### a. Singular Value ( $\sqrt{\lambda_i}$ )

Dari matriks simetris  $S$  yang diperoleh dihitung *eigenvalue* dengan menggunakan persamaan karakteristik untuk mendapatkan nilai *eigenvalue*. Selanjutnya, dari *eigenvalue* yang telah diperoleh, dihitung *singular value* yang merupakan akar kuadrat dari *eigenvalue*. *Singular value* kemudian diurutkan secara menurun, dan hasilnya disajikan pada Tabel 4.4.



**Tabel 4.4 Eigenvalues dan Singular Values**

No	Eigenvalues ( $\lambda_i$ )	Singular Values ( $\sqrt{\lambda_i}$ )
1	7429.9076	86.19691
2	4277.7075	65.40419
3	4125.3369	64.22879
4	1738.2489	41.69231
⋮	⋮	⋮
23	288.9948	16.99985

b. *Eigenvector ( $U_i$ )*

Setelah mendapatkan *singular value*, nilai tersebut kemudian digunakan untuk menghitung nilai *eigenvector* dari *eigenvalue* yang bersesuaian. Hasil dari perhitungan *eigenvector* disajikan pada Tabel 4.5.

**Tabel 4.5 Eigenvector**

No	$U_1$	$U_2$	$U_3$	$U_4$	...	$U_{23}$
1	0.2071	0.3125	-0.0327	-0.3891	...	0.0937
2	0.2075	0.2849	0.1109	-0.2950	...	-0.1902
3	0.1989	0.1778	0.2391	0.00269	...	0.2507
4	0.1894	0.0276	0.2973	0.2067	...	-0.2931
⋮	⋮	⋮	⋮	⋮	⋮	⋮
23	0.2479	0.1432	-0.2518	0.2256	...	0.0181

c. *Principal Component ( $V_i$ )*

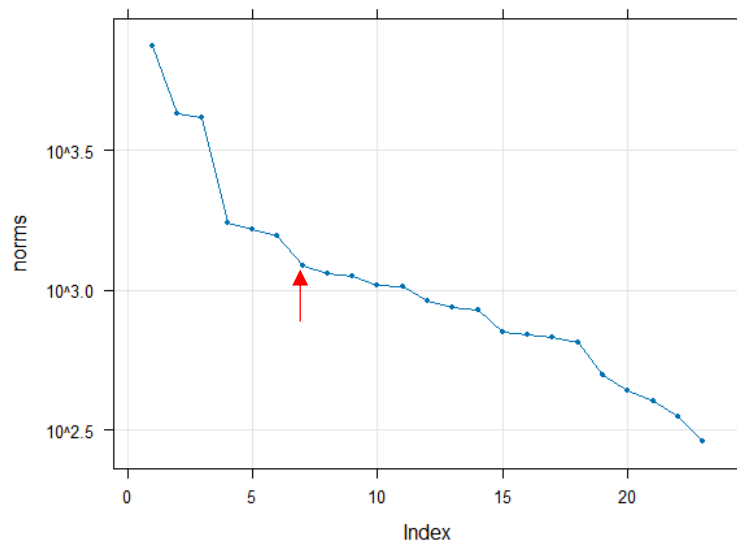
Nilai *eigenvalue* dan *eigenvector* yang telah didapatkan kemudian digunakan menghitung nilai *principal component*. Untuk mendapatkan nilai *principal component* dihitung menggunakan persamaan  $V_i$  yang telah dijelaskan pada Sub Bab 2.1.2. Nilai  $V_i$  didapatkan dengan mengalikan matriks  $S$  yang telah ditranspos dan nilai *eigenvector* kemudian dibagi dengan *singular value* yang telah didapatkan sebelumnya. Dengan menggunakan persamaan tersebut, diperoleh nilai *principal component* dan hasilnya disajikan pada Tabel 4.6.

**Tabel 4.6** *Principal Component*

No	$V_1$	$V_2$	$V_3$	$V_4$	...	$V_{23}$
1	0.1525	0.2380	0.0496	-0.2899	...	-0.1941
2	0.1403	0.1997	-0.0632	-0.1369	...	0.1832
3	0.1375	0.1237	-0.1625	0.0712	...	-0.1750
4	0.1284	0.0075	-0.1988	0.0905	...	0.2190
⋮	⋮	⋮	⋮	⋮	⋮	⋮
38	0.2160	0.1530	-0.1476	0.0388	...	0.2090

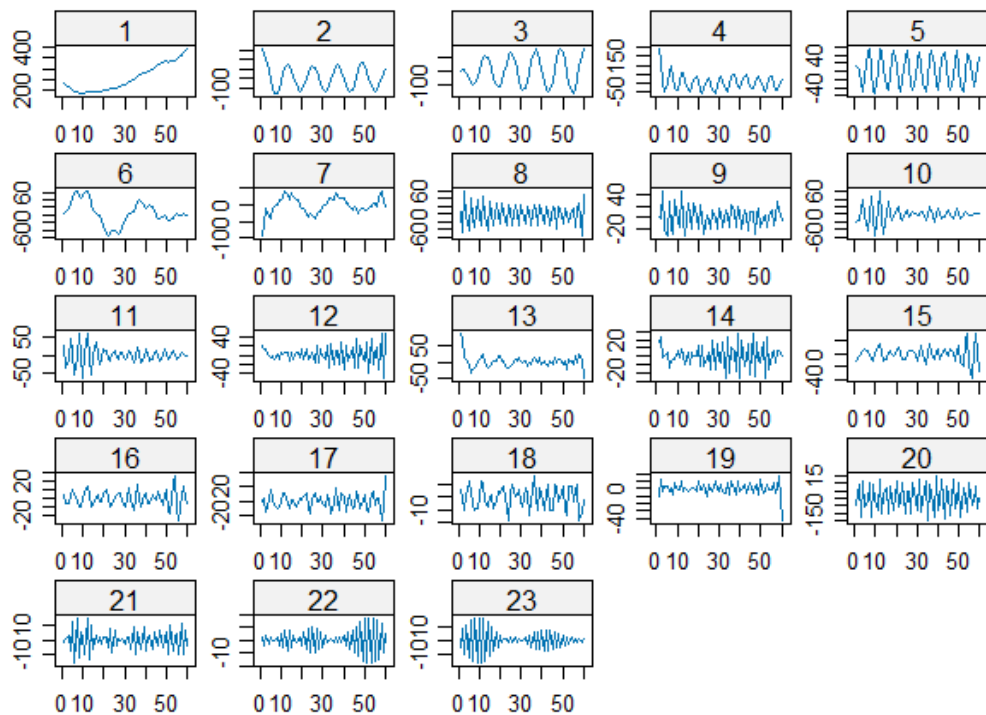
### 4.2.1.3 Grouping

Setelah mendapatkan nilai dari tahapan SVD maka dilakukan tahap *grouping*. Tahap ini dilakukan pengelompokan *eigen triple* hasil SVD berdasarkan karakteristik setiap komponen yang berhubungan dengan *trend*, musiman, dan *noise*. Langkah awal dalam tahapan *grouping* yaitu menentukan parameter *grouping effect* ( $r$ ) yang digunakan untuk membatasi jumlah *eigen triple* yang digunakan dalam mendefinisikan komponen *trend* dan musiman. Berikut adalah grafik *singular value* yang digunakan untuk menentukan nilai  $r$ .

**Gambar 4.2** Grafik *Singular Value*

Nilai parameter  $r$  ditentukan berdasarkan jumlah *eigen triple* yang tidak mencerminkan *noise* pada grafik. Diketahui bahwa grafik yang menurun secara lambat atau perlahan dari *singular value* biasanya terkait dengan komponen *noise*. Gambar 4.2

menunjukkan *singular value* menurun secara tajam pada *eigen triple* 1 hingga 7, kemudian mulai menurun secara lambat pada *eigen triple* 8 hingga 23, sehingga pada *eigen triple* tersebut diidentifikasi sebagai komponen *noise*. Namun, perlu dilakukan pemeriksaan lebih lanjut untuk memastikan bahwa tidak ada komponen *noise* tambahan yang terkait dengan *eigen triple* yang tidak berhubungan dengan komponen *trend* dan musiman. Dalam hal ini, dilakukan pemeriksaan dengan membentuk plot *eigen vector* dari setiap nilai *eigen triple*. Jika terdapat *eigen vector* yang memiliki karakteristik yang relatif sama, maka *eigen vector* tersebut akan dikelompokkan ke dalam satu kelompok sebagai komponen yang saling terkait.



**Gambar 4.3** Plot *Eigen vector*

Gambar 4.3 menunjukkan bahwa deret yang terbentuk dari *eigen triple* 1 menampilkan komponen yang bervariasi secara lambat atau pola menunjukkan peningkatan jangka panjang, sehingga *eigen triple* 1 dikelompokkan ke dalam komponen *trend*. Selanjutnya, berdasarkan teori yang dikemukakan oleh Hassani (2007), *singular value* dari dua *eigen triple* yang mengandung komponen musiman seringkali sangat dekat satu sama lain. Oleh karena itu, komponen musiman dikelompokkan berdasarkan kemiripan *singular value* dari dua *eigen triple* yang berdekatan, yaitu *eigen triple* 2 dan 3, 4 dan 5, 6 dan 7, 10 dan 11, serta 15 dan 16. Dengan demikian, *eigen triple* yang tidak

dikelompokkan ke dalam *trend* ataupun musiman dikelompokkan ke dalam komponen *noise*. Hasil dari pengelompokan ini ditunjukkan pada Tabel 4.7.

**Tabel 4.7** *Eigentriple* dan Komponen yang Terkait

Komponen	<i>Eigentriple</i>
<i>Trend</i>	1
Musiman	2,3,4,5,6,7,10,11,15,16
<i>Noise</i>	8,9,12,13,14,17,18,19,20,21,22,23

Tabel 4.7 menunjukkan hasil dari tahapan *grouping* yang merupakan pengelompokan dari *eigentriple* hasil SVD berdasarkan karakteristik setiap komponen yang saling berhubungan.

#### 4.2.1.4 *Diagonal Averaging*

Dari tahap *grouping* kemudian dilakukan tahap *diagonal averaging*. Tahap *diagonal averaging* adalah proses membentuk data deret dengan satu dimensi dengan panjang  $N$  berdasarkan komponen terkait yang dihasilkan sebelumnya. Setiap komponen digabungkan sesuai dengan pola kelompok masing-masing. Hasil rekonstruksi dari tahap ini dihitung menggunakan Persamaan (2.5) sebagaimana terlampir pada Lampiran 3. *Diagonal averaging* dapat dihitung dengan menjumlahkan hasil rekonstruksi setiap komponen. Hasil perhitungan tersebut kemudian ditampilkan dalam Tabel 4.8 sebagai berikut.

**Tabel 4.8** Hasil Rekonstruksi

Data	<i>Trend</i>	Musiman	<i>Noise</i>
1	234.87	424.74	119.37
2	225.70	302.38	88.916
3	217.92	195.58	88.48
4	208.02	61.036	-2.059
5	200.43	-72.265	-57.167
⋮	⋮	⋮	⋮
58	380.88	77.129	102.98
59	386.17	185.26	-92.442
60	398.11	255.003	43.878

Tabel 4.8 menunjukkan hasil rekonstruksi berdasarkan perhitungan dari setiap komponen *trend*, musiman, dan *noise*. Tahapan selanjutnya adalah melakukan pemodelan terhadap hasil rekonstruksi dari Tabel 4.8 menggunakan metode NN.

#### 4.2.2 *Neural Network*

Sebelum melakukan tahapan analisis pada NN, data dibagi menjadi dua bagian yaitu data latih dan data uji. Data latih digunakan dalam proses pembentukan arsitektur terbaik NN. Sementara, data uji digunakan untuk memvalidasi keakuratan peramalan pada tahapan akurasi peramalan. Setelah itu, diperlukan penentuan variabel terkait yang dijadikan *input* dan target. Variabel *input* digunakan dalam pembentukan arsitektur *neural network* untuk memodelkan setiap komponen (*trend*, musiman, *noise*). Sedangkan, variabel target digunakan untuk menghitung eror dari *output* yang dihasilkan dengan cara melakukan pengurangan diantara keduanya. Eror kemudian dijadikan sebagai dasar dalam melakukan perubahan dari setiap bobot dengan mempropagasikannya kembali. Dalam penentuan variabel, terpilih sebanyak 6 variabel *input* ( $X$ ) dan 1 variabel target ( $Y$ ) untuk digunakan dalam pembentukan arsitektur *neural network*.

**Tabel 4.9** Variabel *Input* dan Target

Variabel	Periode Data
$X_1$	Januari 2018 – Desember 2021
$X_2$	Januari 2018 – November 2021
$X_3$	Januari 2018 – Oktober 2021
$X_4$	Januari 2018 – September 2021
$X_5$	Januari 2018 – Agustus 2021
$X_6$	Januari 2018 – Juli 2021
$Y$	Februari 2018 – Desember 2021

Variabel *input* dan target pada Tabel 4.9 memiliki jumlah periode data yang berbeda namun nantinya dalam proses pelatihan jumlah data akan terhitung sebanyak 48 data dengan ketentuan data dimasukkan berulang mulai dari awal hingga memenuhi jumlah data latih yang seharusnya. Setelah pendefinisian *input*, langkah selanjutnya adalah menentukan banyaknya *neuron* yang akan digunakan pada *hidden layer*. Pada *hidden layer* ini menggunakan satu lapisan dengan fungsi aktivasi *sigmoid biner*. Jumlah

neuron pada *hidden layer* ditentukan melalui proses *trial and error*, dengan rentang jumlah *neuron* yang diuji antara 1 hingga 10. Dalam penentuan jumlah *neuron* ini dipilih nilai MAPE minimum sebagai kriteria penentuan yang optimal.

**Tabel 4.10** Hasil MAPE berdasarkan Neuron

Data	<i>Neuron Hidden</i>	MAPE
Komponen <i>Trend</i>	1	4.256
	2	3.242
	3	2.495
	4	<b>2.064</b>
	5	2.787
	6	3.979
	7	3.127
	8	5.222
	9	5.723
	10	2.433
Komponen Musiman	1	25.088
	2	26.033
	3	<b>22.791</b>
	4	25.038
	5	23.543
	6	25.698
	7	23.247
	8	24.808
	9	23.620
	10	23.133
Komponen <i>Noise</i>	1	188.657
	2	173.817
	3	166.134
	4	171.907
	5	166.596
	6	156.796
	7	157.056
	8	158.205
	9	159.456
	10	<b>150.1885</b>

Tabel 4.10 menunjukkan hasil MAPE dari proses *trial and error* yang telah diuji dari *neuron* 1 hingga 10 pada setiap komponen. Berdasarkan Tabel 4.10 arsitektur *neural network* yang terbentuk pada komponen *trend* yaitu 6-4-1 yang artinya 6

*neuron input*, 4 *neuron hidden*, dan 1 *neuron output* dengan nilai MAPE sebesar 2.06%. Pada komponen musiman membentuk arsitektur 6-3-1 yang artinya 6 *neuron input*, 3 *neuron hidden*, dan 1 *neuron output* dengan nilai MAPE sebesar 22.79%. Dan pada komponen *noise* membentuk arsitektur 6-10-1 yang artinya 6 *neuron input*, 10 *neuron hidden*, dan 1 *neuron output* dengan nilai MAPE sebesar 150.18%.

Kemudian, langkah selanjutnya adalah melakukan inisialisasi bobot dan bias sebelum memulai proses pelatihan pada NN. Penentuan bobot awal dilakukan dengan cara menginisialisasi secara acak bilangan yang diberikan pada setiap *neuron* yang berhubungan, mulai pada *neuron input layer* ke *neuron hidden layer* maupun *neuron hidden layer* ke *neuron output layer*. Informasi mengenai nilai bobot awal dan bias melalui metode inisialisasi acak tertera pada Lampiran 4. Dalam Tabel 4.11 hingga Tabel 4.12 menyajikan bobot dan bias awal pada komponen *trend*.

**Tabel 4.11** Bobot Awal dan Bias *Input Layer* ke *Hidden Layer* pada Komponen *Trend*

$v[i, ]$	$v[, j]$			
	1	2	3	4
0	-0.6264	0.7383	1.1249	0.7821
1	0.1836	0.5757	-0.0449	0.0745
2	-0.8356	-0.3053	-0.0161	-1.9893
3	1.5952	1.5117	0.9438	0.6198
4	0.3295	0.3898	0.8212	-0.056
5	-0.8204	-0.6212	0.5939	-0.1557
6	0.4874	-2.2146	0.9189	-1.4707

**Tabel 4.12** Bobot Awal dan Bias *Hidden Layer* ke *Output Layer* pada Komponen *Trend*

$w[j, ]$	$w[, k]$
	1
0	-0.4781
1	0.4179
2	1.3586
3	-0.1027
4	0.3876

Tabel 4.11 menyajikan nilai bobot dan bias awal dari *input layer* ke *hidden layer*, sementara Tabel 4.12 menyajikan nilai bobot awal dan bias dari *hidden layer* ke *output layer* pada komponen *trend*. Proses perhitungan pada NN yang disajikan dilakukan dengan menggunakan data komponen *trend* dan hanya pada data latih ke-1. Proses perhitungan dilakukan dengan kondisi berhenti hingga mencapai eror minimum berdasarkan *output* pada *software* yang digunakan. Namun, dalam perhitungan yang disajikan ini hanya dilakukan 1 kali iterasi.

#### 4.2.2.1 Feedforward

Proses inisialisasi bobot dan bias yang telah dilakukan, kemudian dilanjutkan dengan proses normalisasi data. Normalisasi bertujuan agar data berada pada rentang yang tidak terlalu jauh, namun tetap tidak kehilangan karakteristik data asli. Data telah dikelompokkan berdasarkan variabel *input* dan target sebelum dilakukan normalisasi. Proses normalisasi dilakukan untuk mengaktifkan fungsi aktivasi *sigmoid biner* pada NN. Hasil normalisasi ditunjukkan dalam Tabel 4.13 yang merujuk pada Lampiran 5.

**Tabel 4.13** Hasil Normalisasi

No	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_6$	$Y$
1	0.2409	0.2409	0.2409	0.2409	0.2409	0.2409	0.1982
2	0.1982	0.1982	0.1982	0.1982	0.1982	0.1982	0.1621
3	0.1621	0.1621	0.1621	0.1621	0.1621	0.1621	0.1160
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
47	0.6567	0.6567	0.2409	0.1982	0.1621	0.1160	0.6844
48	0.6844	0.2409	0.1982	0.1621	0.1160	0.0807	0.1982

Proses *feedforward* digunakan untuk menghitung nilai dari *output layer*. Setiap unit di *input layer* menerima sinyal *input* dan menyebarkan sinyal tersebut pada tiap unit di *hidden layer*. Kemudian setiap unit di *hidden layer* menyebarkan sinyal yang diterima pada tiap unit di *output layer*. Dengan menggunakan fungsi aktivasi diperoleh nilai dari *output layer*. Prosedur *feedforward* dilakukan dengan langkah-langkah berikut:

Menghitung sinyal *input* pada *hidden layer* ( $Z_{net_j}$ ) menggunakan Persamaan (2.6).

Untuk menghitung sinyal *input* pada *hidden layer* memerlukan bobot dan bias awal yang terdapat pada Tabel 4.11 dan data hasil normalisasi data latih ke-1 yang terdapat



pada Tabel 4.13. Setiap *neuron* pada *hidden layer* memiliki sinyal *input* masing-masing, sehingga pada perhitungan ini terdapat empat sinyal *input* yang dihasilkan.

$$\begin{aligned}
 Z_{net_1} &= v_{01} + x_1v_{11} + x_2v_{21} + x_3v_{31} + x_4v_{41} + x_5v_{51} + x_6v_{61} \\
 &= (-0.6264) + 0.2409 \times 0.1836 + 0.2409 \times (-0.8356) + \\
 &\quad 0.2409 \times 1.5952 + 0.2409 \times 0.3295 + 0.2409 \times \\
 &\quad (-0.8204) + 0.2409 \times 0.487 \\
 &= -0.3999
 \end{aligned}$$

$$\begin{aligned}
 Z_{net_2} &= v_{02} + x_1v_{12} + x_2v_{22} + x_3v_{32} + x_4v_{42} + x_5v_{52} + x_6v_{62} \\
 &= 0.7383 + 0.2409 \times 0.5757 + 0.2409 \times (-0.3053) \\
 &\quad + 0.2409 \times 1.5117 + 0.2409 \times 0.3898 \\
 &\quad + 0.2409 \times (-0.6212) + 0.2409 \times (-2.2146) \\
 &= 0.5783
 \end{aligned}$$

$$\begin{aligned}
 Z_{net_3} &= v_{03} + x_1v_{13} + x_2v_{23} + x_3v_{33} + x_4v_{43} + x_5v_{53} + x_6v_{63} \\
 &= 1.1249 + 0.2409 \times (-0.0449) + 0.2409 \times (-0.0161) \\
 &\quad + 0.2409 \times 0.9438 + 0.2409 \times 0.8212 \\
 &\quad + 0.2409 \times 0.5939 + 0.2409 \times 0.9189 \\
 &= 1.8999
 \end{aligned}$$

$$\begin{aligned}
 Z_{net_4} &= v_{04} + x_1v_{14} + x_2v_{24} + x_3v_{34} + x_4v_{44} + x_5v_{54} + x_6v_{64} \\
 &= 0.7821 + 0.2409 \times 0.0745 + 0.2409 \times (-1.9893) \\
 &\quad + 0.2409 \times 0.6198 + 0.2409 \times (-0.056) \\
 &\quad + 0.2409 \times (-0.1557) + 0.2409 \times (-1.4707) \\
 &= 0.0647
 \end{aligned}$$

Sinyal *input* yang didapatkan selanjutnya digunakan untuk menghitung fungsi aktivasi *hidden layer* atau sinyal *output* di *hidden layer* ( $z_j$ ) menggunakan Persamaan (2.7).

$$\begin{aligned}
 z_1 &= f(Z_{net_1}) = \frac{1}{1 + e^{0.3999}} = 0.4013 \\
 z_2 &= f(Z_{net_2}) = \frac{1}{1 + e^{-0.5783}} = 0.6406
 \end{aligned}$$

$$z_3 = f(Z_{net_3}) = \frac{1}{1 + e^{-1.8999}} = 0.8698$$

$$z_4 = f(Z_{net_4}) = \frac{1}{1 + e^{-0.0647}} = 0.5161$$

Perhitungan untuk sinyal *input* pada *output layer* memerlukan bobot dan bias awal yang terdapat pada Tabel 4.12. Sinyal *input* pada *output layer* ( $Y_{net_1}$ ) dihitung menggunakan Persamaan (2.8).

$$\begin{aligned} Y_{net_1} &= w_{01} + z_1w_{11} + z_2w_{21} + z_3w_{31} + z_4w_{41} \\ &= (-0.4781) + 0.4013 \times 0.4179 + 0.6406 \times 1.3586 + \\ &\quad 0.8698 \times (-0.1027) + 0.5161 \times 0.3876 \\ &= 0.6707 \end{aligned}$$

Sinyal *input* yang dihasilkan untuk *output layer* kemudian digunakan untuk menghitung fungsi aktivasi *output layer* atau sinyal *output* di *output layer* ( $y_1$ ) menggunakan Persamaan (2.9).

$$y_1 = f(Y_{net_1}) = \frac{1}{1 + e^{-0.6707}} = 0.6616$$

#### 4.2.2.2 Backpropagation

Tahapan setelah dilakukan *feedforward* adalah tahap *backpropagation*. Pada tahap ini, dilakukan perhitungan kesalahan bobot mulai dari *output layer* ke *hidden layer* kemudian ke *input layer*. *Backpropagation* menggunakan error dari *output* untuk mengubah nilai bobot-bobotnya dalam arah mundur. Tahapan *backpropagation* dilakukan dengan langkah-langkah sebagai berikut:

Menghitung faktor kesalahan bobot  $w_{jk}$  ( $\delta_l$ ) pada *output* berdasarkan error di setiap *output* ( $y_l$ ) dengan menggunakan Persamaan (2.10). Untuk menghitung faktor kesalahan digunakan data pada variabel target yang telah dinormalisasi pada Tabel 4.13.

$$\begin{aligned} \delta_1 &= (t_1 - y_1)f'(Y_{net_1}) = (t_1 - y_1)y_1(1 - y_1) \\ &= (0.1982 - 0.6616)0.6616(1 - 0.6616) = -0.1037 \end{aligned}$$

Hasil dari faktor kesalahan kemudian digunakan untuk menghitung nilai kesalahan pada bobot  $w_{jk}(\Delta w_{jk})$ . Kemudian nilai kesalahan pada bobot nantinya digunakan untuk memperbaharui bobot  $w_{jk}$  dengan *learning rate* ( $\alpha$ ) sebesar 0.01 menggunakan Persamaan (2.11).

$$\begin{aligned}\Delta w_{11} &= \alpha \delta_1 z_1 = 0.01 \times (-0.1037) \times 0.4013 = -0.000416 \\ \Delta w_{21} &= \alpha \delta_1 z_2 = 0.01 \times (-0.1037) \times 0.6406 = -0.000664 \\ \Delta w_{31} &= \alpha \delta_1 z_3 = 0.01 \times (-0.1037) \times 0.8698 = -0.000902 \\ \Delta w_{41} &= \alpha \delta_1 z_4 = 0.01 \times (-0.1037) \times 0.5161 = -0.000535\end{aligned}$$

Selain itu juga menghitung nilai kesalahan bias  $w_{0k}(\Delta w_{0k})$  yang digunakan untuk memperbaharui bias  $w_{0k}$  menggunakan Persamaan (2.12).

$$\Delta w_{01} = \alpha \delta_1 = 0.01 \times (-0.1037) = -0.001037$$

Setelah perhitungan kesalahan pada *output layer* selanjutnya menghitung jumlah faktor kesalahan pada bobot *hidden layer* berdasarkan eror di setiap *hidden layer* menggunakan Persamaan (2.13).

$$\begin{aligned}\delta_{net_1} &= \delta_1 w_{11} = (-0.1037) \times 0.4179 = -0.0433 \\ \delta_{net_2} &= \delta_1 w_{21} = (-0.1037) \times 1.3586 = -0.1409 \\ \delta_{net_3} &= \delta_1 w_{31} = (-0.1037) \times (-0.1027) = 0.0106 \\ \delta_{net_4} &= \delta_1 w_{41} = (-0.1037) \times 0.3876 = -0.0402\end{aligned}$$

Faktor kesalahan pada bobot *hidden layer* selanjutnya digunakan untuk mengaktifkan faktor kesalahan dengan fungsi aktivasi untuk menentukan *output* yang kemudian akan kembali dijadikan *input* pada lapisan berikutnya menggunakan Persamaan (2.14).

$$\begin{aligned}\delta_1 &= \delta_{net_1} f'(Z_{net_1}) = \delta_{net_1} z_1 (1 - z_1) \\ &= (-0.0433) \times 0.4013 (1 - 0.4013) = -0.0104 \\ \delta_2 &= \delta_{net_2} f'(Z_{net_2}) = \delta_{net_2} z_2 (1 - z_2) \\ &= (-0.1409) \times 0.6406 (1 - 0.6406) = -0.0324 \\ \delta_3 &= \delta_{net_3} f'(Z_{net_3}) = \delta_{net_3} z_3 (1 - z_3) \\ &= 0.0106 \times 0.8698 (1 - 0.8698) = 0.00120\end{aligned}$$

$$\begin{aligned}\delta_4 &= \delta_{net_4} f'(Z_{net_4}) = \delta_{net_4} z_4(1 - z_4) \\ &= (-0.0402) \times 0.5161(1 - 0.5161) = -0.01004\end{aligned}$$

Setelah perhitungan faktor kesalahan bobot dan bias pada *output layer* ke *hidden layer*, selanjutnya dihitung faktor kesalahan bobot dan bias pada *hidden layer* ke *input layer*. Untuk nilai kesalahan bobot  $v_{ij}(\Delta v_{ij})$  yang digunakan untuk memperbaharui bobot  $v_{ij}$  dihitung menggunakan Persamaan (2.15).

$$\Delta v_{11} = \alpha \delta_1 x_1 = 0.01 \times (-0.0104) \times 0.2409 = -0.0000250963$$

Berturut-turut persamaan diatas digunakan untuk menghitung nilai kesalahan bobot pada setiap lapisan yang menghubungkan *input layer* dan *hidden layer* satu sama lain sehingga akan diperoleh nilai kesalahan dari masing-masing *neuron* yang hasilnya ditampilkan pada Tabel 4.14

**Tabel 4.14** Nilai Kesalahan Bobot

<i>i</i>	<i>j</i>			
	1	2	3	4
1	-0.000025	-0.000078	0.0000029	-0.000024
2	-0.000025	-0.000078	0.0000029	-0.000024
3	-0.000025	-0.000078	0.0000029	-0.000024
4	-0.000025	-0.000078	0.0000029	-0.000024
5	-0.000025	-0.000078	0.0000029	-0.000024
6	-0.000025	-0.000078	0.0000029	-0.000024

Selain itu, menghitung nilai kesalahan bias  $v_{0j}(\Delta v_{0j})$  yang digunakan untuk memperbaharui bias  $v_{0j}$  menggunakan Persamaan (2.16).

$$\begin{aligned}\Delta v_{01} &= \alpha \delta_1 = 0.01 \times (-0.0104) = -0.000104 \\ \Delta v_{02} &= \alpha \delta_2 = 0.01 \times (-0.0324) = -0.000324 \\ \Delta v_{03} &= \alpha \delta_3 = 0.01 \times 0.0012 = 0.000012 \\ \Delta v_{04} &= \alpha \delta_4 = 0.01 \times (-0.0100) = -0.00010042\end{aligned}$$

#### 4.2.2.3 Perubahan Bobot dan Bias

Setelah melakukan tahap *backpropagation* atau menghitung faktor kesalahan pada setiap *layer*, tahap selanjutnya adalah proses ubah bobot dan bias, yang merupakan proses akhir dari pelatihan data. Pada tahapan ini, bobot diperbaharui dengan menambahkan bobot lama dengan nilai kesalahan bobot pada setiap bagian atau pada

setiap *neuron* dari masing-masing *layer*. Langkah-langkah dalam proses perubahan bobot dan bias sebagai berikut:

Bobot antara *hidden layer* ke-*j* dan *output layer* ke-*k* baru dihitung menggunakan Persamaan (2.17).

$$\begin{aligned}w_{11}(\text{baru}) &= w_{11}(\text{lama}) + \Delta w_{11} \\ &= 0.4179 - 0.00041 = 0.4174 \\ w_{21}(\text{baru}) &= w_{21}(\text{lama}) + \Delta w_{21} \\ &= 1.358 - 0.00066 = 1.3579 \\ w_{31}(\text{baru}) &= w_{31}(\text{lama}) + \Delta w_{31} \\ &= (-0.1027) - 0.000902 = -0.1036 \\ w_{41}(\text{baru}) &= w_{41}(\text{lama}) + \Delta w_{41} \\ &= 0.3876 - 0.000535 = 0.3870\end{aligned}$$

Bias menuju *output layer* ke-*k* baru dihitung menggunakan Persamaan (2.18).

$$\begin{aligned}w_{01}(\text{baru}) &= w_{01}(\text{lama}) + \Delta w_{01} \\ &= (-0.4781) - 0.00103 = -0.4791\end{aligned}$$

Bobot antara *input layer* ke-*i* dan *hidden layer* ke-*j* baru dihitung menggunakan Persamaan (2.19).

$$\begin{aligned}v_{11}(\text{baru}) &= v_{11}(\text{lama}) + \Delta v_{11} \\ &= 0.1836 - 0.0000250963 = 0.18382\end{aligned}$$

Berturut-turut dilakukan perhitungan dengan persamaan diatas untuk menghitung bobot  $v_{ij}(\text{baru})$ . Bias menuju *hidden layer* ke-*j* baru dihitung menggunakan Persamaan (2.20).

$$\begin{aligned}v_{01}(\text{baru}) &= v_{01}(\text{lama}) + \Delta v_{01} \\ &= (-0.6264) - 0.000104162 = -0.6265 \\ v_{02}(\text{baru}) &= v_{02}(\text{lama}) + \Delta v_{02} \\ &= 0.7383 - 0.000324461 = 0.7379 \\ v_{03}(\text{baru}) &= v_{03}(\text{lama}) + \Delta v_{03} \\ &= 1.1249 + 0.000012059 = 1.1249\end{aligned}$$

$$v_{04}(\text{baru}) = v_{04}(\text{lama}) + \Delta v_{04}$$

$$= 0.7821 - 0.00010042 = 0.7819$$

Pada algoritma ini setelah tahapan propagasi maju selanjutnya dilakukan propagasi mundur untuk melakukan perbaikan bobot, kemudian dilakukan perubahan bobot. Proses tersebut dilakukan terus menerus hingga mencapai minimum eror dan beberapa kali iterasi. Namun pada perhitungan ini hanya menghasilkan nilai bobot dan bias akhir dengan satu kali iterasi. Hasil akhir dari perhitungan bobot dan bias yang diperoleh disajikan pada Tabel 4.15 hingga Tabel 4.16 berikut.

**Tabel 4.15** Bobot Akhir dan Bias *Input Layer* ke *Hidden Layer* (1 kali iterasi)

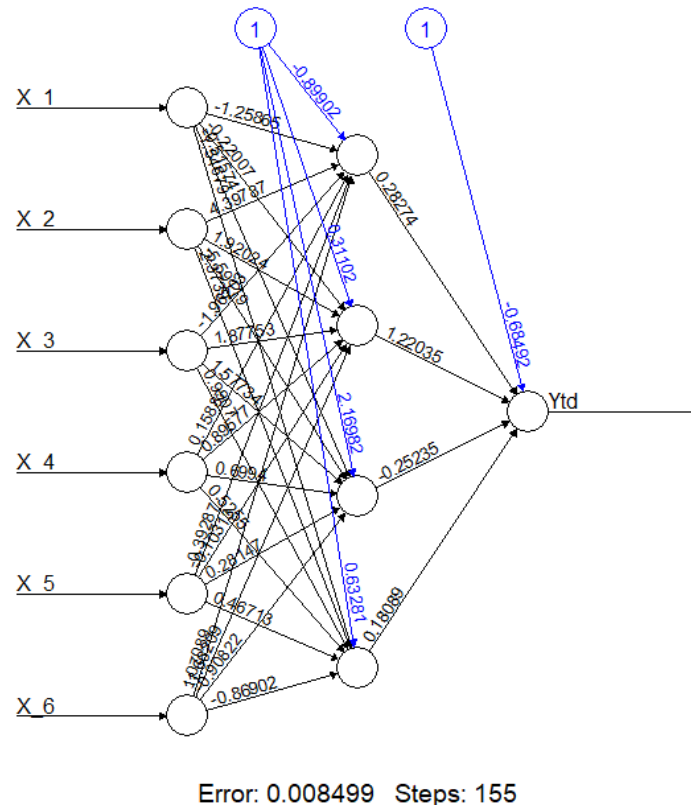
$v[i,]$	$v[,j]$			
	1	2	3	4
0	-0.6265	0.737976	1.124912	0.782
1	0.183575	0.575622	-0.0449	0.074476
2	-0.83563	-0.30538	-0.0161	-1.98932
3	1.595175	1.511622	0.943803	0.619776
4	0.329475	0.389722	0.821203	-0.05602
5	-0.82043	-0.62128	0.593903	-0.15572
6	0.487375	-2.21468	0.918903	-1.47072

**Tabel 4.16** Bobot Akhir dan Bias *Hidden Layer* ke *Output Layer* (1 kali iterasi)

$w[j,]$	$w[,k]$
	1
0	0.41748367
1	1.35793534
2	-0.1036024
3	0.38706451
4	-0.4791374

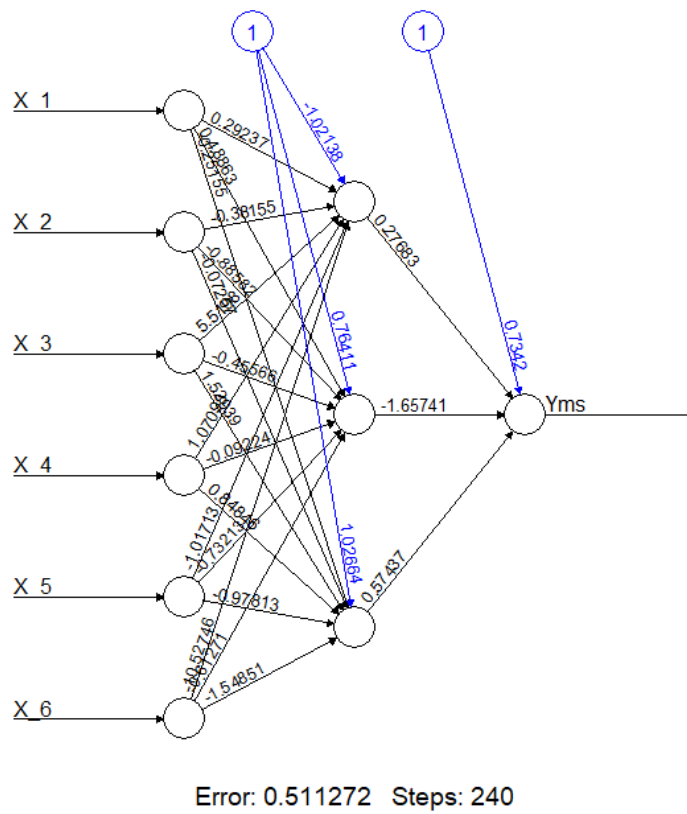
Tabel 4.15 menampilkan hasil bobot dan bias akhir dari perhitungan untuk *input layer* ke *hidden layer*. Sementara itu, Tabel 4.16 menunjukkan hasil bobot dan bias akhir untuk *hidden layer* ke *output layer* dengan hanya 1 kali iterasi. Tahapan iterasi

terus dilakukan hingga mencapai eror minimum. Gambar 4.4 hingga Gambar 4.6 merupakan gambaran plot arsitektur model *neural network* yang secara otomatis didapatkan melalui *software* pada setiap komponen, gambar tersebut menunjukkan hasil nilai bobot dan bias akhir pada masing-masing *neuron* yang telah optimal. Nilai bobot dituliskan dengan warna hitam sedangkan yang berwarna biru merupakan nilai bias pada masing-masing *neuron*.



**Gambar 4.4** Plot Arsitektur *Neural Network* Komponen *Trend*

Plot arsitektur ini memberikan gambaran tentang struktur jaringan, termasuk jumlah lapisan yang ada dalam jaringan, jumlah *neuron* yang terdapat pada setiap lapisan, serta pola koneksi antara *neuron-neuron* tersebut. Selain itu, plot juga mencerminkan bobot dan bias yang telah dioptimalkan selama proses pelatihan jaringan. Gambar 4.4 merupakan hasil dari metode NN dengan arsitektur terbaik untuk komponen *trend* yaitu 6-4-1 (6 *neuron* input – 4 *neuron* hidden – 1 *neuron* output). Bobot dan bias pada plot komponen *trend* merupakan parameter-parameter yang telah di optimalkan selama proses pelatihan jaringan dengan melakukan 155 kali iterasi.



**Gambar 4.5** Plot Arsitektur *Neural Network* Komponen Musiman

Gambar 4.5 merupakan hasil dari metode NN dengan arsitektur terbaik untuk komponen musiman yaitu 6-3-1 (6 *neuron input* – 3 *neuron hidden* – 1 *neuron output*). Bobot dan bias pada plot komponrn musiman ini telah di optimalkan selama proses pelatihan jaringan dengan melakukan 240 kali iterasi.



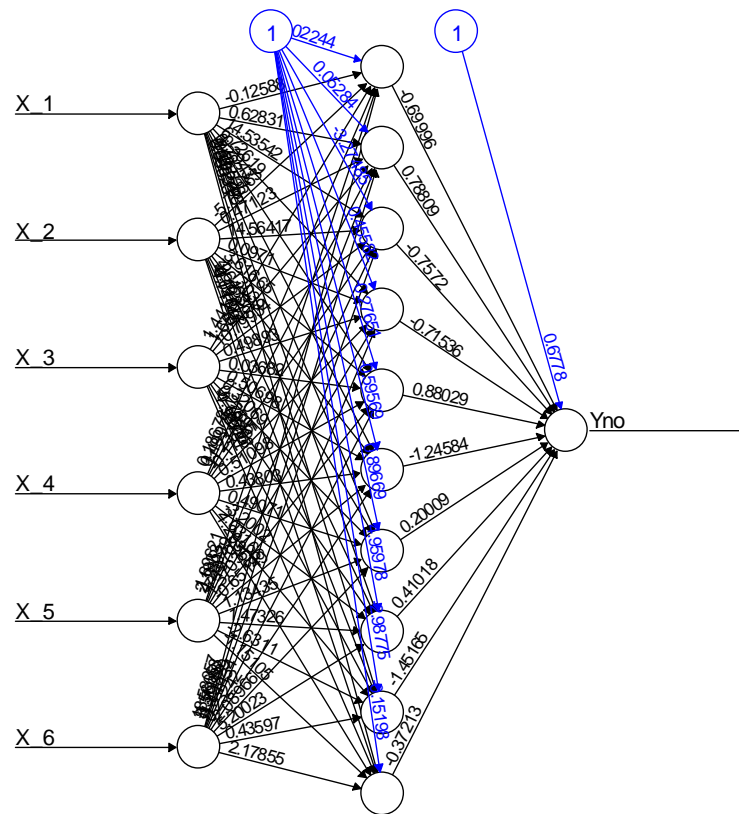


Figure 4.6: Noise Component Architecture

**Gambar 4.6** Arsitektur *Neural Network* Komponen *Noise*

Gambar 4.6 merupakan hasil dari metode NN dengan arsitektur terbaik untuk komponen *noise* yaitu 6-10-1 (6 *neuron input* – 10 *neuron hidden* – 1 *neuron output*). Bobot dan bias pada plot komponen *noise* telah di optimalkan selama proses pelatihan jaringan dengan melakukan 3611 kali iterasi.

### 4.3 Prediksi Curah Hujan Kabupaten Pangkajene dan Kepulauan

Dalam proses pelatihan digunakan data sebanyak 48, yaitu periode Januari 2018 hingga Desember 2021. Sementara itu, pada tahap pengujian digunakan data sebanyak 12 yaitu periode Januari 2022 hingga Desember 2022. Setelah proses pelatihan dilakukan, diperoleh arsitektur terbaik pada metode *neural network* dan selanjutnya tahap pengujian dilakukan prediksi menggunakan data uji yang kemudian akurasi hasil prediksi dihitung menggunakan MAPE. Hasil peramalan dan akurasi peramalan dapat dilihat pada Tabel 4.17 dan terlampir pada Lampiran 6.

**Tabel 4.17** Hasil Prediksi dan Akurasi Hasil Prediksi

Tahun	Bulan	Data Aktual	Data Prediksi	<i>Percentage of Error</i>
2022	Januari	624.5658	591.9101	5.2285
	Februari	473.7574	551.8779	16.4895
	Maret	370.7063	442.0837	19.2544
	April	423.8105	387.5624	8.5529
	Mei	432.7966	394.5393	8.8395
	Juni	380.3555	409.0994	7.5571
	Juli	361.0110	379.7306	5.1853
	Agustus	454.2319	433.231	4.6233
	September	633.3421	616.1032	2.7218
	Oktober	638.9280	589.4911	7.7374
	November	733.8548	730.4077	0.4697
	Desember	624.5658	626.0324	0.2348
MAPE				7.2412

Tabel 4.17 menunjukkan eror pada setiap nilai prediksi dengan data aktualnya. Hasil akurasi menunjukkan bahwa terdapat eror yang cukup besar pada beberapa hasil prediksi namun juga terdapat eror yang cukup kecil antara data prediksi dan data aktual. Selain itu dilakukan juga perhitungan akurasi antara metode tunggal menggunakan MAPE dan korelasi sebagai kriteria kebaikan model. Hasil perbandingan akurasi prediksi disajikan pada Tabel 4.18 dan terlampir pada Lampiran 7.

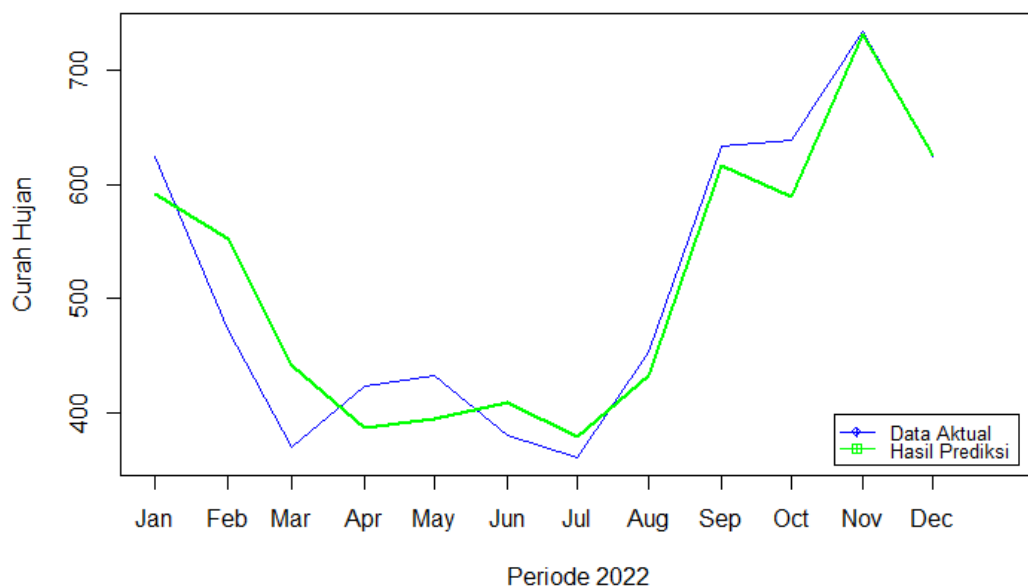
**Tabel 4.18** Perbandingan MAPE dan Korelasi

Akurasi	MAPE	Korelasi
SSA	19.19%	0.98
NN	82.19%	0.81
SSA-NN	7.24%	0.94

Tabel 4.18 menunjukkan hasil akurasi antara data aktual dan data hasil prediksi SSA, NN, dan SSA-NN. Hasil analisis menunjukkan bahwa pendekatan *hybrid singular spectrum analysis* dan *neural network* (SSA-NN) menghasilkan akurasi peramalan yang

lebih baik daripada metode tunggal, baik SSA maupun NN. Ketika menggunakan metode SSA, diperoleh MAPE sebesar 19.19%, yang menandakan tingkat kesalahan rata-rata dalam meramalkan curah hujan sebesar 19%. Sementara itu, metode NN menghasilkan MAPE sebesar 82.19%, menunjukkan tingkat kesalahan yang lebih tinggi dalam peramalan curah hujan. Namun, ketika menggabungkan metode SSA dan NN dalam pendekatan *hybrid* SSA-NN, didapatkan MAPE sebesar 16.69%.

Adapun grafik hasil perbandingan prediksi metode *hybrid* SSA-NN dengan data aktual untuk data curah hujan Kabupaten Pangkajene dan Kepulauan yang disajikan pada Gambar 4.7.



**Gambar 4.7** Perbandingan Data Aktual dengan Hasil Prediksi SSA-NN

Gambar 4.7 menunjukkan bahwa hasil prediksi dengan menggunakan metode *hybrid singular spectrum analysis* dan *neural network* pada data uji cenderung hampir mengikuti pola data aktual dengan baik. Koefisien korelasi yang diperoleh dari data aktual dan hasil prediksi ini sebesar 0.946. Dengan tingkat korelasi yang mendekati 1 menunjukkan bahwa korelasi yang kuat antara data aktual dan hasil prediksi. Hasil perbandingan menunjukkan peningkatan dalam akurasi peramalan, mengindikasikan bahwa pendekatan *hybrid* mampu memberikan hasil yang lebih akurat dalam meramalkan curah hujan. Dengan demikian, pendekatan *hybrid* SSA-NN dapat dianggap sebagai pilihan yang optimal untuk meramalkan curah hujan dengan tingkat kesalahan yang lebih rendah, sehingga dapat memberikan informasi yang baik dan akurat untuk pengambilan keputusan terkait cuaca dan lingkungan.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Peramalan data curah hujan Kabupaten Pangkajene dan Kepulauan pada pos hujan Bungoro menghasilkan kesimpulan yaitu model *hybrid singular spectrum analysis* dan *neural network* terbaik pada *window length 23* dengan arsitektur model *neural network* 6-4-1 (6 *neuron input* – 4 *neuron hidden* – 1 *neuron output*) untuk komponen *trend*, 6-3-1 (6 *neuron input* – 3 *neuron hidden* – 1 *neuron output*) untuk komponen musiman, serta 6-10-1 (6 *neuron input* – 10 *neuron hidden* – 1 *neuron output*) untuk komponen *noise*. Tingkat akurasi peramalan pada metode *hybrid singular spectrum analysis* dan *neural network* (SSA-NN) mencapai nilai MAPE sebesar 7.24% yang menunjukkan bahwa tingkat kesalahan rata-rata peramalan sebesar 7.24%. Dan dengan tingkat korelasi sebesar 0.946 yang berarti mendekati 1, hal ini menunjukkan korelasi yang kuat pada data aktual dan hasil prediksi.

#### 5.2 Saran

Dari kesimpulan yang diambil, berikut saran untuk penelitian selanjutnya

1. Algoritma pada *neural network* yang juga dapat digunakan yaitu *reccurent*, algoritma ini dapat dijadikan pembanding dengan kinerja algoritma *backpropagation*.
2. Metode *machine learning* lainnya juga dapat dikombinasikan dengan *singular spectrum analysis* seperti *long short-term memory* atau *convolutional neural network*.

## DAFTAR PUSTAKA

- Abduh, M. (2023). *Banjir Rendam 7 Kecamatan di Pangkep, 80 Orang Mengungsi*. detikSulsel.
- Abhishek, K., Khairwa, A., Pratap, T., & Prakash, S. (2012). A Stock Market Prediction Model Using Artificial Neural Network. *2012 Third International Conference on Computing, Communication and Networking Technologies (ICCCNT'12)*, 1–5.
- Achmalia, A. F. (2019). Peramalan Penjualan Semen Menggunakan Backpropagation Neural Network dan Recurrent Neural Network (Studi kasus di PT Semen Indonesia (Persero) Tbk). *Skripsi*. Universitas Negeri Semarang: Semarang.
- Adiputra, R., Herdiani, E. T., & Sahrinan, S. (2021). Peramalan Jumlah Penumpang Kapal Laut Menggunakan Metode Fuzzy Runtun Waktu Chen Orde Tinggi. *ESTIMASI: Journal of Statistics and Its Application*, 2(1), 38–48.
- Aini, N. N. (2017). Peramalan Curah Hujan dengan Menggunakan GSTAR-Backpropagation Neural Network. *Skripsi*. Universitas Brawijaya: Malang.
- Amalina, N. (2016). Penerapan Metode Artificial Neural Network untuk Meramalkan Nilai Ekspor Migas dan Non Migas di Indonesia. *Skripsi*. Institut Teknologi Sepuluh Nopember: Surabaya.
- Asrof, A., Ischak, R., & Darmawan, G. (2017). Peramalan Produksi Cabai Merah di Jawa Barat Menggunakan Metode Singular Spectrum Analysis (SSA). *Statistika*, 17(2), 77–87.
- Badrul, M. (2016). Optimasi Neural Network dengan Algoritma Genetika untuk Prediksi Hasil Pemilukada. *Bina Insani ICT Journal*, 3(1), 229–242.
- Chrisantama, A. R., Sulandari, W., & Sugiyanto, S. (2021). Penerapan Metode Auto Singular Spectrum Analysis pada Peramalan Data Indeks Harga Saham Gabungan di Indonesia. *Pattimura Proceeding: Conference of Science and Technology*, 405–410.
- Christienova, S. I., Pratiwi, E. W., & Darmawan, G. (2018). Perbandingan Model Peramalan Singular Spectrum Analysis (SSA) dan Fourier Series Analysis (FSA) pada Data Suhu Udara di Surabaya. *BIMIPA*, 25(1), 94–106.
- Desvina, A. P., & Ratnawati, R. (2014). Penerapan Model Vector Autoregressive (VAR) untuk Peramalan Curah Hujan Kota Pekanbaru. *SITEKIN: Jurnal Sains, Teknologi dan Industri*, 11(2), 151–159.

- Diwanda, A. A., Setiawan, I. N., & Setiawan, W. (2021). Peramalan Kebutuhan Energi Listrik Jangka Panjang di Provinsi Bali Rentang Tahun 2020-2030 Menggunakan Neural Network. *Jurnal SPEKTRUM*, 8(2), 99–109.
- Eksiandayani, S. (2016). Pemodelan Peramalan Inflasi Umum dan Inflasi Menurut Kelompok Pengeluaran di Indonesia dengan Metode Hibrida Arimax-NN. *Tesis*. Institut Teknologi Sepuluh Nopember: Surabaya.
- Faulina, R. (2014). Perbandingan Akurasi Ensemble ARIMA dalam Peramalan Curah Hujan di Kota Batu, Malang, Jawa Timur. *Jurnal Matematika, Sains, dan Teknologi*, 15(2), 75–83.
- Faulina, R., & Aini, N. (2018). Metode Hibrida Berbasis Preprocessing Data untuk Peramalan Curah Hujan Jangka Panjang. *Jurnal Ilmiah Soulmath: Jurnal Edukasi Pendidikan Matematika*, 6(2), 105–112.
- Fausett, L. (1994). *Fundamentals Of Neural Network Architectures, Algorithms, and Applications*. Prentice-Hall.
- Fauziah, N., Wahyuningsih, S., & Nasution, Y. N. (2016). Peramalan Menggunakan Fuzzy Time Series Chen (Studi Kasus: Curah Hujan Kota Samarinda). *Jurnal Statistika Universitas Muhammadiyah Semarang*, 4(2), 52–61.
- Fitriyanti. (2022). Aplikasi Jaringan Syaraf Tiruan Backpropagation dalam Prediksi Curah Hujan Bulanan di Kabupaten Wajo Sulawesi Selatan. *Jurnal Pendidikan Fisika*, 11(1), 44–55.
- Golyandina, N., Nekrutkin, V., & Zhigljavsky, A. (2001). *Analysis of time series structure: SSA and related techniques*. CRC press.
- Habinuddin, E., Binarto, A., & Sartika, E. (2019). Peramalan Curah Hujan Kota Bandung dengan Menggunakan Metode Analisis Spektral. *Sigma-Mu*, 11(1), 1–12.
- Idrus, R. A., Ruliana, & Aswi. (2022). Penerapan Metode Singular Spectrum Analysis dalam Peramalan Jumlah Produksi Beras di Kabupaten Gowa. *VARIANSI: Journal of Statistics and Its Application on Teaching and Research*, 4(2), 49–58.
- Isnawati, S. (2018). Model Hibrida Singular Spectrum Analysis dan Automatic Arima untuk Peramalan Air Terjual di PDAM Giri Tirta Sari Kabupaten Wonogiri Jawa Tengah. *Skripsi*. Institut Teknologi Sepuluh Nopember: Surabaya.
- Kusumadewi, F. (2014). Peramalan Harga Emas Menggunakan Feedforward Neural Network dengan Algoritma Backpropagation. *Skripsi*. Universitas Negeri Yogyakarta: Yogyakarta.

- Ma'rifah, N., Rahayu, S. P., & Suhartono, S. (2013). Peramalan Pendapatan Operasional Bank Menggunakan Metode Fungsi Transfer dan Neural Network. *Jurnal Sains dan Seni ITS*, 2(2), 219–224.
- Nielsen, M. (2015). *Neural Networks and Deep Learning*. Determination Press.
- Ningsih, S. S. (2021). Peramalan Harga Beras di Tingkat Perdagangan Besar (Grosir) di Indonesia Menggunakan Metode Singular Spectrum Analysis (SSA). *Skripsi*. Universitas Muhammadiyah Semarang: Semarang.
- PangkepKab. *Selayang Pandang*. URL: <https://pangkepKab.go.id/selayang-pandang>
- Purnama, E. (2022). Aplikasi Metode Singular Spectrum Analysis (SSA) pada Peramalan Curah Hujan di Provinsi Gorontalo. *Jambura Journal of Probability and Atatistics*, 3(2), 161–170.
- Rahayu, D., Wihandika, R. C., & Perdana, R. S. (2018). Implementasi Metode Backpropagation untuk Klasifikasi Kenaikan Harga Minyak Kelapa Sawit. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(4), 1547–1552.
- Ryandhi, R. (2017). Penerapan Metode Artificial Neural Network (ANN) untuk Peramalan Inflasi di Indonesia. *Skripsi*. Institut Teknologi Sepuluh Nopember: Surabaya.
- Sakinah, A. M. (2018). Akurasi Peramalan Long Horizon dengan Singular Spectrum Analysis. *KUBIK: Jurnal Publikasi Ilmiah Matematika*, 3(2), 93–99.
- Saraswati, E., Umaidah, Y., & Voutama, A. (2021). Penerapan Algoritma Artificial Neural Network untuk Klasifikasi Opini Publik Terhadap Covid-19. *Generation Journal*, 5(2), 109–118.
- Satriani, S., Nursalam, & Ibtnas, R. (2020). Peramalan Indeks Harga Konsumen (IHK) di Sulawesi Selatan dengan Menggunakan Metode Singular Spektrum Analysis (SSA). *Jurnal MSA (Matematika dan Statistika serta Aplikasinya)*, 8(1), 82–89.
- Setyowati, E., Salehah, N. A., Lee, M. H., Rahayu, S. P., & Ulama, B. S. S. (2019). A Hybrid Singular Spectrum Analysis and Neural Networks for Forecasting Inflow and Outflow Currency of Bank Indonesia. *Soft Computing in Data Science: 4th International Conference*, 3–18.
- Shafira, D. A., Utami, T. W., & Arum, P. R. (2020). Peramalan Wisatawan Mancanegara Berkunjung ke Bali Menggunakan Singular Spectrum Analysis

- (SSA). Dalam D. A. Shafira (Ed.), *Prosiding Seminar Nasional Unimus* (Vol. 3). Universitas Muhammadiyah Semarang.
- Sholikhah, I. U. (2021). Penerapan Artificial Neural Network untuk Memprediksi Indeks Harga Saham Jakarta Islamic Index. *Skripsi*. Universitas Islam Negeri Maulana Malik Ibrahim: Malang.
- Sinaga, H. D. E., & Irawati, N. (2018). Perbandingan Double Moving Average dengan Double Exponential Smoothing pada Peramalan Bahan Medis Habis Pakai. *JURTEKSI (Jurnal Teknologi dan Sistem Informasi)*, 4(2), 197–204.
- Siringoringo, M., Wahyuningsih, S., Purnamasari, I., & Arumsari, M. (2022). Peramalan Jumlah Produksi Kelapa Sawit Provinsi Kalimantan Timur Menggunakan Metode Singular Spectrum Analysis. *VARIANSI: Journal of Statistics and Its application on Teaching and Research*, 4(3), 162–172.
- Sitohang, Y. O., & Darmawan, G. (2018). Perbandingan Akurasi Recurrent Forecasting dan Vector Forecasting pada Metode Singular Spectrum Analysis dalam Peramalan Jumlah Wisatawan Mancanegara yang Masuk Melalui Bandara Ngurah Rai Bali Tahun 2017. *Euclid*, 5(1), 125–133.
- Sonata, M. E. (2021). Peramalan Produksi Padi di Provinsi Jawa Timur Menggunakan Metode Feedforward Neural Network. *Skripsi*. Universitas Muhammadiyah Semarang: Semarang.
- Supriyati, S., Tjahjono, B., & Effendy, S. (2018). Analisis Pola Hujan untuk Mitigasi Aliran Lahar Hujan Gunungapi Sinabung. *Jurnal Ilmu Tanah dan Lingkungan*, 20(2), 95–100.
- Suryani, A. R., Sugiman, & Hendikawati, P. (2018). Peramalan Curah Hujan dengan Metode Autoregressive Integrated Moving Average with Exogenous Input (ARIMAX). *UNNES Journal of Mathematics*, 7(1), 120–129.
- Suryani, I., & Wahono, R. S. (2015). Penerapan Exponential Smoothing untuk Transformasi Data dalam Meningkatkan Akurasi Neural Network pada Prediksi Harga Emas. *Journal of Intelligent Systems*, 1(2), 67–75.
- Swarinoto, Y. S., & Husain, H. (2012). Estimasi Curah Hujan Harian dengan Metode Auto Estimator (Kasus Jayapura dan sekitarnya). *Jurnal Meteorologi dan Geofisika*, 13(1), 53–61.
- Syaifullah, M. D. (2014). Validasi Data TRMM Terhadap Data Curah Hujan Aktual di Tiga DAS di Indonesia. *Jurnal Meteorologi dan Geofisika*, 15(2), 109–118.
- Syamsiah, N. O., & Purwandani, I. (2019). Penerapan Neural Network untuk Peramalan Data Time Series Univariate Jumlah Wisatawan Mancanegara. *Jurnal Mantik Penusa*, 3(3).



- Tjasyono, B. (2009). *Meteorologi Indonesia I: Karakteristik dan Sirkulasi Atmosfer*. Badan Meteorologi Klimatologi dan Geofisika.
- Utami, H., Sari, Y. W., Subanar, S., Abdurakhman, A., & Gunardi, G. (2019). Peramalan Beban Listrik Daerah Istimewa Yogyakarta dengan Metode Singular Spectrum Analysis (SSA). *Media Statistika*, 12(2), 214–225.
- Utami, N. A. G., Sulandari, W., & Handajani, S. S. (2021). Peramalan Curah Hujan Bulanan di Pos Hujan Jatisrono dengan Metode Singular Spectrum Analysis (SSA). *Prosiding Seminar Nasional Aplikasi Sains dan Teknologi*, 84–93.
- Wicaksono, A., Helmi, & Yunandari. (2019). Prediksi Outflow Uang Kartal di Kalimantan Barat dengan Metode Singular Spectrum Analysis (SSA). *Bimaster: Buletin Ilmiah Matematika, Statistika dan Terapannya*, 8(3), 545–554.
- Windarto, Y.E. (2020). Analisis Penyakit Kardiovaskular menggunakan Metode Korelasi Pearson, Sperman dan Kendall. *Jurnal Saintekom*, 10(2), 119-127.

# LAMPIRAN

**Lampiran 1.** Data Curah Hujan Bulanan Kabupaten Pangkajene dan Kepulauan pada Pos Hujan Bungoro Periode 2018-2022.

Tahun	Bulan	Curah Hujan	Tahun	Bulan	Curah Hujan
2018	Januari	779	2021	Januari	810
2018	Februari	617	2021	Februari	436
2018	Maret	502	2021	Maret	586
2018	April	267	2021	April	253
2018	Mei	71	2021	Mei	108
2018	Juni	97	2021	Juni	132
2018	Juli	55	2021	Juli	73
2018	Agustus	2	2021	Agustus	68
2018	September	1	2021	September	71
2018	Oktober	2	2021	Oktober	366
2018	November	351	2021	November	301
2018	Desember	984	2021	Desember	704
2019	Januari	581	2022	Januari	767
2019	Februari	265	2022	Februari	633
2019	Maret	343	2022	Maret	317
2019	April	228	2022	April	78
2019	Mei	43	2022	Mei	221
2019	Juni	86	2022	Juni	221
2019	Juli	1	2022	Juli	98
2019	Agustus	3	2022	Agustus	35
2019	September	0	2022	September	180
2019	Oktober	1	2022	Oktober	561
2019	November	20	2022	November	479
2019	Desember	302	2022	Desember	697
2020	Januari	536			
2020	Februari	281			
2020	Maret	294			
2020	April	79			
2020	Mei	27			
2020	Juni	8			
2020	Juli	41			
2020	Agustus	75			
2020	September	19			
2020	Oktober	163			
2020	November	224			
2020	Desember	870			

Lampiran 2. Output Hasil Rekonstruksi menggunakan *software R Studio*

```

> MAPE #L=16
[1] 30.34192
> DA[49:60]
[1] 753.47718 623.56761 347.01359 152.32102 212.01228 158.33176 104
.80986
[8] 90.66758 174.33639 363.58183 560.96509 712.52426

> MAPE #L=17
[1] 19.61249
> DA[49:60]
[1] 761.01630 632.52601 354.13076 139.35551 221.02632 170.87758 96
.33386
[8] 59.10023 178.28202 406.67426 555.22676 651.73761

> MAPE #L=18
[1] 27.63155
> DA[49:60]
[1] 754.79047 595.06594 387.29009 190.34543 163.79057 175.02534 105
.06800
[8] 54.24227 192.28793 403.55741 514.69763 651.24732

> MAPE #L=19
[1] 40.99694
> DA[49:60]
[1] 646.39261 624.84043 445.44152 250.64980 197.00564 148.72606 30
.55192
[8] 39.98675 254.73885 456.67574 539.25733 601.12298

> MAPE #L=20
[1] 19.91408
> DA[49:60]
[1] 779.60299 646.46392 356.61942 161.98216 186.32614 216.83513 89
.33231
[8] 23.03628 198.74861 432.91930 543.31626 642.93690

> MAPE #L=21
[1] 32.50589
> DA[49:60]
[1] 720.57215 650.85778 351.03189 194.38279 182.18063 163.13027 112
.08595
[8] 71.33362 175.66549 351.45968 555.14937 722.51498

> MAPE #L=22
[1] 22.65149
> DA[49:60]
[1] 775.5114 670.9458 325.9654 177.0966 191.4829 197.5458 100.9559
14.1551
[9] 186.2622 407.0261 543.5871 669.9046

> MAPE #L=23
[1] 19.19156
> DA[49:60]
[1] 772.13247 639.43123 332.77280 152.31918 191.02593 207.21677 82
.28889
[8] 23.27363 207.12536 458.01324 571.44210 653.12181

> MAPE #L=24
[1] 32.68016
> DA[49:60]
[1] 780.3286 598.1483 439.7171 172.0697 193.9668 108.9067 146.1923
58.3291
[9] 144.2305 492.9986 447.2455 754.1391

```

Lampiran 3. Output Hasil Rekonstruksi menggunakan *software R Studio*

```

> outputSSA=cbind(Trend,Musiman,Noise)
> outputSSA
      Trend      Musiman      Noise
[1,] 234.8787 424.7477624 119.373536
[2,] 225.7006 302.3825136  88.916869
[3,] 217.9253 195.5848842  88.489843
[4,] 208.0226  61.0366107  -2.059232
[5,] 200.4338 -72.2659645 -57.167843
[6,] 194.2959 -122.9535570  25.657619
[7,] 188.9375 -68.9282735 -65.009233
[8,] 184.6744 -63.2902351 -119.384213
[9,] 183.0646 -217.5999511  35.535348
[10,] 183.4264 -202.9705748  21.544222
[11,] 186.0919  220.1455091 -55.237451
[12,] 188.9372  580.2039941 214.858827
[13,] 188.5887  438.8110197 -46.399740
[14,] 189.0927  173.4105666 -97.503265
[15,] 190.6472  112.5939780  39.758797
[16,] 191.6646  43.7870583  -7.451624
[17,] 192.9697  -93.8312368 -56.138438
[18,] 194.1790 -174.1330253  65.954043
[19,] 194.9369 -203.9183341   9.981392
[20,] 196.0589 -216.4846937  23.425754
[21,] 197.9520 -239.3505699  41.398538
[22,] 200.4794 -240.8080429  41.328599
[23,] 203.8011  -93.5481162 -90.253030
[24,] 204.4625  176.0692047 -78.531687
[25,] 206.3680  288.4675179  41.164453
[26,] 208.9258  159.1451377 -87.070972
[27,] 212.9632  -3.0863669  84.123163
[28,] 217.5249 -123.4619068 -15.062978
[29,] 222.3209 -203.0177084   7.696850
[30,] 227.0925 -227.1080273   8.015537
[31,] 232.3472 -196.9649768   5.617793
[32,] 237.8257 -161.6500396  -1.175617
[33,] 243.0206 -180.0687947 -43.951803
[34,] 248.2327 -147.2625301  62.029849
[35,] 254.0310  112.8020879 -142.833107
[36,] 262.4868  456.3123998  151.200757
[37,] 269.5182  493.8490746  46.632721
[38,] 275.1276  296.5594751 -135.687105
[39,] 278.3935  151.2517506  156.354702
[40,] 281.2426  16.7970318  -45.039661
[41,] 284.8994 -132.5004504 -44.398979
[42,] 289.9744 -193.9551061  35.980660
[43,] 295.8656 -198.4991551 -24.366455
[44,] 302.5761 -220.1953059 -14.380789
[45,] 309.9889 -187.9604080 -51.028459
[46,] 318.0247  -72.1135505  120.088818
[47,] 324.3021  82.3291548 -105.631235
[48,] 330.2484  302.7798023  70.971820
[49,] 333.6073  438.5252056  -5.132469
[50,] 333.7224  305.7087791  -6.431226
[51,] 333.3589  -0.5861152 -15.772800
[52,] 333.8503 -181.5311597 -74.319184
[53,] 337.0939 -146.0679927  29.974074
[54,] 341.3671 -134.1503553  13.783230
[55,] 346.5809 -264.2920339  15.711109
[56,] 354.0356 -330.7619345  11.726366
[57,] 365.6452 -158.5198017 -27.125362
[58,] 380.8837  77.1295851  102.986758
[59,] 386.1768  185.2652627 -92.442095
[60,] 398.1181  255.0036625  43.878192

```

**Lampiran 4. Output Bobot dan Bias Awal dengan Nilai Random**

```
> neuralnetworkTrendLAGI$startweights
[[1]]
[[1]][[1]]
      [,1]      [,2]      [,3]      [,4]
[1,] -0.6264538  0.7383247  1.12493092  0.78213630
[2,]  0.1836433  0.5757814 -0.04493361  0.07456498
[3,] -0.8356286 -0.3053884 -0.01619026 -1.98935170
[4,]  1.5952808  1.5117812  0.94383621  0.61982575
[5,]  0.3295078  0.3898432  0.82122120 -0.05612874
[6,] -0.8204684 -0.6212406  0.59390132 -0.15579551
[7,]  0.4874291 -2.2146999  0.91897737 -1.47075238

[[1]][[2]]
      [,1]
[1,] -0.4781501
[2,]  0.4179416
[3,]  1.3586796
[4,] -0.1027877
[5,]  0.3876716
```

**Lampiran 5.** Data Hasil Normalisasi untuk *Input Neural Network*.

Y	X1	X2	X3	X4	X5	X6
0.198258	0.240936	0.240936	0.240936	0.240936	0.240936	0.240936
0.162102	0.198258	0.198258	0.198258	0.198258	0.198258	0.198258
0.116055	0.162102	0.162102	0.162102	0.162102	0.162102	0.162102
0.080767	0.116055	0.116055	0.116055	0.116055	0.116055	0.116055
0.052226	0.080767	0.080767	0.080767	0.080767	0.080767	0.080767
0.027309	0.052226	0.052226	0.052226	0.052226	0.052226	0.052226
0.007486	0.027309	0.027309	0.027309	0.027309	0.027309	0.027309
0	0.007486	0.007486	0.007486	0.007486	0.007486	0.007486
0.001682	0	0	0	0	0	0
0.014077	0.001682	0.001682	0.001682	0.001682	0.001682	0.001682
0.027308	0.014077	0.014077	0.014077	0.014077	0.014077	0.014077
0.025687	0.027308	0.027308	0.027308	0.027308	0.027308	0.027308
0.028031	0.025687	0.025687	0.025687	0.025687	0.025687	0.025687
0.035259	0.028031	0.028031	0.028031	0.028031	0.028031	0.028031
0.03999	0.035259	0.035259	0.035259	0.035259	0.035259	0.035259
0.046059	0.03999	0.03999	0.03999	0.03999	0.03999	0.03999
0.051682	0.046059	0.046059	0.046059	0.046059	0.046059	0.046059
0.055206	0.051682	0.051682	0.051682	0.051682	0.051682	0.051682
0.060424	0.055206	0.055206	0.055206	0.055206	0.055206	0.055206
0.069227	0.060424	0.060424	0.060424	0.060424	0.060424	0.060424
0.080979	0.069227	0.069227	0.069227	0.069227	0.069227	0.069227
0.096425	0.080979	0.080979	0.080979	0.080979	0.080979	0.080979
0.0995	0.096425	0.096425	0.096425	0.096425	0.096425	0.096425
0.108361	0.0995	0.0995	0.0995	0.0995	0.0995	0.0995
0.120255	0.108361	0.108361	0.108361	0.108361	0.108361	0.108361
0.139029	0.120255	0.120255	0.120255	0.120255	0.120255	0.120255
0.16024	0.139029	0.139029	0.139029	0.139029	0.139029	0.139029
0.182542	0.16024	0.16024	0.16024	0.16024	0.16024	0.16024

**Lampiran 5.** Data Hasil Normalisasi untuk *Input Neural Network* (Lanjutan).

Y	X1	X2	X3	X4	X5	X6
0.20473	0.182542	0.182542	0.182542	0.182542	0.182542	0.182542
0.229164	0.20473	0.20473	0.20473	0.20473	0.20473	0.20473
0.254639	0.229164	0.229164	0.229164	0.229164	0.229164	0.229164
0.278796	0.254639	0.254639	0.254639	0.254639	0.254639	0.254639
0.303032	0.278796	0.278796	0.278796	0.278796	0.278796	0.278796
0.329994	0.303032	0.303032	0.303032	0.303032	0.303032	0.303032
0.369314	0.329994	0.329994	0.329994	0.329994	0.329994	0.329994
0.40201	0.369314	0.369314	0.369314	0.369314	0.369314	0.369314
0.428094	0.40201	0.40201	0.40201	0.40201	0.40201	0.40201
0.44328	0.428094	0.428094	0.428094	0.428094	0.428094	0.428094
0.456528	0.44328	0.44328	0.44328	0.44328	0.44328	0.44328
0.473532	0.456528	0.456528	0.456528	0.456528	0.456528	0.456528
0.497131	0.473532	0.473532	0.473532	0.473532	0.473532	0.473532
0.524525	0.497131	0.497131	0.497131	0.497131	0.497131	0.497131
0.555729	0.524525	0.524525	0.524525	0.524525	0.524525	0.524525
0.590198	0.555729	0.555729	0.555729	0.555729	0.555729	0.240936
0.627565	0.590198	0.590198	0.590198	0.590198	0.240936	0.198258
0.656755	0.627565	0.627565	0.627565	0.240936	0.198258	0.162102
0.684405	0.656755	0.656755	0.240936	0.198258	0.162102	0.116055
0.198258	0.684405	0.240936	0.198258	0.162102	0.116055	0.080767



**Lampiran 6.** Hasil Prediksi Metode *Hybrid SSA-NN* pada Data Uji

```

> #Unstandardized Data Aktual
> t=(datatren1*maxYt-minYt)+minYt
> s=(datamusim1*maxYm-minYm)+minYm
> n=(datanoise1*maxYn-minYn)+minYn
> #Menggabungkan data aktual
> YAuji=t+s+n
> YAuji
[1] 624.5658 473.7574 370.7063 423.8105 432.7966 380.3555 361.0110
454.2319 633.3421
[10] 638.9280 733.8548 624.5658
> #Menggabungkan Hasil Prediksi
> ptotal=prediktren+predikmusim+prediknoise
> ptotal
Time Series:
Start = 1
End = 12
Frequency = 1
[1] 591.9101 551.8779 442.0837 387.5624 394.5393 409.0994 379.7306
433.2311 616.1032
[10] 589.4911 730.4077 626.0324

```

Lampiran 7. Output Perbandingan menggunakan MAPE dan Korelasi *Pearson*

```

> #MAPE SSA
> residu=datacurah[49:60]-DA[49:60]
> PEI=(residu/datacurah[49:60])*100
> MAPE=(sum(abs(PEI)))/(length(datacurah[49:60]))
> MAPE
[1] 19.19156
> #Korelasi
> cor.test(datacurah[49:60],DA[49:60], method = "pearson")

Pearson's product-moment correlation

data: datacurah[49:60] and DA[49:60]
t = 15.573, df = 10, p-value = 2.437e-08
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.9280643 0.9945458
sample estimates:
      cor
0.9800004

> #MAPE NN
> residu=dataLAGI[49:60]-predik.NNLAGI[49:60]
> PEI=(residu/dataLAGI[49:60])*100
> MAPE=(sum(abs(PEI)))/(length(dataLAGI[49:60]))
> MAPE
[1] 82.1989
> #Korelasi
> cor.test(dataLAGI[49:60],predik.NNLAGI[49:60], method = "pearson")

Pearson's product-moment correlation

data: dataLAGI[49:60] and predik.NNLAGI[49:60]
t = 4.453, df = 10, p-value = 0.001229
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.4537408 0.9463947
sample estimates:
      cor
0.8153287

> #MAPE KESELURUHAN
> residu=YAuji-ptotal
> PEI=(residu/YAuji)*100
> MAPE=(sum(abs(PEI)))/(length(YAuji))
> MAPE
[1] 7.241227
> #Korelasi
> cor.test(YAuji,ptotal, method = "pearson")

Pearson's product-moment correlation

data: YAuji and ptotal
t = 9.2686, df = 10, p-value = 3.174e-06
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.8154465 0.9852086
sample estimates:
      cor
0.9464314

```