

**PREDIKSI KEBUTUHAN ENERGI LISTRIK DI JAWA
TENGAH TAHUN 2023-2028 MENGGUNAKAN METODE
ADALINE DAN METODE BACKPROPAGATION-FUZZY
CLUSTERING**

SKRIPSI

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



**Oleh : ALVIAN RIZKY FAJAR ANGGARA
NIM : 2008046030**

**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI WALISONGO
SEMARANG
2025**

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini :

Nama : Alvian Rizky Fajar Anggara
NIM : 2008046030
Jurusan/Program Studi : Matematika/ Matematika

menyatakan bahwa skripsi yang berjudul :

**PREDIKSI KEBUTUHAN ENERGI LISTRIK DI JAWA TENGAH
TAHUN 2023-2028 MENGGUNAKAN METODE ADALINE DAN
METODE BACKPROPAGATION-FUZZY CLUSTERING**

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

Semarang, 14 Febuari 2025

Pembuat pernyataan,



Alvian Rizky Fajar Anggara
NIM : 2008046030



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

PENGESAHAN

Naskah skripsi berikut ini :

Judul : **Prediksi Kebutuhan Energi Listrik di Jawa Tengah tahun 2023-2028 menggunakan Metode Adaline dan Metode Backpropagation-Fuzzy Clustering**

Penulis : Alvian Rizky Fajar Anggara

NIM : 2008046030

Jurusan : Matematika

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

Semarang, 26 Mei 2025

DEWAN PENGUJI

Penguji I,

Any

Any Muanalifah, M.Si., Ph.D

NIP : 198201132011012009

Penguji II,

Pradi Kurniawan

Pradi Kurniawan, M.Sc.

NIP : 199012261019031012

Penguji III,

Seftina

Seftina Diyah Miasary, M.Sc.

NIP : 198709212019032016

Penguji IV,

Agus Wayan Yulianto

Agus Wayan Yulianto, M.Sc.

NIP : 198907162019031007

Pembimbing,

Yolanda

Yolanda Norasia, M.Si.

NIP : 199409232019031010

NOTA DINAS

Semarang, 14 Febuari 2025

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

Assalamu'alaikum warahmatullahi wabarakatuh

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

Judul : Prediksi Kebutuhan Energi Listrik di Jawa Tengah
tahun 2023-2028 menggunakan Metode Adaline dan
Metode Backpropagation-Fuzzy Clustering
Nama : Alvian Rizky Fajar Anggara
NIM : 2008046030
Jurusan : Matematika

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

Wassalamu'alaikum warahmatullahi wabarakatuh

Pembimbing,



Yolanda Norasia, M.Si.
NIP : 199409232019031010

ABSTRAK

Penelitian ini bertujuan untuk memprediksi kebutuhan energi listrik di Provinsi Jawa Tengah pada tahun 2024 hingga 2028 dengan menggunakan dua metode jaringan saraf tiruan, yaitu *Adaline* dan *Backpropagation-Fuzzy Clustering*. Data yang digunakan merupakan data historis konsumsi energi listrik dari tahun 2013 hingga 2023, yang diklasifikasikan berdasarkan enam sektor utama: rumah tangga, industri, komersial, sosial, gedung pemerintah, dan penerangan jalan umum. Proses pemodelan diawali dengan normalisasi data dan pembagian data menjadi *training* dan *testing*. Pada metode *Backpropagation*, data terlebih dahulu diproses dengan *Fuzzy C-Means* untuk memperoleh derajat keanggotaan tiap cluster yang digunakan sebagai fitur tambahan. Hasil penelitian menunjukkan bahwa metode *Adaline* menghasilkan nilai *Mean Absolute Percentage Error* (MAPE) *testing* sebesar 2.67%, sedangkan metode *Backpropagation-Fuzzy Clustering* menghasilkan MAPE sebesar 4.14%. Dengan demikian, metode *Adaline* memiliki akurasi yang lebih tinggi dibandingkan dengan *Backpropagation-Fuzzy Clustering* dalam memprediksi kebutuhan energi listrik pada data ini. Hasil prediksi dari kedua metode divisualisasikan dalam bentuk grafik untuk memudahkan interpretasi pertumbuhan konsumsi energi tiap sektor.

Kata kunci : Prediksi Energi Listrik, *Adaline*, *Backpropagation-Fuzzy Clustering*, *Mean Absolute Percentage Error* (MAPE).

KATA PENGANTAR

Assalamu'alaikum Wr. Wb. Segala puji dan syukur penulis panjatkan kepada Allah SWT, yang telah memberikan kesehatan, kenikmatan, dan kemudahan dalam menyelesaikan skripsi yang berjudul "Prediksi Kebutuhan Energi Listrik di Jawa Tengah tahun 2023-2028 menggunakan Metode Adaline dan Metode Backpropagation-Fuzzy Clustering"

Salawat dan salam dihaturkan kepada junjungan kita, Nabi Agung Muhammad SAW. Beliau adalah manusia pilihan yang telah menerima wahyu dan membawa ilmu ke dunia ini sehingga kita bisa lepas dari zaman kebodohan menuju zaman Islam rahmatan lil alamin.

Penyusunan skripsi ini merupakan salah satu syarat yang harus dipenuhi oleh penulis untuk menyelesaikan pendidikan strata 1 (S1) di Progam Studi Matematika Fakultas Sains dan Teknologi UIN Walisongo Semarang. Dalam penyusunan skripsi ini penulis mendapatkan dukungan serta bimbingan dari berbagai pihak, baik secara langsung maupun tidak langsung. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Prof.Dr.H. Musahadi, M.Ag., selaku Dekan Fakultas Sains dan Teknologi UIN Walisongo Semarang;
2. Budi Cahyono, S.Pd., M.Si. dan Mujiasih, M.Pd, selaku Ketua dan Sekretaris Jurusan Matematika UIN Walisongo Semarang;
3. Any Muanalifah, M.Si., Ph.D dan Prihadi Kurniawan, M.Sc., selaku Ketua dan Sekretaris Progam Studi Matematika Fakultas Sains dan Teknologi UIN Walisongo Semarang;

4. Yolanda Norasia, M.Si., selaku dosen pembimbing yang telah memberikan arahan dan bimbingan dalam menyelesaikan skripsi ini;
5. Nur Khasanah, M.Si., selaku wali dosen yang telah memberikan bimbingan kepada penulis selama menjadi mahasiswa Progam Studi Matematika Fakultas Sains dan Teknologi UIN Walisongo Semarang;
6. Segenap Dosen Jurusan Matematika Fakultas Sains dan Teknologi UIN Walisongo Semarang yang telah memberikan ilmu dan pelajaran bagi penulis selama penulis melakukan studi S1 di UIN Walisongo Semarang;
7. Teman-teman seperjuangan dan seangkatan 2020 terutama teman-teman satu kelas penulis;
8. Segenap keluarga terutama Bapak Ismo Pranoto dan Ibu Sri Wahyuni yang telah mendukung dan mendoakan penulis sehingga penulis dapat menyelesaikan studi;
9. HMJ Matematika UIN Walisongo Semarang;
10. Segenap teman-teman Asisten Laboratorium Matematika (Aslab).

Terakhir penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan dan jauh dari kata sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran demi memperbaiki skripsi ini. Akhir kata mohon maaf atas kesalahan dan kekurangannya.

Semarang, 14 Febuari 2025

Penulis,

Alvian Rizky Fajar Anggara

NIM : 2008046030

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN KEASLIAN	ii
PENGESAHAN	iv
NOTA PEMBIMBING I.....	iv
ABSTRAK	v
KATA PENGANTAR	viii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN	1
A. Latar Belakang Masalah	1
B. Rumusan Masalah	5
C. Tujuan Penelitian.....	6
D. Manfaat Penelitian	6
BAB II LANDASAN PUSTAKA	7
A. Kajian Teori	7
1. Jaringan Saraf Tiruan	7
2. Normalisasi Data	11
3. <i>Backpropagation</i>	12
4. <i>Adaline</i>	20
5. <i>Fuzzy C-Means</i>	23
6. <i>Mean Absolute Percent Error(MAPE)</i>	27
B. Kajian Pustaka	28
BAB III Metode Penelitian	31
A. Langkah-Langkah Penelitian	31
B. Kerangka Kerja Penelitian	32
BAB IV Hasil dan Pembahasan	35
A. Deskripsi Data.....	35
B. Normalisasi Data	37
C. Pola <i>Input</i> dan <i>Output</i>	38

D.	Metode <i>Adaline</i> Pada Kebutuhan Energi Listrik	40
1.	Pembagian Data <i>Training</i> dan Data <i>Testing</i> .	40
2.	Hasil Prediksi	41
E.	Metode <i>Backpropagation</i> - Fuzzy Clustering Pada Kebutuhan Energi Listrik	45
1.	Pembagian Data <i>Training</i> dan <i>Testing</i>	45
2.	Proses <i>Fuzzy C-Means</i> Sebelum <i>Backpropagation</i>	46
3.	Penentuan Jumlah <i>Neuron</i> Tersenyi . . .	48
4.	Hasil Prediksi	50
F.	Analisis Hasil Prediksi	54
	BAB V PENUTUP	57
A.	Kesimpulan	57
B.	Saran	58
	DAFTAR PUSTAKA	60
	Lampiran-lampiran	66

DAFTAR TABEL

Tabel	Judul	Halaman
Tabel 4.1	Data Kebutuhan Energi Listrik Berdasarkan Sektor Tahun 2012-2023 (GWh)	35
Tabel 4.2	Hasil Normalisasi Data Konsumsi Energi Tiap Sektor dengan Rentang 0.1–0.9	38
Tabel 4.3	Pola <i>Input</i> dan <i>Output</i> Multi-Sektor	39
Tabel 4.4	Perbandingan Proporsi Pembagian Data Dari Hasil MSE	41
Tabel 4.5	Perbandingan Nilai Data Metode <i>Adaline</i>	42
Tabel 4.6	Perbandingan Nilai Data Metode <i>Adaline</i> Setelah Denormalisasi	43
Tabel 4.7	Evaluasi Kinerja Model <i>Adaline</i> (MSE dan MAPE)	44
Tabel 4.8	Data Prediksi per Sektor Tahun 2024-2028 Dengan Metode <i>Adaline</i>	45
Tabel 4.9	Perbandingan Proporsi Pembagian Data Dari Hasil MAPE	46
Tabel 4.10	Kombinasi Data <i>Training</i> Dengan Hasil <i>Fuzzy C-Means</i>	48
Tabel 4.11	Kombinasi Data <i>Testing</i> Dengan Hasil <i>Fuzzy C-Means</i>	48
Tabel 4.12	Perbandingan Jumlah <i>Neuron</i> Tersembunyi Dari Hasil MAPE	49
Tabel 4.13	Perbandingan Nilai Data Metode <i>Backpropagation-Fuzzy Clustering</i>	50

Tabel 4.14	Perbandingan Nilai Data Metode <i>Backpropagation-Fuzzy Clustering</i> Setelah Denormalisasi	51
Tabel 4.15	Evaluasi Kinerja Model <i>Backpropagation-Fuzzy Clustering</i> (MSE dan MAPE)	53
Tabel 4.16	Data Prediksi per Sektor Tahun 2024-2028 Dengan Metode <i>Backpropagation-Fuzzy Clustering</i>	53
Tabel 4.17	Hasil Analisis	54
Tabel 0.1	Hasil Normalisasi Data Kebutuhan Energi Listrik per Sektor Tahun 2021-2023	70
Tabel 0.2	Hasil Denormalisasi Metode <i>Adaline</i>	83
Tabel 0.3	Hasil Denormalisasi Metode <i>Backpropagation-Fuzzy Clustering</i>	84

DAFTAR GAMBAR

Gambar	Judul	Halaman
Gambar 2.1	<i>Neuron buatan sederhana</i>	8
Gambar 2.2	<i>Jaringan saraf sederhana</i>	9
Gambar 2.3	<i>Arsitektur Backpropagation</i>	14
Gambar 2.4	<i>Arsitektur Adaline</i>	21
Gambar 3.1	<i>Kerangka Kerja Penelitian</i>	34
Gambar 4.1	<i>Grafik Konsumsi Energi Listrik Berdasarkan Sektor Tahun 2013-2023</i>	36
Gambar 4.2	<i>Perbandingan Data Aktual dan Prediksi per Sektor (2021-2023)</i>	43
Gambar 4.3	<i>Prediksi Konsumsi Energi Listrik per Sektor (2024-2028))</i>	45
Gambar 4.4	<i>Perbandingan Data Aktual dan Prediksi per Sektor (2021-2023)</i>	52
Gambar 4.5	<i>Prediksi Konsumsi Energi Listrik per Sektor (2024-2028))</i>	54
Gambar 4.6	<i>Perbandingan Hasil Prediksi Metode Adaline dan Backpropagation-Fuzzy Clustering per Sektor (2024-2028))</i>	56

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Data Statistik Kebutuhan Energi listrik Provinsi Jawa Tengah Tahun 2012 - 2023 (GWh)	66
Lampiran 2 Hasil Normalisasi	67
Lampiran 3 Script Metode <i>Adaline</i> Dalam Bahasa Pemrograman <i>Python</i>	71
Lampiran 4 Script Metode <i>Backpropagation-Fuzzy</i> <i>Clustering</i> Dalam Bahasa Pemrograman <i>Python</i>	76
Lampiran 5 Hasil Denormalisasi	83

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Pada saat ini energi yang sangat dibutuhkan pada kemajuan manusia diberbagai bidang yaitu energi listrik (Bratawijaya, 2016). Mulai dari bidang rumah tangga, industri, komersial, sosial, gedung pemerintah dan jalan umum. Dalam bidang rumah tangga selain untuk pencahayaan energi listrik dibutuhkan juga untuk menyalakan peralatan seperti setrika, televisi, kipas angin dan peralatan elektronik lainnya. Pada bidang industri energi listrik digunakan untuk menyalakan mesin-mesin dalam beroperasi. Begitu juga dalam bidang komersial, sosial, gedung pemerintah dan jalan umum, hampir seluruh peralatan yang diperlukan di setiap sektor membutuhkan energi listrik (Kurniawati, 2018).

Persediaan energi listrik harus sesuai dengan kebutuhan konsumen. Jika energi listrik yang dihasilkan kurang dari kebutuhan konsumen maka akan terjadinya pemadaman, sedangkan jika melebihi kebutuhan maka akan mengalami rugi daya (Fatchurin, 2020). Berdasarkan permasalahan tersebut, maka penting dilakukan prediksi konsumsi energi listrik. Mengingat energi listrik tidak dapat disimpan dalam jumlah yang besar, namun harus disediakan dalam jangka waktu yang diperlukan (Bratawijaya, 2016).

Prediksi kebutuhan energi listrik harus menggunakan metode perhitungan dengan tingkat akurasi hasil yang sangat baik. Maka sangat penting untuk menguji setiap metode prediksi yang digunakan terlebih dahulu sebelum lanjut melakukan sebuah

prediksi. jika metode yang dipilih akurat maka ketidaksesuaian antara hasil prediksi dengan data asli tidak akan terjadi. Untuk memilih metode prediksi yang paling tepat dalam prediksi kebutuhan energi listrik dapat ditentukan dari nilai kesalahan (*error*) yang terendah (Rahman, 2012).

Banyak metode yang telah dipakai untuk memprediksi jumlah kebutuhan energi listrik. Diantaranya adalah Metode Deret Waktu (*Time Series*), Metode Kausal (Sebab-Akibat) dan Metode *Artificial Intelligence* (Sistem cerdas) (Akbar, 2024). Metode Deret Waktu Menganalisa dengan mencari hubungan antara variabel terkait dengan variabel bebas terhadap waktu seperti metode *Moving average*(MA), *Long Short-Term Memory - Deep Learning*(LSTM) dan *Autoregressive Integrated Moving Average* (Akbar, 2024). Metode ini bertujuan untuk membaca pola data masalalu yang digunakan dalam melakukan eksplorasi di masa yang akan datang. Metode kausal dalam prosesnya menggabungkan beberapa variabel yang mungkin dapat mempengaruhi data kuantitas ramalannya. Metode ini akan mencari hubungan antara variabel satu dengan yang lainnya yang memiliki keterkaitan sebab akibat seperti metode Regresi Linier Sederhana, Regresi Linier Berganda dan Ekonometrika (Masruroh, 2020). Dan metode *Artificial Intelligence* merupakan kecerdasan buatan pada komputasi yang diprogram kemampuannya dalam membaca pola melalui proses pembelajaran (Zulinda, 2020). Seiring dengan perkembangan teknologi komputasi saat ini, menyebabkan pemilihan metode *Artificial Intelligence* menjadi lebih banyak digunakan dalam prediksi kebutuhan energi listrik.

Artificial Intelligence memiliki beberapa program yaitu Logika Fuzzy dan jaringan Syaraf Tiruan (*Neural Network*) (Laksono,

2014). Logika *Fuzzy* merupakan penduga numerik yang dinamis dan terstruktur dengan kemampuan mengembangkan sistem intelijen pada suatu lingkungan yang tidak pasti seperti *Fuzzy C-Means* dan *Fuzzy Tsukamoto* (Laksono, 2014). Dimana desain rangkaian digital yang digunakan deneralisasinya dengan *input* berupa "0" dan "1". Singkatnya metode yang dapat melakukan suatu peramalan dengan prinsip seperti manusia melakukan penalaran dengan nalurinya (Akbar, 2024). Dan pada Jaringan Syaraf Tiruan (JST) informasi sistem pemrosesannya berdasarkan jaringan syaraf biologis manusia sebagai generalisasi model matematika (Sutojo, 2011). Jaringan Syaraf Tiruan (JST) memiliki kelebihan mampu beradaptasi yang menjadikannya bebas estimator dan melakukan pembelajaran dari data inputan yang diberikan. Selain itu JST juga dapat mencapai kemampuan komputasi paralel dengan mempelajari pola hubungan antara input dan output sehingga memberikan data hasil ramalan yang mendekati data aslinya seperti *Backpropagation* dan *Adaline* (Avisia, 2023). Kelebihan tersebut menjadi alasan yang tepat untuk menggunakan JST guna memprediksi kebutuhan energi listrik.

Salah satu metode pada JST adalah *Adaline* yang merupakan salah satu metode dalam metode yang menggunakan lapisan tunggal (*Single Layer*) sehingga membuat pelatihan cenderung cepat (Siang, 2004). Dengan merubah bobotnya menggunakan aturan delta (sering disebut juga *least mean square*). *Adaline* menggunakan fungsi aktivasi linear yang membuat metode *Adaline* cocok untuk data dengan pola sederhana dan cenderung linier (Siang, 2004). Sedangkan metode *Backpropagation* merupakan teknik jaringan saraf tiruan yang menggunakan

lapisan banyak (*Multi Layer*) (Siang, 2004). Dalam metode *Backpropagation* menggunakan fungsi aktivasi non-linear sehingga *Backpropagation* mampu menyelesaikan masalah kompleks, terutama jika terdapat hubungan nonlinier antara input dan output (Siang, 2004).

Tri Hastono dari Universitas PGRI Yogyakarta (2020) dalam penelitiannya yang berjudul "Desain Aplikasi Prediksi Kebutuhan Buah Durian Pelanggan Menggunakan *Neural Network*", menerapkan metode *Adaline* untuk memperkirakan kebutuhan buah durian pelanggan. Tujuan dari penelitian ini adalah untuk memberikan solusi dalam mengelola pasokan dan permintaan buah durian, sehingga kerugian dalam bisnis jual beli oleh tengkulak dapat dikurangi, serta aktivitas bisnis menjadi lebih terstruktur. Muhammad Addin Bratawijaya (2016) dalam penelitian yang berjudul "Analisis Perbandingan Metode Jaringan Saraf Tiruan dan Dengan *Fuzzy Clustering* Pada Proyeksi Kebutuhan Energi Listrik di Indonesia Sampai Tahun 2019", menyatakan metode *Backpropagation* dengan *Fuzzy Clustering* dapat dikombinasikan dan memiliki akurasi lebih baik dengan nilai *error* 1,66% dibandingkan metode *Backpropagation* saja yang mendapatkan *error* 3,31%.

Masruroh dan Kemal Farouq Mauladi dari Universitas Islam Lamongan (2020) dalam penelitian berjudul "Perbandingan Metode Regresi Linear dan *Neural Network Backpropagation* dalam Prediksi Nilai Ujian Nasional SMP Menggunakan *Software R*", menyimpulkan bahwa metode *Neural Network Backpropagation* memberikan hasil prediksi yang lebih akurat dibandingkan dengan metode regresi linear dalam memperkirakan nilai Ujian Nasional SMP di wilayah Lamongan.

Akhmad Bakhrun dari Universitas Komputer Indonesia (2013), melalui penelitiannya yang berjudul "Perbandingan Metode *Adaline* dan *Backpropagation* untuk Prediksi Jumlah Pencari Kerja di Jawa Barat", melakukan peramalan jumlah pencari kerja di wilayah Jawa Barat dengan menggunakan dua metode, yaitu *Adaline* dan *Backpropagation*. Hasil penelitian menunjukkan bahwa metode *Backpropagation* menghasilkan akurasi yang lebih tinggi dibandingkan metode *Adaline*.

Berdasarkan uraian tersebut, peneliti tertarik untuk memilih metode *Backpropagation-Fuzzy Clustering* dan *Adaline* karena keduanya termasuk dalam model pembelajaran terawasi (*supervised learning*) yang mempelajari hubungan antara *input* (fitur) dan *output* (target) berdasarkan data latih. Setelah proses pelatihan, model tersebut dapat digunakan untuk memprediksi *output* dari data baru. Selain itu, kedua metode ini telah terbukti dapat diaplikasikan dalam peramalan (*forecasting*) (Siang, 2004). Dari dua metode tersebut, peneliti ingin mendapatkan hasil prediksi kebutuhan energi listrik serta mendapatkan metode mana yang memiliki tingkat *error* paling kecil. Maka dari itu peneliti akan melakukan penelitian dengan judul **"Prediksi Kebutuhan Energi Listrik di Jawa Tengah Tahun 2023-2028 Menggunakan Metode *Adaline* dan Metode *Backpropagation-Fuzzy Clustering*".**

B. Rumusan Masalah

1. Bagaimana hasil prediksi kebutuhan energi listrik menggunakan metode *Backpropagation-Fuzzy Clustering* ?
2. Bagaimana hasil prediksi kebutuhan energi listrik

menggunakan metode jaringan saraf tiruan *Adaline* ?

3. Bagaimana perbandingan hasil proyeksi jaringan saraf tiruan *Backpropagation-Fuzzy Clustering* dan jaringan saraf tiruan *Adaline* ?

C. Tujuan Penelitian

1. Mengetahui hasil prediksi kebutuhan energi listrik di Jawa Tengah Tahun 2023-2028 menggunakan metode *Backpropagation-Fuzzy Clustering*.
2. Mengetahui hasil prediksi kebutuhan energi listrik di Jawa Tengah Tahun 2023-2028 menggunakan metode jaringan saraf tiruan *Adaline*.
3. Mendapatkan metode yang lebih akurat dalam memprediksi kebutuhan energi listrik di Jawa Tengah Tahun 2023-2028.

D. Manfaat Penelitian

1. Dapat menjadi metode alternatif dalam memprediksi kebutuhan energi listrik oleh PT Perusahaan Listrik Negara (Persero).
2. Dapat menjadi rujukan bagi peneliti lain dalam mengembangkan metode yang lebih efektif dan efisien.

BAB II

LANDASAN PUSTAKA

A. Kajian Teori

1. Jaringan Saraf Tiruan

Jaringan Saraf Tiruan (JST) merupakan sistem pemrosesan informasi yang memiliki karakteristik kinerja tertentu, terutama kemampuan untuk belajar dari data, mengenali pola, dan membuat prediksi atau keputusan berdasarkan data tersebut (Fausett, 1994). Jaringan saraf tiruan dikembangkan sebagai generalisasi dari model matematika kognisi manusia berdasarkan asumsi-asumsi berikut (Fausett, 1994):

- a. Pemrosesan informasi terjadi pada banyak elemen sederhana yang disebut *neuron*.
- b. Sinyal dikirimkan dari satu *neuron* ke *neuron* lainnya melalui jalur koneksi.
- c. Setiap jalur koneksi memiliki bobot yang digunakan untuk mengalikan sinyal yang dikirimkan.
- d. Setiap *neuron* menerapkan fungsi aktivasi (biasanya nonlinier) terhadap *net input* (jumlah dari sinyal *input* yang telah dikalikan bobot) untuk menentukan sinyal *output*-nya.

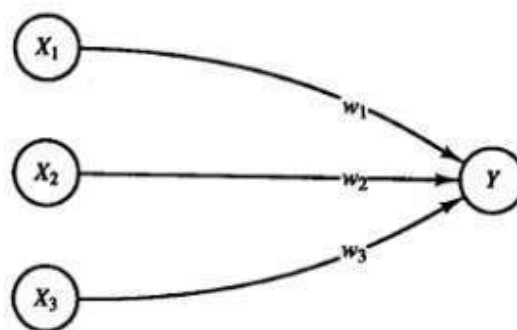
JST memiliki ciri-ciri seperti pola koneksi antar *neuron* (arsitektur jaringan), metode penyesuaian bobot (algoritma pelatihan), dan fungsi aktivasi yang digunakan (Siang,

2004). Jaringan saraf terdiri dari sejumlah besar elemen pemrosesan sederhana yang disebut *neurons*, *units*, *cells*, atau *nodes*. Setiap *neuron* terhubung melalui jalur koneksi terarah, masing-masing dengan bobot tertentu yang mewakili informasi untuk memecahkan masalah (Fausett, 1994).

Jaringan saraf tiruan dapat digunakan untuk berbagai permasalahan seperti pengenalan pola, klasifikasi, pemetaan dari *input* ke *output*, klusterisasi data, serta optimasi dengan batasan tertentu. Setiap *neuron* memiliki status internal (tingkat aktivasi) yang merupakan fungsi dari *input* yang diterima. *Neuron* tersebut mengirimkan aktivasinya sebagai sinyal ke beberapa *neuron* lainnya. Meskipun *neuron* hanya dapat menghasilkan satu sinyal pada satu waktu, sinyal tersebut dapat dikirimkan ke banyak *neuron* penerima.

Sebagai contoh, perhatikan *neuron Y* pada Gambar 2.1 yang menerima masukan dari *neuron X₁*, *X₂*, dan *X₃* dengan sinyal *output* masing-masing *x₁*, *x₂*, dan *x₃*. Bobot pada koneksi dari *X₁*, *X₂*, dan *X₃* ke *Y* adalah *w₁*, *w₂*, dan *w₃*. Maka, *net input* ke *neuron Y* dirumuskan sebagai (Fausett, 1994):

$$y_{in} = w_1x_1 + w_2x_2 + w_3x_3 \quad (2.1)$$

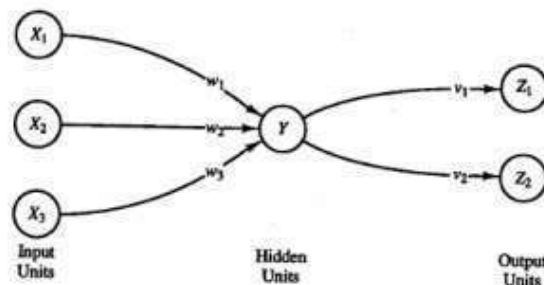


Gambar 2.1. *Neuron* buatan sederhana

Aktivasi y dari *neuron Y* kemudian dihitung dengan fungsi aktivasi, salah satu contohnya adalah fungsi sigmoid logistik (Fausett, 1994):

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

Selanjutnya, misalkan *neuron Y* mengirimkan sinyal ke *neuron* Z_1 dan Z_2 melalui bobot v_1 dan v_2 seperti ditunjukkan pada Gambar 2.2. Meskipun sinyal yang dikirim adalah sama (y), nilai *net input* yang diterima masing-masing *neuron* berbeda karena dikalikan dengan bobot yang berbeda (Siang, 2004).



Gambar 2.2. Jaringan saraf sederhana

Menurut Fausett (1994), jaringan saraf tiruan dapat digunakan untuk berbagai permasalahan berikut:

- a. *Function Approximation*: JST mampu mempelajari hubungan kompleks dan nonlinier antara *input* dan *output*, misalnya untuk prediksi iklim, harga pasar, dan sistem fisik.
- b. *Pattern Classification*: JST digunakan untuk mengklasifikasikan data berdasarkan karakteristik *input*, seperti pengenalan tulisan tangan atau diagnosis penyakit.
- c. *Pattern Association*: JST dapat mengingat atau menghasilkan pola *output* tertentu berdasarkan sebagian *input*, seperti dalam sistem rekomendasi.

- d. *Data Clustering*: Model *unsupervised* seperti *Self-Organizing Map (SOM)* digunakan untuk mengelompokkan data tanpa label, seperti dalam segmentasi pasar.
- e. *Constrained Optimization*: JST dapat digunakan untuk menyelesaikan masalah optimisasi dengan batasan, seperti penjadwalan atau alokasi sumber daya.

Struktur JST umumnya terdiri dari tiga lapisan: lapisan *input*, satu atau lebih lapisan tersembunyi (*hidden layer*), dan lapisan *output*. Arsitekturnya bisa bermacam-macam (Siang, 2004):

- a. *Feedforward network*: sinyal mengalir satu arah dari *input* ke *output*.
- b. *Recurrent network*: terdapat koneksi balik untuk menyimpan memori jangka pendek.
- c. *Single-layer network*: hanya terdiri dari satu lapisan *output*.
- d. *Multi-layer network*: memiliki beberapa lapisan tersembunyi untuk menangani relasi kompleks.

Fungsi aktivasi dalam setiap *neuron* penting untuk menghasilkan keluaran dari *net input*. Fungsi-fungsi yang umum digunakan meliputi (Fausett, 1994):

- a. Fungsi langkah (*step function*).
- b. Fungsi sigmoid.
- c. Fungsi tangen hiperbolik (*tanh*).
- d. Fungsi ReLU (*Rectified Linear Unit*).

Proses pembelajaran dalam JST dilakukan dengan menyesuaikan bobot koneksi guna meminimalkan kesalahan antara *output* dan target. Tahapan utamanya adalah (Siang, 2004):

- a. *Forward propagation*: menghitung *output* berdasarkan *input*.
- b. *Error calculation*: menghitung selisih antara *output* aktual dan target.
- c. *Backward propagation*: memperbarui bobot berdasarkan kesalahan yang dihitung.

2. Normalisasi Data

Normalisasi data merupakan pengelompokan ulang dari domain ke rentang nilai baru antara 0.1 sampai dengan 0.9 (Sabry, 2017). Normalisasi data berfungsi untuk menyamakan skala nilai data yang berbeda, sehingga setiap variabel memiliki bobot yang seimbang dalam proses perhitungan (Pratiwi, 2019). Z-Score, Decimal Scaling, Softmax, dan Min-Max adalah beberapa macam metode dari normalisasi data. (Pratiwi, 2019).

Metode Min-Max dipakai pada penelitian ini, dengan persamaan berikut ini (Sabry, 2017):

$$x' = \frac{(b - a)(x - \min(x))}{\max(x) - \min(x)} + a \quad (2.3)$$

Keterangan :

x'	: Nilai hasil normalisasi
x	: Nilai data asli
$\min(x)$	: Nilai minimum dari data asli
$\max(x)$	: Nilai maksimum dari data asli
a	: Batas bawah rentang normalisasi
b	: Batas atas rentang normalisasi

Sedangkan untuk mengembalikan data normalisasi kedalam bentuk aslinya menggunakan rumus denormalisasi berikut (Siang, 2004).

$$x = \frac{x' - a}{b}(\max(x) - \min(x)) + \min(x) \quad (2.4)$$

Keterangan :

x	: Nilai asli setelah denormalisasi
x'	: Nilai hasil prediksi normalisasi
$\max(x)$	: Nilai maksimum dari data asli
$\min(x)$	: Nilai minimum dari data asli
a	: Batas bawah rentang normalisasi
b	: Batas atas rentang normalisasi

3. *Backpropagation*

Backpropagation adalah metode JST yang dapat digunakan untuk peramalan (Avisia, 2023). Proses *Backpropagation* dioperasikan dengan menyesuaikan bobot jaringan saraf tiruan secara mundur, bergantung dengan nilai *error* yang dihasilkan dalam proses pengenalan pola (Siang, 2004).

Pelatihan jaringan *Backpropagation* terdiri dari tiga tahap utama, yaitu (Siang, 2004).

a. Perambatan maju (*Feedforward Propagation*)

Feedforward propagation adalah proses dalam jaringan saraf tiruan (JST) untuk menghitung *output* berdasarkan *input* yang diberikan. Pada tahap ini, data *input* dikalikan dengan bobot, dijumlahkan, ditambahkan bias, lalu diproses melalui fungsi aktivasi untuk menghasilkan *output* di setiap *neuron*, mulai dari *input layer* hingga *output layer*. Hasil akhir dari *forward propagation* adalah *output* prediksi jaringan yang nantinya akan dibandingkan dengan target untuk menghitung *error*.

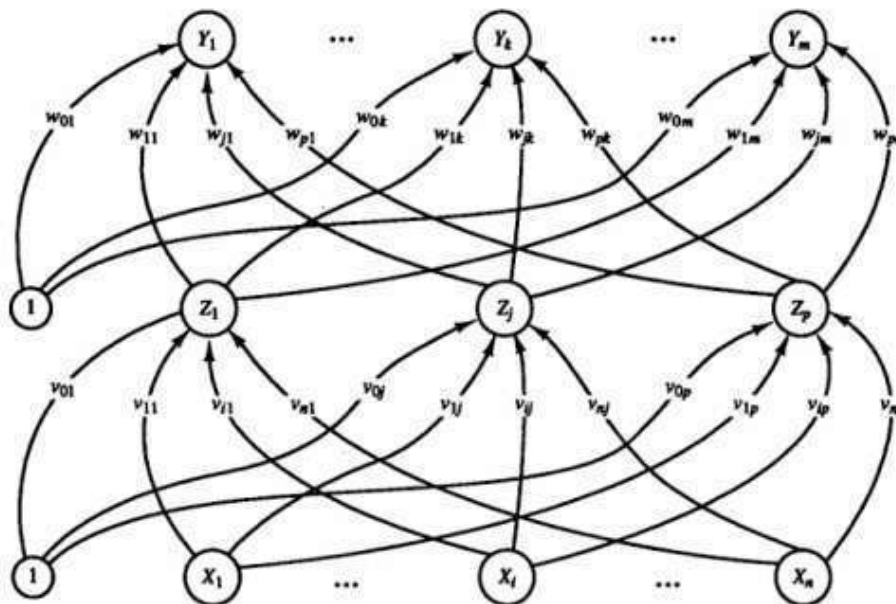
b. Perambatan balik (*Backpropagation*)

Backpropagation adalah proses dalam jaringan saraf tiruan (JST) yang digunakan untuk memperbaiki bobot dan bias berdasarkan *error* dari hasil prediksi. Setelah *error* dihitung, nilai tersebut disebarkan mundur dari *output layer* ke *hidden layer* dengan menghitung turunan (*gradien*) dari fungsi *error* terhadap bobot. Proses ini menentukan seberapa besar kontribusi setiap bobot terhadap *error*.

c. Pembaruan bobot dan bias

Pembaruan bobot dan bias adalah langkah dalam pelatihan jaringan saraf tiruan (JST) untuk memperbaiki kinerja jaringan berdasarkan *error* yang dihitung. Setelah dilakukan *backward propagation*, nilai *gradien* dari *error* digunakan untuk memperbarui bobot dan bias agar jaringan dapat menghasilkan prediksi yang lebih akurat.

Untuk memahami algoritma *Backpropagation*, perlu dikenali terlebih dahulu arsitektur dasar jaringan *Multilayer Perceptron* (MLP) yang terdiri dari lapisan *input*, tersembunyi, dan *output*. Gambar 2.3 menunjukkan arsitektur dengan satu lapisan tersembunyi (unit Z). Unit tersembunyi dan output dapat memiliki bias, masing-masing dilambangkan v_{0j} dan w_{0k} , yang berfungsi seperti bobot dari unit dengan output tetap 1. Gambar hanya menunjukkan aliran informasi saat *feedforward*, sedangkan pada fase *backpropagation*, sinyal kesalahan mengalir berbalik arah untuk pembaruan bobot (Fausett, 1994).



Gambar 2.3. Arsitektur *Backpropagation*

Selama *feedforward*, setiap unit *input* (X_i) menerima sinyal masukan dan menyiarkan sinyal ini ke masing-masing unit tersembunyi $Z_1, \dots, Z_p$. Setiap unit tersembunyi kemudian menghitung aktivasinya dan mengirimkan sinyalnya (z_j) ke setiap unit *output*. Setiap unit *output* (Y_k) menghitung aktivasinya (y_k) untuk membentuk respons jaringan untuk pola masukan yang diberikan.

Selama pelatihan, setiap unit *output* membandingkan aktivasi y_k yang dihitung dengan nilai targetnya t_k untuk menentukan *error* terkait untuk pola tersebut dengan unit tersebut. Berdasarkan *error* ini, faktor $\delta_k (k = 1, \dots, m)$ dihitung. δ_k digunakan untuk mendistribusikan *error* pada unit *output* Y_k kembali ke semua unit di lapisan sebelumnya (unit tersembunyi yang terhubung ke Y_k). Ini juga digunakan (nanti) untuk memperbarui bobot antara *output* dan lapisan tersembunyi. Dengan cara yang sama, faktor $\delta_j (j = 1, \dots, p)$ dihitung untuk setiap unit tersembunyi Z_j . Tidak perlu menyebarkan *error* kembali ke lapisan *input*, tetapi δ_j digunakan untuk memperbarui bobot antara lapisan tersembunyi dan lapisan *input*.

Setelah semua faktor δ telah ditentukan, bobot untuk semua lapisan disesuaikan secara bersamaan. Penyesuaian bobot w_{jk} (dari unit tersembunyi Z_j ke unit *output* Y_k) didasarkan pada faktor δ_k , dan aktivasi z_j dari unit tersembunyi Z_j . Penyesuaian bobot v_{ij} (dari unit *input* X_i ke unit tersembunyi Z_j) didasarkan pada faktor δ_j , dan aktivasi x_i dari unit *input*.

Notasi yang digunakan dalam algoritma pelatihan *Backpropagation* adalah sebagai berikut (Fausett, 1994).

- $\mathbf{X}$: Vektor pelatihan *input*, $\mathbf{X} = (x_1, x_2, \dots, x_n)$
- $\mathbf{t}$: Vektor target *output*, $\mathbf{t} = (t_1, t_2, \dots, t_m)$
- δ_k : Penyesuaian bobot koreksi *error* pada w_{jk} akibat kesalahan di unit *output* Y_k ; juga informasi kesalahan yang disebarkan kembali ke unit tersembunyi.
- δ_j : Penyesuaian bobot koreksi *error* pada v_{ij} akibat *backpropagation* dari unit *output* ke unit tersembunyi Z_j .

- α : Laju pembelajaran (*learning rate*).
- x_i : Unit *input* ke- i ; sinyal *input* dan *output*-nya sama, yaitu x_i .
- v_{0j} : Bias pada unit tersembunyi ke- j .
- Z_j : Unit tersembunyi ke- j .
- z_{in_j} : *Net input* ke unit Z_j , dirumuskan sebagai:

$$z_{in_j} = v_{0j} + \sum_i x_i v_{ij}$$

- z_j : Sinyal *output* dari Z_j , yaitu:

$$z_j = f(z_{in_j})$$

- w_{0k} : Bias pada unit *output* ke- k .
- Y_k : Unit *output* ke- k .
- y_{in_k} : *Net input* ke unit Y_k , dihitung dengan:

$$y_{in_k} = w_{0k} + \sum_j z_j w_{jk}$$

- y_k : Sinyal *output* (aktivasi) dari Y_k :

$$y_k = f(y_{in_k})$$

Secara detail algoritma *Backpropagation* sebagai berikut (Fausett, 1994).

Tahap 0. Inisialisasi bobot. (Atur ke nilai angka kecil).

Tahap 1. Jika kondisi ($Epoch < \text{Maksimum } Epoch$) dan ($MSE < \text{Target Error}$) belum terpenuhi, maka kerjakan langkah 2-9.

Tahap 2. Untuk pasangan pelatihan (*training*), lakukan langkah 3-8.

Tahap 3. Setiap unit *input* ($X_i, i = 1, \dots, n$) menerima sinyal *input* x_i , dan menyebarkan sinyal ini ke semua unit lapisan di atasnya (unit tersembunyi).

Tahap 4. Setiap unit tersembunyi ($Z_j, j = 1, \dots, p$) jumlahkan pada sinyal *input* terbobotnya,

$$z_{inj} = v_{0j} + \sum_{i=1}^n x_i v_{ij}, \quad (2.5)$$

Menerapkan fungsi aktivasinya untuk menghitung sinyal *outputnya*,

$$z_j = f(z_{inj}), \quad (2.6)$$

dan mengirimkan sinyal ke semua unit dilapisan atasnya (unit *output*).

Tahap 5. Setiap unit *output* ($Y_k, k = 1, \dots, m$) dijumlahkan dengan bobot sinyal *inputnya*,

$$y_{in_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad (2.7)$$

dan menggunakan fungsi aktivasinya untuk menghitung sinyal *outputnya*,

$$y_k = f(y_{in_k}). \quad (2.8)$$

Backpropagation:

Tahap 6. Setiap nilai *output* ($Y_k, k = 1, \dots, m$) mendapatkan pola target yang sesuai dengan pola *input* pelatihan dan menghitung informasi *error*nya,

$$\delta_k = (t_k - y_k) f'(y_{in_k}), \quad (2.9)$$

menghitung koreksi bobotnya (digunakan untuk memperbarui w_{jk}),

$$\Delta w_{jk} = \alpha \delta_k z_j, \quad (2.10)$$

hitung juga koreksi biasnya (digunakan untuk memperbarui w_{0k}),

$$\Delta w_{0k} = \alpha \delta_k \quad (2.11)$$

dan kirimkan δ_k ke unit lapisan bawah.

Tahap 7. Tiap nilai tersembunyi ($Z_j, j = 1, 2, 3, \dots, p$) menambahkan delta *input*nya (dari unit di lapisan atasnya)

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (2.12)$$

dikalikan dengan turunan fungsi aktivasinya untuk menghitung informasi *error*nya,

$$\delta_j = \delta_{in_j} f'(z_{in_j}), \quad (2.13)$$

menghitung koreksi bobotnya (digunakan untuk

memperbarui v_{ij}),

$$\Delta v_{ij} = \alpha \delta_j x_i, \quad (2.14)$$

hitung juga koreksi biasnya (digunakan untuk memperbarui v_{0j}),

$$\Delta v_{0j} = \alpha \delta_j \quad (2.15)$$

Perubahan bobot dan bias.

Tahap 8. Setiap unit *output* ($Y_k, k = 1, \dots, m$) dilakukan dalam perubahan bias dan bobot ($j = 0, \dots, p$):

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk}. \quad (2.16)$$

Setiap unit tersembunyi ($Z_j, j = 1, \dots, p$) dilakukan perubahan bobot dan bias ($i = 0, \dots, n$):

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (2.17)$$

Tahap 9 Uji kondisi berhenti (ketika $Epoch < \text{Maksimum } Epoch$) atau *error* sudah sangat kecil (misalnya $< 10^{-6}$) (Fausett, 1994).

Perhatikan bahwa dalam menerapkan algoritma ini, array terpisah harus digunakan untuk delta untuk unit *output* (Tahap 6, δ_k) dan delta untuk unit tersembunyi (Tahap 7, δ_j).

Suatu *epoch* merupakan satu siklus melalui keseluruhan set vektor pelatihan. Biasanya, banyak *epoch* diperlukan untuk melatih jaringan saraf *Backpropagation* (Siang, 2004). Algoritma di atas memperbarui bobot setelah setiap pola pelatihan disajikan. Variasi yang umum adalah pembaruan batch, dimana pembaruan

bobot diakumulasikan selama keseluruhan *epoch* (atau sejumlah penyajian pola lainnya) sebelum diterapkan.

Setelah pelatihan, jaringan saraf *Backpropagation* diterapkan dengan hanya menggunakan fase *feedforward* dari algoritma pelatihan. Prosedur penerapannya adalah sebagai berikut (Fausett, 1994).

Tahap 0. Inisialisasi bobot (dari algoritma pelatihan)

Tahap 1. Untuk setiap vektor *input*, lakukan Tahap 2-4.

Tahap 2. Untuk $i = 1, \dots, n$: atur aktivasi unit *input* x_i ;

Tahap 3. Untuk $j = 1, \dots, p$:

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij}; \quad (2.18)$$

$$z_j = f(z_in_j). \quad (2.19)$$

Tahap 4. Untuk $k = 1, \dots, m$:

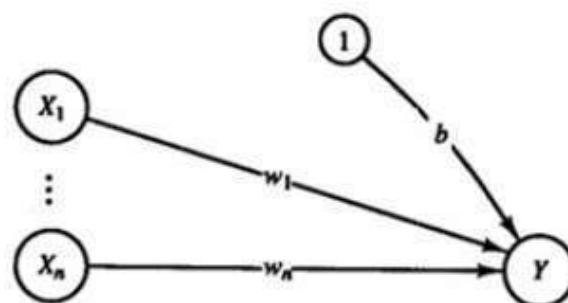
$$y_in_j = v_{0j} + \sum_{i=1}^p z_i w_{jk}; \quad (2.20)$$

$$y_j = f(y_in_j). \quad (2.21)$$

4. Adaline

Adaline merupakan salah satu metode jaringan saraf tiruan (JST) yang paling sederhana. Metode ini pertama kali dikembangkan oleh Bernard Widrow dan Marcian Hoff pada tahun 1960. Secara umum, *Adaline* dapat dilatih menggunakan aturan

delta, yang juga dikenal sebagai *least mean squares* (LMS) atau aturan Widrow-Hoff (Fausett, 1994). Selama pelatihan, aktivasi unit adalah *net input*. Fungsi aktivasi yang digunakan di *Adaline* adalah fungsi identitas (Siang, 2004). Aturan pembelajaran meminimalkan kesalahan kuadrat rata-rata antara aktivasi dan nilai target. Hal ini memungkinkan jaringan untuk terus belajar pada semua pola pelatihan, bahkan setelah nilai *output* yang benar dihasilkan (jika fungsi ambang batas diterapkan) untuk beberapa pola. *Adaline* hanya menggunakan satu unit (neuron) yang menerima masukan dari beberapa unit. Unit ini juga menerima masukan dari "unit" yang sinyalnya selalu +1, agar bobot bias dapat dilatih dengan proses yang sama (aturan delta) seperti yang digunakan untuk melatih bobot lainnya (Sutojo, 2011). Satu ADALINE ditunjukkan pada Gambar 2.4



Gambar 2.4. Arsitektur *Adaline*

Beberapa *Adaline* yang menerima sinyal dari unit masukan yang sama dapat digabungkan dalam jaringan satu lapis. Namun, jika *Adaline* digabungkan sehingga keluaran dari beberapa di antaranya menjadi masukan bagi yang lain, maka jaringan tersebut menjadi berlapis-lapis, dan penentuan bobot menjadi lebih sulit (Avisia, 2023).

Dalam algoritma *Adaline* menggunakan notasi sebagai berikut

(Fausett, 1994).

- x_i : Nilai *input* ke- i (hasil dari s_i , data pelatihan)
- n : Jumlah fitur (unit *input*)
- w_i : Bobot yang terhubung ke input ke- i
- b : Bias (intersep)
- y_n : Jumlah input bersih (net input) ke unit output.
- t : Target *output* (nilai target pelatihan)
- α : Laju pembelajaran (*learning rate*), nilai konstan yang menentukan besar perubahan bobot
- $w_i(\text{baru}), b(\text{baru})$: Bobot dan bias setelah diperbarui
- $w_i(\text{lama}), b(\text{lama})$: Bobot dan bias sebelum diperbarui
- *Epoch* : Satu siklus lengkap melalui seluruh dataset pelatihan

Algoritma pelatihan *Adaline* di jelaskan sebagai berikut (Fausett, 1994).

Tahap 0. Inisialisasi bobot. (Umumnya digunakan nilai acak kecil.)
Menetapkan *learning rate* α .

Tahap 1. Jika kondisi ($\text{Epoch} < \text{Maksimum Epoch}$) belum terpenuhi, lakukan Tahap 2-6.

Tahap 2. Untuk setiap pasangan latihan bipolar $s:t$, lakukan Tahap 3-5.

Tahap 3. Tetapkan aktivasi unit *input*, $i = 1, \dots, n$:

$$x_i = s_i. \quad (2.22)$$

Tahap 4. Hitunglah *net input* ke unit output:

$$y_{in} = b + \sum_i x_i w_i. \quad (2.23)$$

Tahap 5. Perbarui bias dan bobot, $i = 1, \dots, n$:

$$b(\text{baru}) = b(\text{lama}) + \alpha(t - y_{in}). \quad (2.24)$$

$$w_i(\text{baru}) = w_i(\text{lama}) + \alpha(t - y_{in_j})x_i. \quad (2.25)$$

Tahap 6. Tes untuk kondisi berhenti:

Jika perubahan berat terbesar yang terjadi pada Langkah 2 lebih kecil dari toleransi yang ditetapkan, maka hentikan; jika tidak, lanjutkan.

Setelah pelatihan, jaringan saraf *Adaline* Prosedur penerapannya adalah sebagai berikut (Sutojo, 2011).

Tahap 0. Ambil bobot w_i dan bias b hasil dari pelatihan sebelumnya.

Tahap 1. Untuk setiap vektor *input* uji $x = (x_1, \dots, x_n)$, lakukan Tahap 2.

Tahap 2. Hitung *net input* terhadap unit output:

$$y_{in} = b + \sum_{i=1}^n x_i w_i \quad (2.26)$$

5. Fuzzy C-Means

Fuzzy C-Means (FCM) algoritma clustering berbasis partisi yang menggunakan prinsip logika *fuzzy*, di mana satu data

dapat menjadi anggota lebih dari satu cluster dengan derajat keanggotaan yang berbeda-beda antara 0 dan 1 (Sutojo, 2011). FCM cocok digunakan untuk menangani data yang tidak memiliki batasan kelompok yang tegas, misalnya data sosial, medis, maupun data prediksi yang bersifat kompleks (Santhosh, 2021).

FCM bekerja dengan meminimalkan fungsi objektif (*cost function*) yang menghitung jarak antara titik data ke pusat cluster, dikalikan dengan derajat keanggotaan fuzzy (Kusumadewi, 2003):

$$J_m = \sum_{i=1}^N \sum_{j=1}^C \mu_{ij}^m \cdot \|x_i - c_j\|^2 \quad (2.27)$$

Keterangan :

J_m	: Fungsi objektif total (nilai yang diminimalkan)
N	: Jumlah data
C	: Jumlah cluster
μ_{ij}	: Derajat keanggotaan data i terhadap cluster j
m	: Indeks <i>fuzziness</i> , biasanya $m = 2$
x_i	: Data i
c_j	: Pusat cluster j
$\ x_i - c_j\ ^2$	: Jarak <i>Euclidean</i> kuadrat antara x_i dan c_j

Dalam algoritma *Fuzzy C-Means* menggunakan notasi sebagai berikut.

- x_j : Vektor data ke- j , dengan $j = 1, 2, \dots, n$
- n : Jumlah total data
- c : Jumlah klaster
- v_i : Pusat klaster ke- i , dengan $i = 1, 2, \dots, c$

- $U = [u_{ij}]$: Matriks keanggotaan fuzzy, dengan u_{ij} adalah derajat keanggotaan data x_j terhadap kluster i
- m : Parameter fuzzifikasi ($m > 1$), mengatur tingkat kekaburan keanggotaan (umumnya $m = 2$)
- $|x_j - v_i|$: Jarak antara data x_j dan pusat kluster v_i (biasanya menggunakan Euclidean)
- ε : Batas toleransi untuk konvergensi (misalnya 10^{-5})
- t : Indeks iterasi

Algoritma dari *Fuzzy C-Means* adalah sebagai berikut (Kusumadewi, 2003).

Tahap 0. Inisialisasi:

- Tentukan jumlah kluster c ,
- Pilih nilai parameter fuzzifikasi $m > 1$ (umumnya $m = 2$),
- Tentukan batas toleransi ε dan maksimum iterasi,
- Inisialisasi matriks keanggotaan $U^{(0)} = [u_{ij}]$ secara acak, dengan $i = 1, \dots, c$ dan $j = 1, \dots, n$,

ruby Copy code

Tahap 1. Jika kondisi konvergensi ($\|U^{(t+1)} - U^{(t)}\| < \varepsilon$) belum terpenuhi dan iterasi belum melebihi batas maksimum, lakukan Tahap 2–4.

Tahap 2. Hitung pusat kluster (centroid) v_i untuk setiap kluster $i = 1, \dots, c$:

$$v_i = \frac{\sum_{j=1}^n u_{ij}^m x_j}{\sum_{j=1}^n u_{ij}^m} \quad (2.28)$$

Tahap 3. Perbarui derajat keanggotaan u_{ij} untuk setiap data x_j dan kluster i :

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{\|x_j - v_i\|}{\|x_j - v_k\|} \right)^{\frac{2}{m-1}}} \quad (2.29)$$

Tahap 4. Uji konvergensi:

$$\|U^{(t+1)} - U^{(t)}\| < \varepsilon \quad (2.30)$$

Jika ya, hentikan iterasi. Jika tidak, ulangi dari Tahap 2.

Setelah algoritma konvergen, hasil akhir berupa:

- Pusat kluster v_i untuk $i = 1, \dots, c$,
- Matriks keanggotaan $U = [u_{ij}]$ yang menunjukkan derajat keanggotaan tiap data terhadap setiap kluster,
- Data dapat diklasifikasikan ke kluster berdasarkan keanggotaan tertinggi.

Metode FCM dapat dikombinasikan dengan algoritma *Backpropagation* pada Jaringan Saraf Tiruan (JST) untuk meningkatkan akurasi prediksi. Dalam pendekatan ini, FCM digunakan sebagai pra-proses (*preprocessing*) untuk mengelompokkan data sebelum dilatih menggunakan *Backpropagation* (Kusumadewi, 2006). Dengan mekanisme sebagai berikut (Kusumadewi, 2006).

- a. FCM digunakan untuk mengelompokkan data pelatihan ke dalam beberapa cluster, berdasarkan kemiripan pola data.

- b. Setiap data akan memiliki derajat keanggotaan *fuzzy* terhadap setiap cluster, yang dapat digunakan sebagai fitur tambahan atau dimasukkan langsung sebagai *input* ke JST.
- c. Setelah clustering, data *input* (baik asli maupun hasil *fuzzy*) dimasukkan ke *model Backpropagation* untuk proses pelatihan dan prediksi.

6. Mean Absolute Percent Error(MAPE)

Metode *Mean Absolute Percentage Error* (MAPE) merupakan salah satu metode untuk menghitung tingkat akurasi dengan menghitung rata-rata kesalahan kedalam bentuk persentase terhadap nilai aktual. MAPE dipilih untuk menghitung akurasi karena mampu memberikan hasil yang relatif tepat dalam menilai tingkat kesalahan prediksi (Laksmiana, 2019). Jika nilai MAPE kurang dari 10% dapat dikatakan hasil ramalan sangat bagus, jika kurang dari 20% hasil ramalan bagus, dan cukup jika berada dalam rentang 20% hingga 50% (Astuti, 2020). Berikut merupakan persamaan metode MAPE.

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{y'_i - y_i}{y_i} \right| \quad (2.31)$$

keterangan :

N = Jumlah data

y'_i = Hasil prediksi indeks ke- i

y_i = Nilai asli yang ada di indeks ke- i

B. Kajian Pustaka

Penelitian ini selain melakukan kajian teori peneliti juga melakukan kajian pustaka terhadap berbagai karya ilmiah terdahulu yang relevan. Sumber informasi yang digunakan mencakup buku-buku, artikel jurnal, serta referensi lain yang mendukung. Kajian ini bertujuan untuk memperoleh landasan teoritis yang kuat dalam merumuskan dan menyelesaikan permasalahan yang diangkat dalam penelitian. Adapun beberapa penelitian yang dijadikan acuan teoritis dijelaskan sebagai berikut:

Penelitian yang dilakukan oleh Bakhrun (2013) dengan judul "Perbandingan Metode Adaline dan *Backpropagation* untuk Prediksi Jumlah Pencari Kerja di Jawa Barat", bertujuan untuk mengevaluasi dan membandingkan tingkat akurasi dari dua metode prediksi, yaitu Adaline dan jaringan saraf tiruan *Backpropagation*. Studi ini menggunakan data selama enam tahun terakhir, dengan lima variabel *input*: jumlah penduduk, jumlah lulusan, jumlah angkatan kerja, jumlah lowongan kerja, dan jumlah pengangguran terbuka. Hasil penelitian menunjukkan bahwa metode *Backpropagation* menghasilkan akurasi prediksi yang lebih tinggi, dengan rata-rata akurasi sebesar 93,06%, sedangkan metode Adaline hanya mencapai akurasi sebesar 30,92%. Temuan ini menunjukkan bahwa *Backpropagation* lebih unggul dalam memodelkan prediksi jumlah pencari kerja di wilayah Jawa Barat.

Penelitian oleh Bratawijaya, Hermawan, dan Handoko (2016) berjudul "Analisis Perbandingan Metode Jaringan Saraf Tiruan Tanpa dan Dengan Fuzzy Clustering pada Proyeksi Kebutuhan Energi Listrik di Indonesia Sampai Tahun 2019" bertujuan untuk

memproyeksikan konsumsi energi listrik nasional dari tahun 2014 hingga 2019 serta membandingkan dua metode peramalan, yaitu *JST Backpropagation* dan kombinasi *JST Backpropagation* dengan Fuzzy Clustering. Penelitian ini menggunakan data historis Produk Domestik Bruto (PDB), jumlah pelanggan, dan konsumsi energi listrik Indonesia selama periode 2002–2013. Hasilnya menunjukkan bahwa metode *JST Backpropagation* menghasilkan proyeksi konsumsi energi listrik sebesar 272.630 GWh pada tahun 2019 dengan rata-rata kenaikan tahunan sebesar 6,22% dan nilai MAPE sebesar 3,31%, sedangkan metode hybrid JST-Fuzzy Clustering memberikan proyeksi sebesar 284.663 GWh dengan kenaikan rata-rata 6,96% dan MAPE yang lebih rendah sebesar 1,66%. Hal ini menunjukkan bahwa metode JST-Fuzzy Clustering memiliki tingkat akurasi yang lebih tinggi dan hasil proyeksi yang lebih mendekati data perencanaan resmi pemerintah (RUPTL).

Tahun 2020, Hastono melakukan penelitian pada prediksi kebutuhan buah durian pelanggan menggunakan *Neural Network*, dalam penelitian ini merancang aplikasi *Neural Network* untuk melakukan prediksi dengan metode *Adaline*. Data yang digunakan mencakup empat tahun terakhir terkait permintaan pelanggan terhadap buah durian. Dalam penelitian ini, metode *Adaline* digunakan untuk meminimalisasi kerugian dalam bisnis jual beli durian serta membantu mengorganisir kegiatan bisnis secara lebih efektif.

Pada penelitian Zakia Intan Ali (2020), memprediksi curah hujan di kota padang menggunakan metode *Backpropagation* dan *Adaline*. Penelitian ini membahas perbandingan antara metode *Backpropagation* dan *Adaline* dalam memprediksi siklus hujan. Data yang digunakan yaitu data curah hujan tahun 2003 sampai

dengan 2019 yang didapat dari tiga stasiun penakar hujan yang berada di kota padang.

Masruroh (2020) dalam penelitiannya yang berjudul "Perbandingan Metode Regresi Linear dan *Neural Network Backpropagation* dalam Prediksi Nilai Ujian Nasional SMP Menggunakan Software R", membahas perbandingan antara dua metode prediksi, yaitu regresi linear dan *Backpropagation* neural network, terhadap hasil Ujian Nasional siswa SMP. Penelitian ini memanfaatkan software R untuk proses analisis dan menggunakan data nilai siswa dari SMPN 1 dan SMPN 2 Lamongan. Hasil evaluasi menunjukkan bahwa metode regresi linear menghasilkan nilai rata-rata RMSE sebesar 9,04 dan MAPE sebesar 3,94%. Sementara itu, metode *Backpropagation* memberikan hasil yang lebih akurat dengan nilai rata-rata RMSE sebesar 7,28 dan MAPE sebesar 0,55%. Oleh karena itu, dapat disimpulkan bahwa *Backpropagation* merupakan metode yang lebih unggul dalam melakukan prediksi nilai Ujian Nasional pada kasus ini.

Pada penelitian Lemuel Clark P. Velasco (2021), yang berjudul *Performance analysis of artificial neural network models for hour-to-hour electric load forecasting* Membahas mengenai kinerja jaringan saraf tiruan pada prediksi beban energi listrik jam kedepan.

Tahun 2022, Donni Nasution,dkk .Dalam penelitiannya yang berjudul Perbandingan Algoritma Adaline Berdasarkan Pola *Input Data Dan Aktivasi Output Untuk Prediksi Data*. Menghasilkan kinerja metode *Adaline* dengan empat pola aktivasi menunjukan akurasi yang sama yaitu sebesar 100%. dalam penelitiannya menyatakan bahwa metode *Adaline* dapat memprediksi dengan baik penyakit stroke.

BAB III

Metode Penelitian

A. Langkah-Langkah Penelitian

Penelitian ini dilakukan melalui beberapa tahapan sistematis guna membangun model prediksi kebutuhan energi listrik menggunakan metode *Adaline* dan *Backpropagation-Fuzzy Clustering*. Adapun langkah-langkah penelitian yang dilakukan adalah sebagai berikut:

1. Studi Literatur

Dilakukan kajian pustaka dari berbagai referensi seperti jurnal, buku, dan penelitian terdahulu yang berkaitan dengan jaringan saraf tiruan, metode *Adaline*, *Backpropagation*, dan *Fuzzy C-Means Clustering*.

2. Pengumpulan dan Persiapan Data

Data historis kebutuhan energi listrik per sektor dikumpulkan dari tahun 2012 hingga 2023. Data tersebut kemudian diolah dan disusun dalam format *input* dan *target* sesuai kebutuhan model.

3. Normalisasi Data

Data dinormalisasi menggunakan metode *Min-Max Normalization* ke dalam rentang $[0.1, 0.9]$ agar semua fitur berada dalam skala yang seragam.

4. Pembagian Data *Training* dan *Testing*

Data dibagi menjadi dua bagian, yaitu data *training*

(70%–90%) dan data *testing* (10%–30%) untuk menguji performa model pada data baru.

5. Implementasi Metode *Adaline*

Model *Adaline* dilatih menggunakan data pelatihan untuk membentuk hubungan linier antara *input* dan *output*. Evaluasi dilakukan menggunakan nilai MSE dan MAPE.

6. Implementasi Metode *Backpropagation-Fuzzy Clustering*

Data input diperkaya dengan derajat keanggotaan dari hasil *Fuzzy C-Means*. Model *Backpropagation* dilatih dan dievaluasi dengan metrik yang sama seperti *Adaline*.

7. Prediksi Tahun 2024–2028

Model digunakan untuk memprediksi kebutuhan energi listrik selama lima tahun ke depan menggunakan pendekatan prediksi bertahap (*multi-step forecasting*).

8. Evaluasi dan Perbandingan Metode

Kinerja kedua metode dibandingkan berdasarkan nilai MSE dan MAPE. Hasil juga divisualisasikan untuk melihat tren prediksi antar sektor.

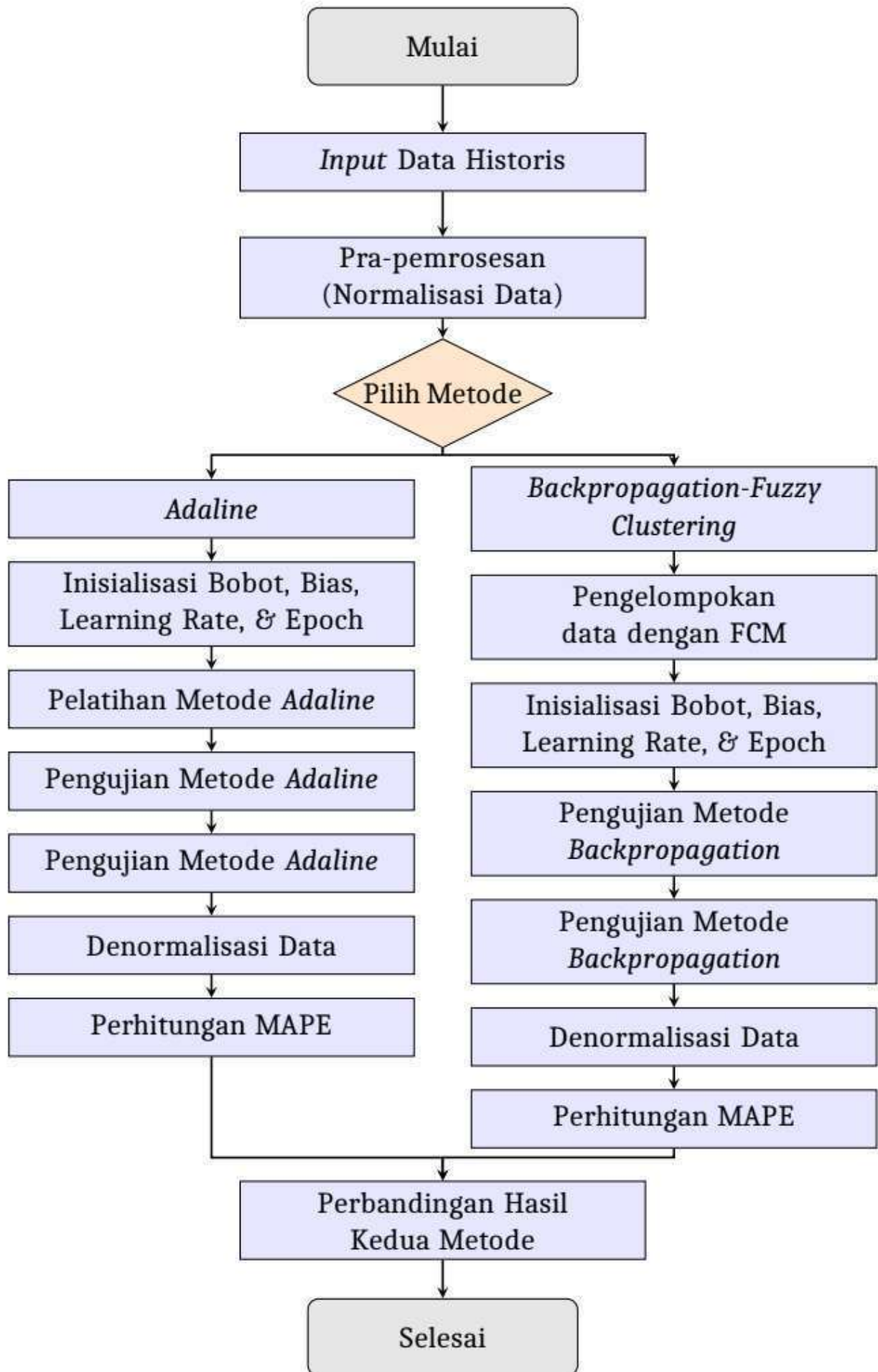
9. Kesimpulan dan Saran

Berdasarkan hasil evaluasi, ditentukan metode yang paling efektif serta diberikan saran untuk pengembangan penelitian selanjutnya.

B. Kerangka Kerja Penelitian

Penelitian ini membandingkan metode *Backpropagation-Fuzzy Clustering* dan *Adaline* dalam memprediksi kebutuhan energi

listrik di Jawa Tengah untuk menentukan metode yang lebih akurat. Proses dimulai dengan pengumpulan dan normalisasi data historis, kemudian diproses menggunakan kedua metode. *Backpropagation-Fuzzy Clustering* menerapkan klasterisasi sebelum pelatihan jaringan saraf tiruan dengan algoritma *Backpropagation*, sementara *Adaline* menggunakan pendekatan pembaruan bobot selama pembelajaran. Tingkat akurasi prediksi dihitung menggunakan *Mean Absolute Percentage Error* (MAPE) untuk menemukan metode yang terbaik. Berikut adalah gambaran kerangka kerja penelitian.



Gambar 3.1. Kerangka Kerja Penelitian

BAB IV

Hasil dan Pembahasan

Penelitian ini memprediksi kebutuhan energi listrik di Jawa Tengah untuk tahun 2021 sampai tahun 2028 dengan memprediksi setiap tahunnya dari 2021 sampai dengan 2028. Untuk prediksi tahun 2021 sampai tahun 2023 akan dievaluasi akurasi menggunakan metode MAPE, sedangkan untuk prediksi tahun 2024 sampai tahun 2028 akan diproyeksikan menggunakan grafik untuk melihat prediksi kenaikan dan penurunan kebutuhan energi listrik di Jawa Tengah. Dengan tahapan sebagai berikut ini.

A. Deskripsi Data

Penelitian ini menggunakan data historis kebutuhan energi listrik di Provinsi Jawa Tengah dari tahun 2013 hingga 2023. Data diklasifikasikan berdasarkan sektor konsumen, seperti sektor rumah tangga, industri, komersial, sosial, pemerintahan, dan penerangan jalan umum. Data kebutuhan energi listrik di Jawa Tengah terlihat pada Tabel 4.1.

Tabel 4.1. Data Kebutuhan Energi Listrik Berdasarkan Sektor Tahun 2012-2023 (GWh)

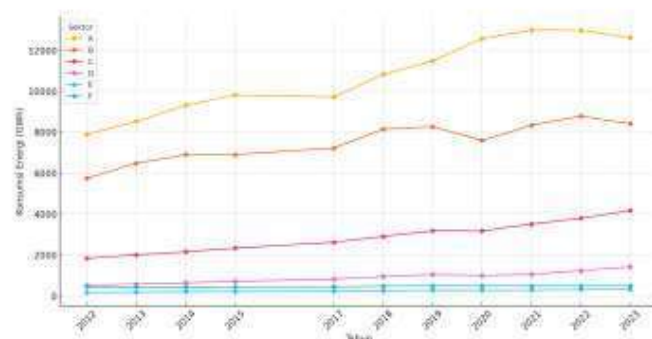
Tahun	A	B	C	D	E	F
2012	7898.11	5739.43	1834.39	522.09	168.2	439.2
2013	8521.86	6475.75	2006.97	578.59	183.45	438.46
2014	9301.29	6898.15	2153.79	640.27	197.04	440.93
2015	9806.95	6901.46	2339.49	706.08	208.52	445.69
2017	9717.11	7223.07	2618.18	819.70	227.56	451.42

Tahun	A	B	C	D	E	F
2018	10815.71	8142.04	2907.20	950.35	260.05	482.68
2019	11475.07	8269.00	3187.26	1046.12	282.44	490.73
2020	12556.06	7592.91	3168.80	995.98	282.37	494.62
2021	12986.94	8332.48	3512.97	1055.15	284.09	489.53
2022	12961.71	8778.3	3789.54	1239.73	304.65	490.7
2023	12620.27	8408.15	4170.67	1418.82	324.66	489.18

Keterangan :

- A** : Rumah Tangga
- B** : Industri
- C** : Komersial
- D** : Sosial
- E** : Gedung Pemerintah
- F** : Penerangan Jalan Umum

Penelitian ini mengkomparasikan dua metode yakni *Adaline* dan *Backpropagation-Fuzzy Clustering*. Dalam Penelitian ini analisis deskriptif bertujuan untuk menjelaskan gambaran umum konsumsi energi listrik berdasarkan data tahun 2013 hingga 2023 dari berbagai sektor di Provinsi Jawa Tengah.



Gambar 4.1. Grafik Konsumsi Energi Listrik Berdasarkan Sektor Tahun 2013-2023

Berdasarkan hasil analisis deskriptif data konsumsi energi listrik Provinsi Jawa Tengah tahun 2013–2023, diketahui bahwa sektor rumah tangga merupakan penyumbang konsumsi energi terbesar dengan tren yang terus meningkat setiap tahunnya, diikuti oleh sektor industri dan komersial. Sementara itu, sektor sosial dan gedung pemerintah menunjukkan lonjakan signifikan sejak tahun 2017, yang mengindikasikan peningkatan kebutuhan layanan publik. Di sisi lain, sektor penerangan jalan umum menunjukkan pola konsumsi yang relatif stabil dari tahun ke tahun.

B. Normalisasi Data

Sebelum dilakukan pelatihan, seluruh data terlebih dahulu dinormalisasi menggunakan metode Min-Max Scaling ke dalam rentang 0.1 hingga 0.9, agar setiap fitur berada dalam skala yang seragam dan mempercepat proses konvergensi. Dalam proses ini menggunakan Persamaan (2.3).

$$x' = \frac{0.8(x - \min(x))}{\max(x) - \min(x)} + 0.1$$

Gunakan rumus normalisasi data diatas untuk diterapkan pada data historis dengan hasil sebagai berikut.

$$\begin{aligned} x_1 &= \frac{0.8(7898.11 - 168.2)}{12986.94 - 168.2} + 0.1 = 0.582413092 \\ x_2 &= \frac{0.8(8521.86 - 168.2)}{12986.94 - 168.2} + 0.1 = 0.621340475 \\ x_3 &= \frac{0.8(9301.29 - 168.2)}{12986.94 - 168.2} + 0.1 = 0.669983633 \\ &\vdots \\ x_{66} &= \frac{0.8(489.18 - 168.2)}{12986.94 - 168.2} + 0.1 = 0.120031922 \end{aligned}$$

Untuk perhitungan lengkap terlihat pada lampiran 2. Hasil normalisasi data kebutuhan listrik berdasarkan sektor terlihat pada Tabel 4.2.

Tabel 4.2. Hasil Normalisasi Data Konsumsi Energi Tiap Sektor dengan Rentang 0.1–0.9

A	B	C	D	E	F
0.5824131	0.447693	0.203985	0.122086	0.1	0.11691
0.6213405	0.493646	0.214755	0.125612	0.10095	0.11687
0.6699836	0.520007	0.223918	0.129461	0.1018	0.11702
0.7015412	0.520214	0.235507	0.133568	0.10252	0.11732
0.6959344	0.540285	0.2529	0.140659	0.1037	0.11768
0.7644965	0.597636	0.270937	0.148813	0.10573	0.11963
0.8056463	0.60556	0.288415	0.15479	0.10713	0.12013
0.8731094	0.563366	0.287263	0.151661	0.10713	0.12037
0.9	0.609522	0.308743	0.155353	0.10723	0.12005
0.8984254	0.637345	0.326003	0.166873	0.10852	0.12013
0.8771166	0.614244	0.349789	0.178049	0.10976	0.12003

C. Pola Input dan Output

Dalam penelitian ini, pola *input* dan *output* sangat menentukan kualitas hasil prediksi model (Zhang, 2017). Pola *Input* dan *Output Multi-Sektor* dipakai karna cocok pada data dan cocok untuk metode *Adaline* dan *Backpropagation-Fuzzy Clustering*. Merujuk pada data historis konsumsi energi dari berbagai sektor (rumah tangga, industri, komersial, sosial, pemerintahan, dan penerangan jalan umum), sedangkan output adalah konsumsi sektor tertentu atau gabungan sektor di tahun berikutnya (Sutaryo, 2021).

Pola *Input* Semua Sektor Tahun ke- t untuk *Output* Semua Sektor Tahun ke- $t+1$ dipilih dalam penelitian ini karena bertujuan

untuk memprediksi konsumsi energi seluruh sektor dari tahun 2021 hingga 2028 secara simultan. Dengan pendekatan ini, keterkaitan antar sektor dapat dimodelkan secara menyeluruh, sejalan dengan pendekatan multivariat pada prediksi waktu yang terbukti efektif dalam sistem energi kompleks (Zhang, 2017) (Sutaryo, 2021). Dengan hasil pola seperti pada Tabel 4.3.

Tabel 4.3. Pola *Input* dan *Output* Multi-Sektor

No	<i>Input</i>				<i>Output</i>			
1	A1	B1	...	F1	A2	B2	...	F2
2	A2	B2	...	F2	A3	B3	...	F3
3	A3	B3	...	F3	A4	B4	...	F4
4	A4	B4	...	F4	A5	B5	...	F5
5	A5	B5	...	F5	A6	B6	...	F6
6	A6	B6	...	F6	A7	B7	...	F7
7	A7	B7	...	F7	A8	B8	...	F8
8	A8	B8	...	F8	A9	B9	...	F9
9	A9	B9	...	F9	A10	B10	...	F10
10	A10	B10	...	F10	A11	B11	...	F11
11	A11	B11	...	F11	A12	B12	...	F12

Tabel 4.3 menyajikan susunan data yang digunakan untuk melatih model prediksi, di mana setiap baris memuat nilai konsumsi energi listrik dari enam sektor (A–F) pada tahun ke- t sebagai *input* dan nilai konsumsi sektor yang sama pada tahun ke- $(t + 1)$ sebagai *output* (target). Pola ini memungkinkan model mempelajari hubungan lintas sektor dan waktu secara simultan, sehingga cocok digunakan dalam pendekatan multivariat seperti *Adaline* maupun *Backpropagation* (Zhang, 2017) (Sutaryo, 2021).

D. Metode Adaline Pada Kebutuhan Energi Listrik

1. Pembagian Data *Training* dan Data *Testing*

Dalam proses pemodelan, data umumnya dibagi menjadi dua bagian utama, yaitu data latih (*training*) dan data uji (*testing*). Proporsi pembagian ini sangat penting dan perlu disesuaikan dengan kebutuhan penelitian. Biasanya, data latih diberikan porsi lebih besar untuk memastikan bahwa model memiliki cukup informasi untuk belajar secara optimal, sementara data uji digunakan untuk mengevaluasi kemampuan model dalam memprediksi data yang belum pernah dilihat sebelumnya. Umumnya, proporsi data uji berkisar antara 10% hingga 30% dari keseluruhan dataset (Kurniawan, 2020). Pembagian data yang tidak tepat dapat berdampak pada performa model. Jika data latih terlalu sedikit, jaringan saraf mungkin gagal memahami pola dengan benar. Sebaliknya, jika terlalu banyak data dimasukkan ke dalam proses pelatihan, hal ini dapat memperlambat proses konvergensi dan menyebabkan overfitting, di mana model justru menghafal data daripada melakukan generalisasi (Kurniawan, 2020).

Dalam penelitian ini, peneliti memakai pembagian data *training* dan data *testing* sebesar 70% data *training* dan 30% data *testing*, 80% data *training* dan 20% data *testing* serta 90% data *training* dan data *testing*. Dari ketiga proporsi tersebut dibandingkan dan di pilih proporsi yang paling baik. Berikut adalah hasil evaluasi untuk masing-masing proporsi pembagian data *training* dan data *testing*.

Tabel 4.4. Perbandingan Proporsi Pembagian Data Dari Hasil MSE

Pembagian Data	MAPE <i>Training</i>	MAPE <i>Testing</i>
70%-30%	1.83%	2.67%
80%-20%	1.70%	3.15%
90%-10%	1.63%	4.01%

Berdasarkan Tabel 4.4 hasil evaluasi kinerja model dengan tiga skenario pembagian data (70%-30%, 80%-20%, dan 90%-10%), diperoleh bahwa pembagian data 70%-30% memberikan hasil terbaik. Hal ini ditunjukkan oleh nilai MAPE *Testing* yang paling rendah sebesar 2.67%, yang menandakan bahwa model mampu melakukan generalisasi dengan baik terhadap data yang belum pernah dilatih. Dengan demikian, pembagian 70%-30% menjadi pilihan yang paling tepat untuk menghasilkan model dengan keseimbangan antara akurasi pelatihan dan kemampuan prediksi.

2. Hasil Prediksi

Setelah memperoleh skenario pembagian data terbaik, yaitu 70% data sebagai data *training* dan 30% sebagai data *testing*, maka dilakukan proses pelatihan model menggunakan metode *Adaline* dengan pola data *input* dan *output* dari tahun 2012 hingga 2023. Data historis ini dimanfaatkan untuk membangun hubungan linier antara variabel-variabel *input* dan *output* pada masing-masing sektor pengguna energi listrik. Model yang telah dilatih kemudian digunakan untuk melakukan prediksi konsumsi energi pada periode selanjutnya. Hasil dari proses pelatihan dan prediksi ditunjukkan pada Tabel 4.5.

Tabel 4.5. Perbandingan Nilai Data Metode *Adaline*

Tahun	A	B	C	D	E	F
2021	0.9	0.60952	0.30874	0.15535	0.10723	0.12005
2021*	0.87152	0.59765	0.30105	0.15828	0.10819	0.12079
2022	0.89843	0.63734	0.32600	0.16687	0.10852	0.12013
2022*	0.91618	0.61660	0.31618	0.16403	0.10957	0.12767
2023	0.87712	0.61424	0.34979	0.17805	0.10976	0.12003
2023*	0.93500	0.62578	0.32248	0.16649	0.11020	0.12211

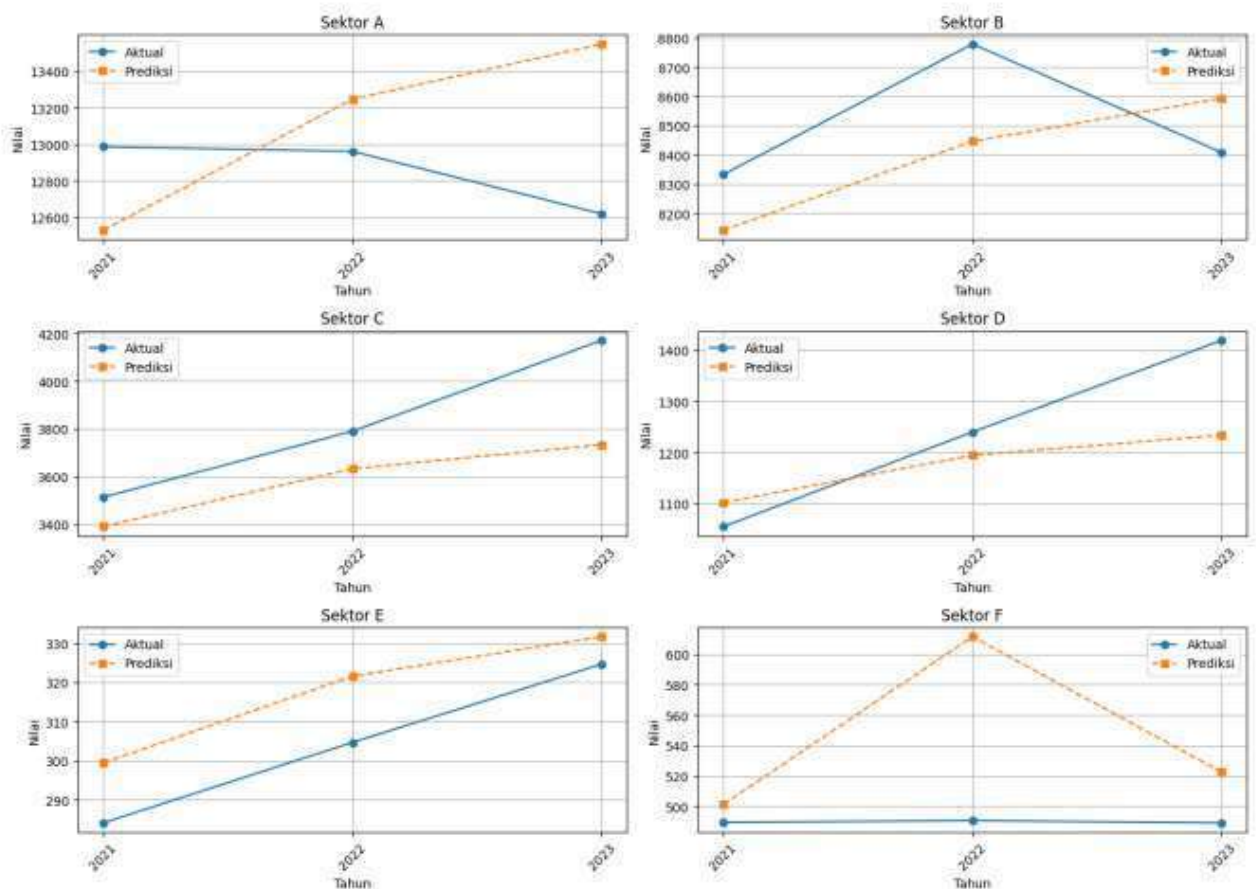
Simbol tanda bintang (*) pada baris tahun menunjukkan nilai hasil prediksi. Hasil prediksi yang diperoleh dari model *Adaline* masih berada dalam bentuk data ternormalisasi. Oleh karena itu, perlu dilakukan proses denormalisasi agar nilai prediksi dapat dikembalikan ke skala aslinya dan dapat dibandingkan secara langsung dengan data aktual. Proses denormalisasi menggunakan Persamaan (2.4) dengan hasil berikut.

$$\begin{aligned}
 x_1 &= \frac{0.87152-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 12530.53 \\
 x_2 &= \frac{0.91618-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 13246.18 \\
 x_3 &= \frac{0.93500-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 13547.79 \\
 &\vdots \\
 x_{18} &= \frac{0.12211-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 522.48
 \end{aligned}$$

Perhitungan secara lengkap terdapat di lampiran 5. Diperoleh hasil denormalisasi dan dibandingkan dengan data aktual, terlihat pada Tabel 0.2 dan Gambar 4.3 .

Tabel 4.6. Perbandingan Nilai Data Metode *Adaline* Setelah Denormalisasi

Tahun	A	B	C	D	E	F
2021	12986.94	8332.48	3512.97	1055.15	284.09	489.53
2021*	12530.53	8142.29	3389.71	1101.98	299.42	501.25
2022	12961.71	8778.30	3789.54	1239.73	304.65	490.70
2022*	13246.18	8445.97	3632.21	1194.23	321.59	611.50
2023	12620.27	8408.15	4170.67	1418.82	324.66	489.18
2023*	13547.79	8593.01	3733.06	1233.61	331.69	522.48



Gambar 4.2. Perbandingan Data Aktual dan Prediksi per Sektor (2021–2023)

Untuk menilai kinerja model *Adaline*, dilakukan evaluasi menggunakan dua metrik, yaitu Mean Squared Error (MSE) dan

Mean Absolute Percentage Error (MAPE). Evaluasi dilakukan baik pada data *training* maupun data *testing*, guna mengukur tingkat kesalahan prediksi dan kemampuan generalisasi model terhadap data baru. Hasil evaluasi tersebut disajikan pada Tabel 4.7, yang menunjukkan metode *Adaline* baik untuk memprediksi data kebutuhan energi listrik.

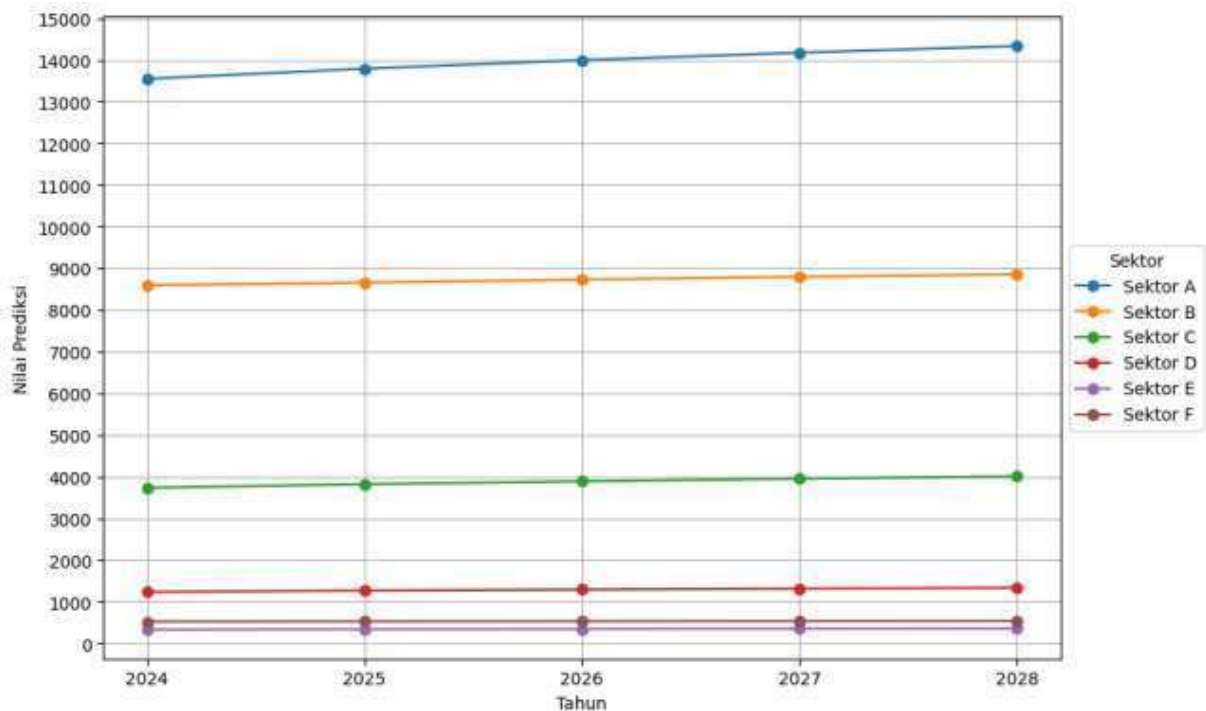
Tabel 4.7. Evaluasi Kinerja Model *Adaline* (MSE dan MAPE)

MSE Training	MSE Testing	MAPE Testing (%)
0.000164	0.000347	2.67%

Setelah model *Adaline* selesai diterapkan pada tahun 2012 sampai 2023, model tersebut digunakan untuk memprediksi konsumsi energi listrik pada enam sektor berbeda dari tahun 2024 hingga 2028. Proses prediksi dilakukan secara bertahap (*multi-step forecasting*) dengan cara rekursif, yaitu hasil prediksi suatu tahun dijadikan masukan untuk prediksi tahun berikutnya (Siang, 2004). Data tahun 2023 digunakan sebagai titik awal *input*, karena merupakan data terakhir dalam pelatihan. Pendekatan ini umum digunakan dalam peramalan data deret waktu karena mampu mengikuti tren data secara berurutan (Sutojo, 2011). Prediksi dihitung menggunakan bobot dan bias akhir dari pelatihan *Adaline* tanpa melibatkan fungsi aktivasi. Hasil prediksi tahun 2024 sampai tahun 2028 selengkapnya ditampilkan pada Tabel 4.8 dan Gambar berikut ini.

Tabel 4.8. Data Prediksi per Sektor Tahun 2024-2028 Dengan Metode Adaline

Tahun	A	B	C	D	E	F
2024	13547.79	8593.01	3733.06	1233.61	331.69	522.48
2025	13791.09	8655.39	3818.78	1264.31	338.63	526.16
2026	13994.64	8729.52	3888.73	1290.29	344.65	529.74
2027	14174.79	8796.53	3950.53	1313.31	350.03	532.97
2028	14334.86	8856.17	4005.43	1333.78	354.84	535.86



Gambar 4.3. *Prediksi Konsumsi Energi Listrik per Sektor (2024-2028)*

E. Metode *Backpropagation* - Fuzzy Clustering Pada Kebutuhan Energi Listrik

1. Pembagian Data *Training* dan *Testing*

Dalam pemodelan menggunakan metode *Backpropagation* berbasis *Fuzzy Clustering*, pembagian data menjadi data latih

dan data uji berperan penting dalam menjamin kemampuan model dalam mempelajari pola sekaligus menghindari overfitting. Umumnya, data uji dialokasikan sebesar 10% hingga 30% dari total data, agar model memperoleh cukup informasi pelatihan sekaligus bisa diuji pada data baru (Kurniawan, 2020).

Penelitian ini menguji tiga skenario pembagian data, yaitu 70%-30%, 80%-20%, dan 90%-10% antara data latih dan uji. Hasil evaluasi terhadap nilai MAPE ditunjukkan pada Tabel 4.9. Dari ketiga skenario tersebut, pembagian 70%-30% memberikan performa terbaik dengan nilai MAPE *testing* terendah sebesar 4.14%, sehingga dipilih sebagai skenario optimal untuk pelatihan model.

Tabel 4.9. Perbandingan Proporsi Pembagian Data Dari Hasil MAPE

Pembagian Data	MAPE <i>Training</i>	MAPE <i>Testing</i>
70%-30%	2.54%	4.14%
80%-20%	2.73%	7.15%
90%-10%	2.13%	4.69%

2. Proses Fuzzy C-Means Sebelum Backpropagation

Sebelum melatih model *Backpropagation*, data *input* terlebih dahulu diproses menggunakan algoritma Fuzzy C-Means (FCM) untuk menambahkan informasi keanggotaan terhadap beberapa cluster. Proses ini bertujuan agar model dapat mengenali pola kompleks melalui fitur tambahan hasil clustering (Bratawijaya, 2016). Dalam penelitian ini digunakan tiga cluster ($c = 3$) dengan parameter fuzzy $m = 2$ dan toleransi konvergensi $\varepsilon = 10^{-5}$. Menjadikan tiga cluster dipilih peneliti karena

pada kecenderungan data yang dapat dikelompokkan menjadi tiga pola konsumsi utama, yaitu rendah, sedang, dan tinggi. Jumlah cluster sebaiknya ditentukan sesuai dengan jumlah data dan kompleksitas variasi dalam data. Jumlah cluster yang melebihi jumlah data menyebabkan hasil klasterisasi tidak berarti (Kusumadewi, 2003). Hasil FCM berupa matriks keanggotaan (U) yang menunjukkan derajat partisipasi setiap data dalam tiap cluster. Matriks ini dikombinasikan dengan fitur asli (X) untuk membentuk *input* baru yang digunakan dalam pelatihan jaringan saraf *Backpropagation*, guna meningkatkan akurasi prediksi (Kusumadewi, 2003).

$$X_{\text{fuzzy}} = [X_{\text{asli}}; |; U]$$

Hasil dari kombinasi X_{asli} dan U pada data *training* terlihat pada Tabel 4.10. Sedangkan untuk hasil pada data *testing* terdapat pada Tabel 4.11.

Tabel 4.10. Kombinasi Data *Training* Dengan Hasil *Fuzzy C-Means*

Data Input Training (X_{Asli})					
0.582413	0.447693	0.203985	0.122086	0.1	0.116913
0.62134	0.493646	0.214755	0.125612	0.100952	0.116867
0.669984	0.520007	0.223918	0.129461	0.1018	0.117021
0.701541	0.520214	0.235507	0.133568	0.102516	0.117318
0.695934	0.540285	0.2529	0.140659	0.103705	0.117675
0.764497	0.597636	0.270937	0.148813	0.105732	0.119626
0.805646	0.60556	0.288415	0.15479	0.10713	0.120129
Hasil Fuzzy C-Means					
0.034862	0.955931	0.009207			
0.159508	0.81745	0.023041			
0.906078	0.070034	0.023888			
0.974643	0.013467	0.01189			
0.933886	0.028225	0.037889			
0.042587	0.010511	0.946902			
0.021587	0.007129	0.971284			

Tabel 4.11. Kombinasi Data *Testing* Dengan Hasil *Fuzzy C-Means*

Data Input Testing (X_{Asli})					
0.873109	0.563366	0.287263	0.151661	0.107125	0.120371
0.9	0.609522	0.308743	0.155353	0.107233	0.120054
0.898425	0.637345	0.326003	0.166873	0.108516	0.120127
Hasil Fuzzy C-Means (U)					
0.180495	0.07505	0.744455			
0.181282	0.084602	0.734116			
0.183707	0.088939	0.727354			

3. Penentuan Jumlah *Neuron* Tersembunyi

Dalam metode *Backpropagation-Fuzzy Clustering* arsitektur jaringan yang digunakan terdiri dari tiga lapisan utama yakni lapisan *input*, lapisan tersembunyi (*hidden layer*), dan

lapisan *output*. Jumlah neuron pada lapisan *input* dan *output* masing-masing disesuaikan dengan jumlah variabel. Sementara itu, jumlah neuron pada lapisan tersembunyi ditentukan menggunakan beberapa aturan praktis, yakni (Heaton, 2008).

1. Jumlah *hidden neuron* harus berada diantara ukuran *input layer* dan *output layer*.
2. Jumlah *hidden neuron* harus $\frac{2}{3}$ ukuran *input layer*, ditambah ukuran *output layer*.
3. Jumlah *hidden neuron* harus kurang dari dua kali ukuran *input layer*.

Berdasarkan *rules of thumb* untuk memperkirakan Jumlah *hidden neuron* (Heaton, 2008). maka pada penelitian ini menggunakan hidden neuron 4, 8, dan 10 untuk menguji ketiga model yang memberikan nilai error paling sedikit. Didapatkan hasil perbandingan MAPE terlihat pada Tabel 4.12.

Tabel 4.12. Perbandingan Jumlah Neuron Tersembunyi Dari Hasil MAPE

Jumlah Neuron Tersembunyi	MAPE Training	MAPE Testing
4 Lapisan	2.85%	7.60%
8 Lapisan	2.54%	4.14%
10 Lapisan	2.27%	9.19%

Tabel 4.12 Menunjukan bahwasannya menggunakan 8 lapisan tersembunyi menghasilkan nilai MAPE *testing* terbaik 4.14%. Maka dalam penelitian ini menggunakan 8 lapisan untuk lapisan tersembunyi.

4. Hasil Prediksi

Sebelum dilakukan pelatihan dengan metode *Backpropagation*, data terlebih dahulu diproses menggunakan metode *Fuzzy C-Means*. Lalu memperoleh skenario pembagian data terbaik dan jumlah *neuron* tersembunyi, yaitu 70% data sebagai data *training* dan 30% sebagai data *testing* dengan 8 lapisan tersembunyi, maka dilakukan proses pelatihan model menggunakan metode *Backpropagation-Fuzzy Clustering* dengan pola data *input* dan *output* dari tahun 2012 hingga 2023. Model yang telah dilatih kemudian digunakan untuk melakukan prediksi konsumsi energi pada periode selanjutnya. Hasil dari proses pelatihan dan prediksi ditunjukkan pada Tabel 4.13.

Tabel 4.13. Perbandingan Nilai Data Metode
Backpropagation-Fuzzy Clustering

Tahun	A	B	C	D	E	F
2021	0.90000	0.60952	0.30874	0.15535	0.10723	0.12005
2021*	0.81988	0.59865	0.30370	0.16113	0.10693	0.12215
2022	0.89843	0.63734	0.32600	0.16687	0.10852	0.12013
2022*	0.81762	0.60041	0.30646	0.16415	0.10782	0.12160
2023	0.87712	0.61424	0.34979	0.17805	0.10976	0.12003
2023*	0.81633	0.59919	0.30383	0.16473	0.10801	0.12072

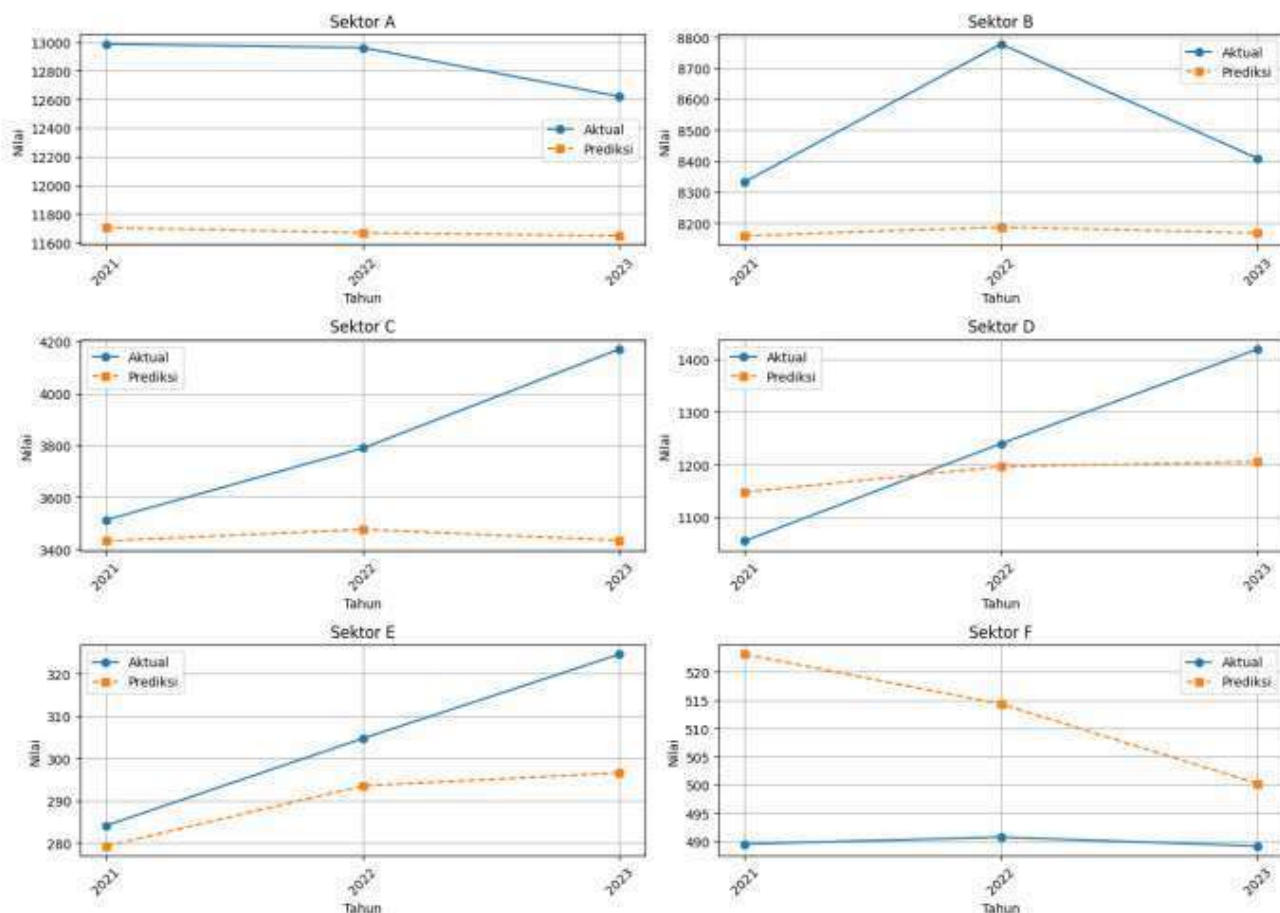
Simbol tanda bintang (*) pada baris tahun menunjukkan nilai hasil prediksi. Hasil prediksi yang diperoleh dari model *Backpropagation-Fuzzy Clustering* masih berada dalam bentuk data ternormalisasi. Oleh karena itu, perlu dilakukan proses denormalisasi agar nilai prediksi dapat dikembalikan ke skala aslinya dan dapat dibandingkan secara langsung dengan data aktual. Proses denormalisasi menggunakan Persamaan (2.4) dengan hasil berikut.

$$\begin{aligned}
 x_1 &= \frac{0.81988-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 11703.14 \\
 x_2 &= \frac{0.89843-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 11666.93 \\
 x_3 &= \frac{0.81633-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 11646.26 \\
 &\vdots \\
 x_{18} &= \frac{0.12072-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 500.21
 \end{aligned}$$

Perhitungan secara lengkap terdapat di lampiran 5. Diperoleh hasil denormalisasi dan dibandingkan dengan data aktual, terlihat pada Tabel 0.3 dan Gambar 4.5 .

Tabel 4.14. Perbandingan Nilai Data Metode
Backpropagation-Fuzzy Clustering Setelah
 Denormalisasi

Tahun	A	B	C	D	E	F
2021	12986.94	8332.48	3512.93	1055.15	284.09	489.53
2021*	11703.14	8158.28	3432.17	1147.71	279.24	523.12
2022	12961.71	8778.30	3789.54	1239.73	304.72	490.75
2022*	11666.93	8186.48	3476.40	1196.10	293.50	514.31
2023	12620.32	8408.15	4170.67	1418.82	324.66	489.18
2023*	11646.26	8166.93	3434.25	1205.40	296.55	500.21



Gambar 4.4. Perbandingan Data Aktual dan Prediksi per Sektor (2021–2023)

Untuk menilai kinerja model *Backpropagation-Fuzzy Clustering*, dilakukan evaluasi menggunakan dua metrik, yaitu *Mean Squared Error* (MSE) dan *Mean Absolute Percentage Error* (MAPE). Evaluasi dilakukan baik pada data *training* maupun data *testing*, guna mengukur tingkat kesalahan prediksi dan kemampuan generalisasi model terhadap data baru. Hasil evaluasi tersebut disajikan pada Tabel 4.15, yang menunjukkan metode *Backpropagation-Fuzzy Clustering* baik untuk memprediksi data kebutuhan energi listrik.

Setelah proses pelatihan selesai dilakukan, model *Backpropagation* digunakan untuk melakukan prediksi konsumsi

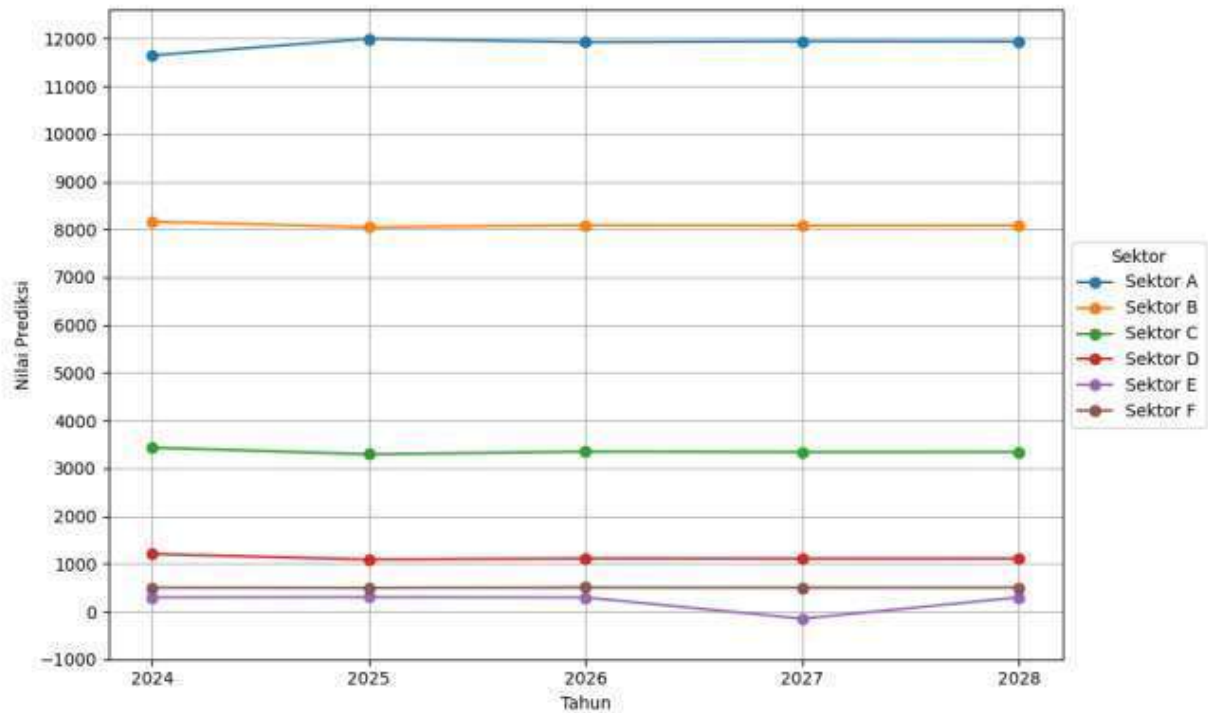
Tabel 4.15. Evaluasi Kinerja Model *Backpropagation-Fuzzy Clustering* (MSE dan MAPE)

MSE Training	MSE Testing	MAPE Testing (%)
0.000244	0.001172	4.14%

energi pada periode tahun 2024 hingga 2028. Pendekatan yang digunakan adalah prediksi bertahap (multi-step forecasting) dengan cara dimana hasil prediksi pada suatu tahun digunakan sebagai *input* untuk memprediksi tahun berikutnya. Metode ini umum digunakan dalam peramalan deret waktu karena mampu memperpanjang proyeksi berdasarkan pola jangka pendek (Zhang, 2017). Hasil prediksi ditampilkan dalam Tabel 4.16, yang menunjukkan estimasi konsumsi energi di masing-masing sektor. Visualisasi pada Gambar 4.5 juga dapat digunakan untuk memperlihatkan tren pertumbuhan antar sektor secara lebih jelas dan informatif.

Tabel 4.16. Data Prediksi per Sektor Tahun 2024-2028 Dengan Metode *Backpropagation-Fuzzy Clustering*

Tahun	A	B	C	D	E	F
2024	11646.26	8166.93	3434.25	1205.40	296.55	500.21
2025	11996.53	8047.40	3293.09	1089.87	301.03	495.08
2026	11922.02	8091.94	3349.97	1111.50	294.46	504.21
2027	11941.73	8083.45	3340.84	1108.45	296.71	501.49
2028	11936.76	8085.85	3343.24	1109.42	296.23	501.97



Gambar 4.5. *Prediksi Konsumsi Energi Listrik per Sektor (2024–2028)*

F. Analisis Hasil Prediksi

Analisis akurasi hasil *Backpropagation-Fuzzy Clustering* dan *Adaline* dilakukan dengan Metode MAPE (*Mean Absolute Percentage Error*), yang mengukur tingkat kesalahan prediksi dalam bentuk persentase. Semakin kecil hasil persentase MAPE, semakin akurat model yang digunakan. Adapun hasil analisisnya adalah sebagai berikut:

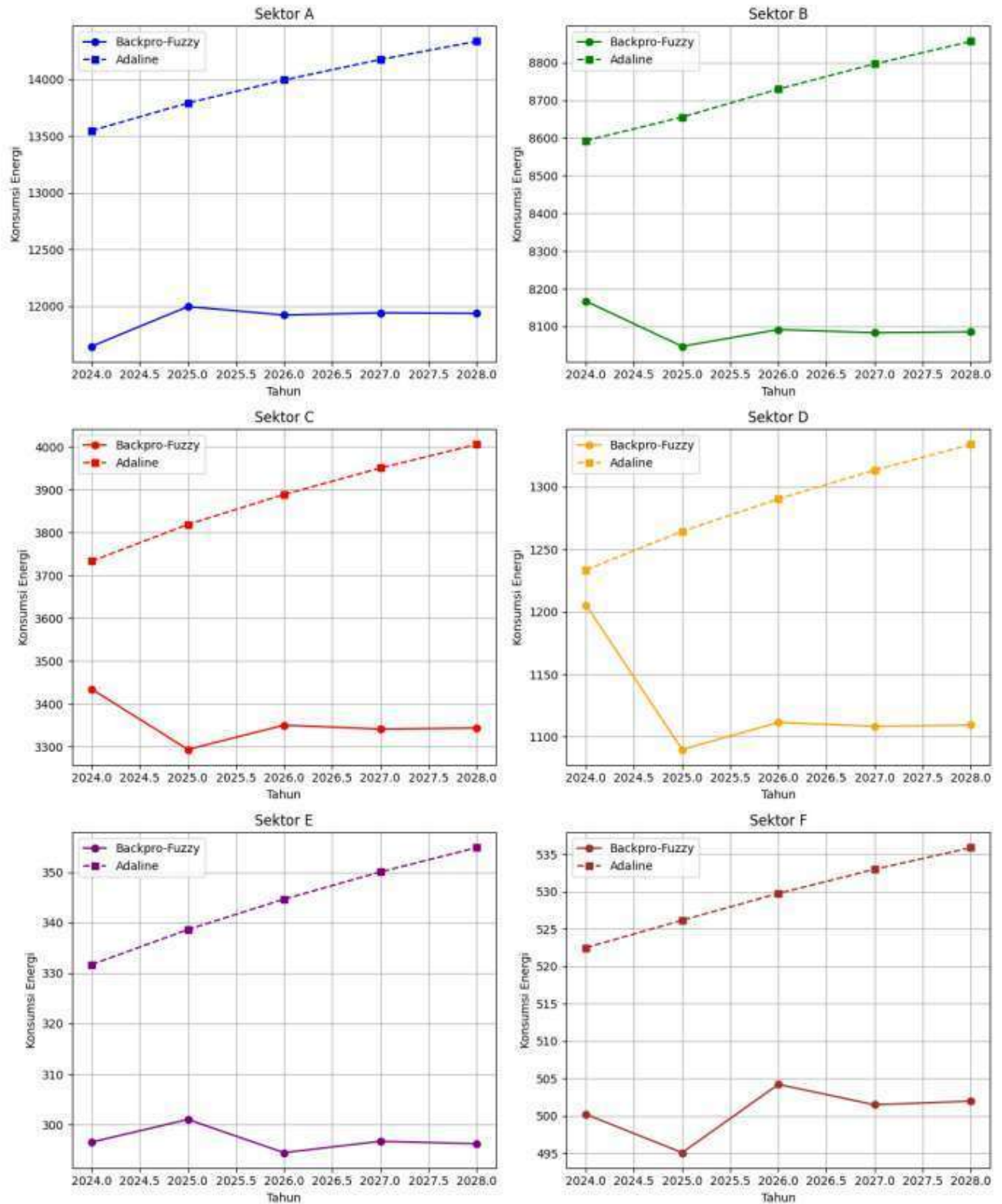
Tabel 4.17. Hasil Analisis

Hasil MAPE	
<i>Adaline</i>	2.67%
<i>Backpropagation-Fuzzy Clustering</i>	4.14%

Dalam proses perhitungan *Mean Absolute Percentage Error*

(MAPE), diperoleh bahwa metode *Adaline* menunjukkan performa yang lebih bagus dibandingkan dengan metode *Backpropagation-Fuzzy Clustering*, dengan hasil kesalahan sebesar 2.67%. Sementara itu, metode *Backpropagation-Fuzzy Clustering* memiliki nilai MAPE sebesar 4.14%, yang masih tergolong rendah namun tidak seakurat metode *Adaline*. Hal ini dipengaruhi oleh metode *Adaline* unggul dalam dataset kecil, pola data yang relatif linier, dan skenario dengan kebutuhan komputasi ringan. Sedangkan untuk metode *Backpropagation-Fuzzy Clustering* baru akan lebih unggul jika dataset besar dan pola hubungan kompleks/non-linier, serta dengan penyetelan arsitektur yang tepat (Bakhrun, 2013).

Pada Gambar 4.6 menyajikan perbandingan hasil prediksi konsumsi energi pada masing-masing sektor (A hingga F) selama periode tahun 2024 hingga 2028. Secara umum, terlihat bahwa metode *Adaline* menghasilkan tren yang cenderung lebih linier dan meningkat secara konsisten, sementara hasil dari metode *Backpropagation-Fuzzy Clustering* menunjukkan naik dan turun atau kecenderungan yang lebih realistis terhadap dinamika data sebelumnya.



Gambar 4.6. Perbandingan Hasil Prediksi Metode *Adaline* dan *Backpropagation-Fuzzy Clustering* per Sektor (2024–2028))

BAB V

PENUTUP

A. Kesimpulan

Penelitian ini bertujuan untuk memprediksi kebutuhan energi listrik di Provinsi Jawa Tengah pada periode 2023 hingga 2028 dengan menggunakan dua metode jaringan saraf tiruan, yaitu *Adaline* dan *Backpropagation* yang dikombinasikan dengan *Fuzzy Clustering*. Berdasarkan hasil analisis data dan evaluasi model, dapat disimpulkan bahwa.

1. Hasil prediksi kebutuhan energi listrik menggunakan metode *Backpropagation-Fuzzy Clustering* menunjukkan bahwa metode ini dapat digunakan untuk meramalkan konsumsi energi listrik pada enam sektor pengguna (rumah tangga, industri, komersial, sosial, PJU, dan pemerintahan). Namun, hasil prediksi cenderung menunjukkan fluktuasi yang tidak stabil, termasuk nilai prediksi negatif pada sektor tertentu. Nilai MAPE pada data pengujian sebesar 4.14%, yang mengindikasikan bahwa tingkat akurasi model ini masih kurang optimal, terutama untuk data yang bersifat linier atau tidak terlalu kompleks.
2. Hasil prediksi menggunakan metode jaringan saraf tiruan *Adaline* memberikan hasil yang lebih akurat dan stabil dibandingkan *Backpropagation-Fuzzy*. *Adaline* mampu mengikuti pola historis data secara lebih konsisten dan menghasilkan nilai prediksi yang logis pada semua sektor. Nilai MAPE pengujian sebesar 2.67%, menunjukkan

bahwa model ini efektif dalam menangani prediksi jangka menengah dengan pola data yang cenderung linier.

3. Perbandingan antara kedua metode menunjukkan bahwa metode *Adaline* lebih unggul dibandingkan *Backpropagation-Fuzzy Clustering* dalam hal akurasi prediksi dan kestabilan hasil. *Adaline* lebih sesuai digunakan karena dalam penelitian ini memakai data yang memiliki kecenderungan pola linier (data konsumsi energi listrik per sektor), sedangkan *Backpropagation-Fuzzy Clustering* memiliki kelemahan pada penelitian ini karena data yang digunakan bersifat sederhana dan linier, sehingga berisiko mengalami *overfitting*. Dengan demikian, *Adaline* direkomendasikan sebagai pendekatan utama dalam peramalan kebutuhan energi listrik jangka menengah di Jawa Tengah.

B. Saran

Berdasarkan hasil penelitian dan kesimpulan yang telah diperoleh, maka beberapa saran yang dapat disampaikan untuk pengembangan penelitian selanjutnya adalah sebagai berikut:

1. Penambahan jumlah data historis dalam proses pelatihan model sangat disarankan (misal penambahan data biaya, jumlah pelanggan dan penambahan pembangkit listrik), agar jaringan saraf tiruan, khususnya *Backpropagation-Fuzzy Clustering*, dapat mempelajari pola yang lebih kompleks dan menghasilkan prediksi yang lebih akurat serta stabil.
2. Model *Adaline* yang telah terbukti memberikan hasil lebih

stabil dapat dikombinasikan dengan metode lain, seperti analisis regresi untuk memberikan validasi tambahan terhadap hasil prediksi.

DAFTAR PUSTAKA

- Agustin, M. & Prahasto, T. 2012. Penggunaan Jaringan Syaraf Tiruan Backpropagation untuk Seleksi Penerimaan Mahasiswa Baru pada Jurusan Teknik Komputer Di Politeknik Negeri Sriwijaya. *Jurnal Sistem Informasi Bisnis*, 2(2).
- Akbar, S.(2023). *Analisis Perbandingan Akurasi Peramalan Konsumsi Energi Listrik Dengan Algoritma Extreme Learning Machine Dan Backpropagation* . Skripsi. Pekanbaru: Progam Sarjana.
- Ali, Z. I. 2020. *Aritificial Neural Network Untuk Memprediksi Curah Hujan di Kota Padang dengan Metode Backpropagation dan Adaline*. Skripsi. Semarang :Progam Sarjana.
- Arikunto, S. 2013.*Prosedur Penelitian Suatu Pendekatan Praktik*. Jakarta: Rineka Cipta.
- Arjmand, B., dkk. 2022. Machine Learning: A New Prospect In Multi Omics Data Analysis Of Cancer. *Frontiers In Genetics*, 13.
- Astuti, D. P. P., & Mashuri, 2020. Penerapan Metode Fuzzy Tsukamoto dan Fuzzy Sugeno Dalam Penentuan Harga Jual Sepeda Motor. *UNNES Journal Of Mathematics*, 9(2), 74-78.
- Avissa.R.(2023). *Analisis Perbandingan Metode Jaringan Saraf Tiruan Adaline (Adaptive Linear Neuron) Dan Backpropagation Dalam Kasus Covid-19 Kota Palembang*. Skripsi. Palembang: Progam Sarjana.

- Bakhrun, A.(2013). *Perbandingan metode adaline dan backpropagation untuk prediksi jumlah pencari kerja di Jawa Barat* . Skripsi. Bandung: Progam Sarjana.
- Badan Pusat Statistik. (2018). "Statistik Listrik 2012-2017". Jakarta: BPS. Diakses dari <https://www.bps.go.id/publication>.
- Badan Pusat Statistik. (2019). "Statistik Listrik 2013-2018". Jakarta: BPS. Diakses dari <https://www.bps.go.id/publication>.
- Badan Pusat Statistik. (2021). "Statistik Listrik 2015-2020". Jakarta: BPS. Diakses dari <https://www.bps.go.id/publication>.
- Badan Pusat Statistik. (2023). "Statistik Listrik 2017-2021". Jakarta: BPS. Diakses dari <https://www.bps.go.id/publication>.
- Badan Pusat Statistik. (2023). "Statistik Listrik 2018-2022". Jakarta: BPS. Diakses dari <https://www.bps.go.id/publication>.
- Badan Pusat Statistik. (2024). "Statistik Listrik 2019-2013". Jakarta: BPS. Diakses dari <https://www.bps.go.id/publication>.
- Bratawijaya, M. A., Hermawan & Handoko, S. 2016. Analisis Perbandingan Metode Jaringan Saraf Tiruan Tanpa dan Dengan Fuzzy Clustering Pada Proyeksi Kebutuhan Energi Listrik Di Indonesia. *TRANSIENT: Jurnal Ilmiah Teknik Elektro*, 5(1).

- Dewi, K. N. A., Bhari, S. & Irwansyah. 2019. Model Prediksi Curah Hujan Harian Menggunakan Jaringan Syaraf Tiruan Backpropagation. *Indonesian Physical Review*, 2(1), 9-17.
- Etikasari, B. & Puspitasari, T. D. 2019. Pengenalan Pola Huruf Dengan Menggunakan Algoritma Adaline. *Jurnal Mnemonic*, 2(1), 12-16.
- Fatchurin, E., Fanani, A. & Hafiyusholeh, M. 2020. Peramalan Penggunaan Bahan Bakar Pada Pembangkit Listrik Tenaga Gas Uap Menggunakan Algoritma Backpropagation Neural Network. *Jurnal Riset dan Aplikasi Matematika*, 4(2), 82-92.
- Fausett, L. 1994. *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. Prentice-Hall
- Halim, S. & Wibisono, A. M. 2000. Penerapan Jaringan Saraf Tiruan Untuk Peramalan. *Jurnal Teknik Industri*, 2(2), 106 - 114.
- Hastono, T. 2020. *Desain Aplikasi Prediksi Kebutuhan Buah Durian Pelanggan Menggunakan Neural Network*. Seri Prosiding; Seminar Nasional Dinamika Informatika Universitas PGRI Yogyakarta, 200–205.
- Heaton, J. 2008. *Introduction to Neural Networks with Java*. Heaton Research.
- Julpan, Nababan, E. B. & Zarlis, M. 2015. Analisis Fungsi Aktivasi Sigmoid Biner dan Sigmoid Bipolar Dalam Algoritma Backpropagation pada Prediksi Kemampuan Siswa. *Jurnal Teknovasi*, 2(1), 103-106.
- Kurniawan, D. 2020. *Pengenalan Pembelajaran Mesin: Teori dan Praktik*. Yogyakarta: ANDI Yogyakarta.

- Kurniawati,D.(2018) *Prediksi Energi Listrik Dengan Metode Jaringan Syaraf Tiruan Backpropagation Menggunakan Matlab Untuk Kota Semarang Tahun 2019-2024*. Skripsi. Semarang: Progam Sarjana.
- Kusumadewi, S. 2003. *Fuzzy Logic: Teori dan Aplikasi*. Yogyakarta: Graha Ilmu.
- Kusumadewi, S. & Hartati, S. 2006. *Neuro-Fuzzy: Integrasi Sistem Fuzzy dan Jaringan Syaraf*. Yogyakarta: Graha Ilmu.
- Laksmmana, R. D., Santoso, E. & Rahayudi, B. 2019. Prediksi Penjualan Roti Menggunakan Metode Exponential Smoothing (Studi Kasus : Harum Bakery). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3(5), 4933-4941.
- Laksono,H.D & Fajira,N.2014. Peramalan Beban Listrik Daerah Sumatera Barat Jangka Panjang Dengan Menggunakan Integrasi Jaringan Saraf Tiruan dan Sistem Fuzzy. *Jurnal Teknologi Informasi dan pendidikan*, 7(1).
- Masruroh & Mauladi, K. F. 2020. Perbandingan Metode Regresi Linear dan Neural Network Backpropagationdalam Prediksi Nilai Ujian Nasional SMP Menggunakan Software R. *JOUTICA*, 5(1), 331-336.
- Nasution,D., Nasution,D. & Solikhun.(2022) Perbandingan Algoritma Adaline Berdasarkan Pola Input Data Dan Aktivasi Output Untuk Prediksi Data. *BRAHMANA: Jurnal Penerapan Kecerdasan Buatan*, 4(1), 96-105.
- Nazir, M. 2003. *Metode Penelitian*. Jakarta: Ghalia Indonesia.

- Pratiwi, D., Santoso, G. B., Muslimah, L., & Rizki, R. D. 2019. An Intelligent Dengue Hemorrhagic Fever Severity Level Detection Based On Deep Neural Network Approach. *Jurnal Ilmu Komputer Dan Informasi (Journal Of Computer Science And Information)*, 12(2), 57-66.
- Rahman, A., Abdullah, A. G. & Hakim, D. L. 2012. Prakiraan Beban Puncak Jangka Panjang Pada Sistem Kelistrikan Indonesia Menggunakan Algoritma Adaptive Neuro-Fuzzy Inference System. *ELECTRANS*, 11(2), 18-26.
- Sabry,A.E. dan Arabo,W.(2017) "A Normalization Methods for Backpropagation: A Comparative Study". *Science Journal of University of Zakho*, 5(4), 319-323.
- Santhosh, R., & Mohanapriya, M. (2021). Generalized fuzzy logic based performance prediction in data mining. *Materials Today: Proceedings*, 45(2), 1770-1774
- Siang, J. J. 2004. *Aplikasi Jaringan Syaraf Tiruan dan Pemrograman Menggunakan MATLAB*. Yogyakarta: ANDI Yogyakarta.
- Sutaryo, D. & Rahayu, S. 2021. Perbandingan Model Multivariate dan Univariate untuk Prediksi Konsumsi Energi Listrik di Indonesia. *Jurnal Teknologi dan Sistem Komputer*
- Sutojo, T., Suhartono, V., & Mulyanto, E. 2011. *Kecerdasan Buatan*. Yogyakarta: ANDI Yogyakarta.
- Velasco, L, C, P., Anerjo, K. A. S. & Macarat, J. S. S. 2021. Performance analysis of artificial neural network models for hour ahead electric load forecasting. *Procedia Computer Science*, 197, 16-24.

- Widrow, B. , L. M. A. 1995. *Perceptrons, Adalines and Backpropagation*.
- Zhang, G., Patuwo, B. E. & Hu, M. Y. 2017. A review on multivariate time series forecasting. *Applied Intelligence*, 46(3), 1-12.
- Zulinda.(2020). *Penerapan Jaringan Syaraf Tiruan Extreme Learning Machinde dan Backpropagation untuk Memprediksi Harga Saham PT BANK MANDIRI (PERSERO) TBK*. Skripsi. Yogyakarta: Progam Sarjana.

Lampiran 1. Data Statistik Kebutuhan Energi listrik Provinsi Jawa Tengah Tahun 2012 - 2023 (GWh)

Rincian	Tahun					
	2012	2013	2014	2015	2017	2018
Rumah Tangga	7898.11	8521.86	9301.29	9806.95	9717.11	10815.71
Industri	5739.43	6475.75	6898.15	6901.46	7223.07	8142.04
Komersial	1834.39	2006.97	2153.79	2339.49	2618.18	2907.20
Sosial	522.09	578.59	640.27	706.08	819.70	950.35
Gedung Pemerintah	168.2	183.45	197.04	208.52	227.56	260.05
Jalan Umum	439.2	438.46	440.93	445.69	451.42	482.68
Jumlah	16601.42	18205.08	19631.47	20408.19	21057.04	23558.03
	2019	2020	2021	2022	2023	
Rumah Tangga	11475.07	12556.06	12986.94	12961.71	12620.27	
Industri	8269.00	7592.91	8332.48	8778.3	8408.15	
Komersial	3187.26	3168.80	3512.97	3789.54	4170.67	
Sosial	1046.12	995.98	1055.15	1239.73	1418.82	
Gedung Pemerintah	282.44	282.37	284.09	304.65	324.66	
Jalan Umum	490.73	494.62	489.53	490.7	489.18	
Jumlah	24750.62	25090.74	26661.16	27564.63	27431.75	

Lampiran 2. Hasil Normalisasi

x_1	$=$	$\frac{0.8(7898.11-168.2)}{12986.94-168.2}$	$+ 0.1$	$=$	0.582413092
x_2	$=$	$\frac{0.8(8521.86-168.2)}{12986.94-168.2}$	$+ 0.1$	$=$	0.621340475
x_3	$=$	$\frac{0.8(9301.29-168.2)}{12986.94-168.2}$	$+ 0.1$	$=$	0.669983633
x_4	$=$	$\frac{0.8(9806.95-168.2)}{12986.94-168.2}$	$+ 0.1$	$=$	0.701541181
x_5	$=$	$\frac{0.8(9717.11-168.2)}{12986.94-168.2}$	$+ 0.1$	$=$	0.69593439
x_6	$=$	$\frac{0.8(10815.71-168.2)}{12986.94-168.2}$	$+ 0.1$	$=$	0.764496511
x_7	$=$	$\frac{0.8(11475.07-168.2)}{12986.94-168.2}$	$+ 0.1$	$=$	0.805646265
x_8	$=$	$\frac{0.8(12556.06-168.2)}{12986.94-168.2}$	$+ 0.1$	$=$	0.87310937
x_9	$=$	$\frac{0.8(12986.94-168.2)}{12986.94-168.2}$	$+ 0.1$	$=$	0.9
x_{10}	$=$	$\frac{0.8(12961.71-168.2)}{12986.94-168.2}$	$+ 0.1$	$=$	0.89842543
x_{11}	$=$	$\frac{0.8(12620.27-168.2)}{12986.94-168.2}$	$+ 0.1$	$=$	0.877116628
x_{12}	$=$	$\frac{0.8(5739.43-168.2)}{12986.94-168.2}$	$+ 0.1$	$=$	0.447692831
x_{13}	$=$	$\frac{0.8(6475.75-168.2)}{12986.94-168.2}$	$+ 0.1$	$=$	0.493645553
x_{14}	$=$	$\frac{0.8(6898.15-168.2)}{12986.94-168.2}$	$+ 0.1$	$=$	0.520006959
x_{15}	$=$	$\frac{0.8(6901.46-168.2)}{12986.94-168.2}$	$+ 0.1$	$=$	0.520213531
x_{16}	$=$	$\frac{0.8(7223.07-168.2)}{12986.94-168.2}$	$+ 0.1$	$=$	0.540284771
x_{17}	$=$	$\frac{0.8(8142.04-168.2)}{12986.94-168.2}$	$+ 0.1$	$=$	0.597636429
x_{18}	$=$	$\frac{0.8(8269.00-168.2)}{12986.94-168.2}$	$+ 0.1$	$=$	0.605559829
x_{19}	$=$	$\frac{0.8(7592.91-168.2)}{12986.94-168.2}$	$+ 0.1$	$=$	0.563365978
x_{20}	$=$	$\frac{0.8(8332.48-168.2)}{12986.94-168.2}$	$+ 0.1$	$=$	0.609521529
x_{21}	$=$	$\frac{0.8(8778.3-168.2)}{12986.94-168.2}$	$+ 0.1$	$=$	0.637344544
x_{22}	$=$	$\frac{0.8(8408.15-168.2)}{12986.94-168.2}$	$+ 0.1$	$=$	0.61424399
x_{23}	$=$	$\frac{0.8(1834.39-168.2)}{12986.94-168.2}$	$+ 0.1$	$=$	0.203984635
x_{24}	$=$	$\frac{0.8(2006.97-168.2)}{12986.94-168.2}$	$+ 0.1$	$=$	0.214755116

$$\begin{aligned}
x_{25} &= \frac{0.8(2153.79-168.2)}{12986.94-168.2} + 0.1 = 0.223917951 \\
x_{26} &= \frac{0.8(2339.49-168.2)}{12986.94-168.2} + 0.1 = 0.235507234 \\
x_{27} &= \frac{0.8(2618.18-168.2)}{12986.94-168.2} + 0.1 = 0.252899895 \\
x_{28} &= \frac{0.8(2907.20-168.2)}{12986.94-168.2} + 0.1 = 0.270937237 \\
x_{29} &= \frac{0.8(3187.26-168.2)}{12986.94-168.2} + 0.1 = 0.288415398 \\
x_{30} &= \frac{0.8(3168.80-168.2)}{12986.94-168.2} + 0.1 = 0.287263335 \\
x_{31} &= \frac{0.8(3512.97-168.2)}{12986.94-168.2} + 0.1 = 0.308742513 \\
x_{32} &= \frac{0.8(3789.54-168.2)}{12986.94-168.2} + 0.1 = 0.326002868 \\
x_{33} &= \frac{0.8(4170.67-168.2)}{12986.94-168.2} + 0.1 = 0.349788669 \\
x_{34} &= \frac{0.8(522.09-168.2)}{12986.94-168.2} + 0.1 = 0.12208579 \\
x_{35} &= \frac{0.8(578.59-168.2)}{12986.94-168.2} + 0.1 = 0.125611878 \\
x_{36} &= \frac{0.8(640.27-168.2)}{12986.94-168.2} + 0.1 = 0.129461242 \\
x_{37} &= \frac{0.8(706.08-168.2)}{12986.94-168.2} + 0.1 = 0.133568354 \\
x_{38} &= \frac{0.8(819.70-168.2)}{12986.94-168.2} + 0.1 = 0.140659222 \\
x_{39} &= \frac{0.8(950.35-168.2)}{12986.94-168.2} + 0.1 = 0.14881291 \\
x_{40} &= \frac{0.8(1046.12-168.2)}{12986.94-168.2} + 0.1 = 0.154789784 \\
x_{41} &= \frac{0.8(995.98-168.2)}{12986.94-168.2} + 0.1 = 0.151660616 \\
x_{42} &= \frac{0.8(1055.15-168.2)}{12986.94-168.2} + 0.1 = 0.155353334 \\
x_{43} &= \frac{0.8(1239.73-168.2)}{12986.94-168.2} + 0.1 = 0.166872719 \\
x_{44} &= \frac{0.8(1418.82-168.2)}{12986.94-168.2} + 0.1 = 0.178049481 \\
x_{45} &= \frac{0.8(168.2-168.2)}{12986.94-168.2} + 0.1 = 0.1 \\
x_{46} &= \frac{0.8(183.45-168.2)}{12986.94-168.2} + 0.1 = 0.100951732 \\
x_{47} &= \frac{0.8(197.04-168.2)}{12986.94-168.2} + 0.1 = 0.101799865 \\
x_{48} &= \frac{0.8(208.52-168.2)}{12986.94-168.2} + 0.1 = 0.102516316 \\
x_{49} &= \frac{0.8(227.56-168.2)}{12986.94-168.2} + 0.1 = 0.103704576 \\
x_{50} &= \frac{0.8(260.05-168.2)}{12986.94-168.2} + 0.1 = 0.105732233
\end{aligned}$$

$$\begin{aligned}
x_{51} &= \frac{0.8(282.44-168.2)}{12986.94-168.2} + 0.1 = 0.107129562 \\
x_{52} &= \frac{0.8(282.37-168.2)}{12986.94-168.2} + 0.1 = 0.107125193 \\
x_{53} &= \frac{0.8(284.09-168.2)}{12986.94-168.2} + 0.1 = 0.107125193 \\
x_{54} &= \frac{0.8(304.65-168.2)}{12986.94-168.2} + 0.1 = 0.107125193 \\
x_{55} &= \frac{0.8(324.66-168.2)}{12986.94-168.2} + 0.1 = 0.107125193 \\
x_{56} &= \frac{0.8(439.2-168.2)}{12986.94-168.2} + 0.1 = 0.107125193 \\
x_{57} &= \frac{0.8(438.46-168.2)}{12986.94-168.2} + 0.1 = 0.107125193 \\
x_{58} &= \frac{0.8(440.93-168.2)}{12986.94-168.2} + 0.1 = 0.107125193 \\
x_{59} &= \frac{0.8(445.69-168.2)}{12986.94-168.2} + 0.1 = 0.107125193 \\
x_{60} &= \frac{0.8(451.42-168.2)}{12986.94-168.2} + 0.1 = 0.107125193 \\
x_{61} &= \frac{0.8(482.68-168.2)}{12986.94-168.2} + 0.1 = 0.107125193 \\
x_{62} &= \frac{0.8(490.73-168.2)}{12986.94-168.2} + 0.1 = 0.107125193 \\
x_{63} &= \frac{0.8(494.62-168.2)}{12986.94-168.2} + 0.1 = 0.107125193 \\
x_{64} &= \frac{0.8(489.53-168.2)}{12986.94-168.2} + 0.1 = 0.107125193 \\
x_{65} &= \frac{0.8(490.7-168.2)}{12986.94-168.2} + 0.1 = 0.107125193 \\
x_{66} &= \frac{0.8(489.18-168.2)}{12986.94-168.2} + 0.1 = 0.107125193
\end{aligned}$$

Tabel 0.1. Hasil Normalisasi Data Kebutuhan Energi Listrik per Sektor Tahun 2021-2023

Tahun	A	B	C	D	E	F
2012	0.582413092	0.447692831	0.203984635	0.12208579	0.1	0.116912739
2013	0.621340475	0.493645553	0.214755116	0.125611878	0.100951732	0.116866556
2014	0.669983633	0.520006959	0.223917951	0.129461242	0.101799865	0.117020706
2015	0.701541181	0.520213531	0.235507234	0.133568354	0.102516316	0.117317771
2017	0.69593439	0.540284771	0.252899895	0.140659222	0.103704576	0.117675372
2018	0.764496511	0.597636429	0.270937237	0.14881291	0.105732233	0.119626266
2019	0.805646265	0.605559829	0.288415398	0.154789784	0.107129562	0.120128655
2020	0.87310937	0.563365978	0.287263335	0.151660616	0.107125193	0.120371425
2021	0.9	0.609521529	0.308742513	0.155353334	0.107232536	0.120053765
2022	0.89842543	0.637344544	0.326002868	0.166872719	0.108515658	0.120126783
2023	0.877116628	0.61424399	0.349788669	0.178049481	0.109764454	0.120031922

Keterangan :

A : Rumah Tangga

B : Industri

C : Komersial

D : Sosial

E : Gedung Pemerintah

F : Penerangan Jalan Umum

Lampiran 3. Script Metode Adaline Dalam Bahasa Pemrograman Python

1. Script Untuk Uji Akurasi

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split

# ===== DATA =====
X_all = pd.read_excel("input 2012-2023.xlsx").to_numpy()
Y_all = pd.read_excel("Target 2012-2023.xlsx").to_numpy()

# ===== SPLIT =====
# Tentukan jumlah data testing (misalnya 20%)
n_total = len(X_all)
n_test = int(0.3 * n_total) # 30% terakhir untuk testing
n_train = n_total - n_test # sisanya untuk training

# Bagi data
X_train = X_all[:n_train]
Y_train = Y_all[:n_train]

X_test = X_all[n_train:]
Y_test = Y_all[n_train:]

# ===== PARAMETER =====
n_input = 6
n_output = 6
alpha = 0.01
epochs = 10000

# Inisialisasi bobot dan bias (semua nol)
weights = [[0.0 for _ in range(n_input)]
            for _ in range(n_output)]
biases = [0.0 for _ in range(n_output)]
```

```

# ===== TRAINING =====
for epoch in range(epochs):
    for i in range(len(X_train)):
        inputs = X_train[i]
        targets = Y_train[i]
        outputs = []

        for j in range(n_output):
            y_in = 0.0
            for k in range(n_input):
                y_in += weights[j][k] * inputs[k]
            y_in += biases[j]
            outputs.append(y_in)

        # Update bobot dan bias
        for j in range(n_output):
            error = targets[j] - outputs[j]
            for k in range(n_input):
                weights[j][k] += alpha * error * inputs[k]
            biases[j] += alpha * error

# ===== EVALUASI TRAINING =====
print("=== Evaluasi pada Data Training ===")
mse_train = 0.0
mape_train = 0.0
n_train = len(X_train)

for i in range(n_train):
    inputs = X_train[i]
    actuals = Y_train[i]
    preds = []

    for j in range(n_output):
        y_in = 0.0
        for k in range(n_input):

```

```

        y_in += weights[j][k] * inputs[k]
    y_in += biases[j]
    preds.append(y_in)

    for j in range(n_output):
        error = actuals[j] - preds[j]
        mse_train += error ** 2
        if actuals[j] != 0:
            mape_train += abs(error / actuals[j])

total_train_points = n_train * n_output
mse_train /= total_train_points
mape_train = (mape_train / total_train_points) * 100

print(f"MSE Training: {mse_train:.6f}")
print(f"MAPE Training: {mape_train:.2f}%")

# ===== PREDIKSI & EVALUASI TESTING =====
print("\n=== Hasil Prediksi pada Data Testing ===")
mse_total = 0.0
mape_total = 0.0
n_data = len(X_test)

for i in range(n_data):
    inputs = X_test[i]
    actuals = Y_test[i]
    preds = []

    for j in range(n_output):
        y_in = 0.0
        for k in range(n_input):
            y_in += weights[j][k] * inputs[k]
        y_in += biases[j]
        preds.append(y_in)

    print(f"\nData {i+1}")

```

```

print("  Prediksi:", [{":.6f}".format(p)
for p in preds])
print("  Aktual  :", [{":.6f}".format(a)
for a in actuals])

for j in range(n_output):
    error = actuals[j] - preds[j]
    mse_total += error ** 2
    if actuals[j] != 0:
        mape_total += abs(error / actuals[j])

# ===== METRIK TESTING =====
total_points = n_data * n_output
mse = mse_total / total_points
mape = (mape_total / total_points) * 100

print("\n=== Evaluasi Kinerja pada Data Testing ===")
print(f"MSE Testing: {mse:.6f}")
print(f"MAPE Testing: {mape:.2f}%\n")

```

2. Script Untuk Prediksi Tahun 2024-2028

```

# Parameter normalisasi (sesuai dengan data sebelumnya)
x_min = 168.2          # contoh nilai minimum asli dari data
x_max = 12986.94       # contoh nilai maksimum asli dari data

# Fungsi denormalisasi
def denormalisasi(x_norm, x_min, x_max):
    return ((x_norm - 0.1) / 0.8) * (x_max - x_min) + x_min

# ===== Prediksi Tahun 2024 2028 =====
print("\n=== Prediksi Tahun 2024 2028 ===")
tahun_awal = 2023
input_prediksi = X_all[-1] # input awal dari tahun 2023

for tahun in range(2024, 2029):
    hasil_prediksi = []

```

```

for j in range(n_output):
    y_in = 0.0
    for k in range(n_input):
        y_in += weights[j][k] * input_prediksi[k]
    y_in += biases[j]
    hasil_prediksi.append(y_in)

# Tampilkan prediksi dalam bentuk normalisasi dan
denormalisasi
pred_norm = ["{:.6f}".format(x) for x in hasil_prediksi]
pred_asli = ["{:.2f}".format(denormalisasi(x, x_min,
x_max)) for x in hasil_prediksi]

print(f"Tahun {tahun}:")
print("  Normalisasi :", pred_norm)
print("  Denormalisasi:", pred_asli)

input_prediksi = hasil_prediksi

```

Lampiran 4. Script Metode *Backpropagation-Fuzzy Clustering* Dalam Bahasa Pemrograman *Python*

1. Script Untuk Uji Akurasi

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

np.random.seed(42)

# === Fungsi Aktivasi Sigmoid ===
def sigmoid(x):
    return 1 / (1 + np.exp(-x))

def sigmoid_derivative(x):
    return x * (1 - x)

# === Fuzzy C-Means Manual ===
def fuzzy_cmeans(X, c=3, m=2, max_iter=10000, epsilon=1e-5):
    N = X.shape[0]
    U = np.random.dirichlet(np.ones(c), size=N)

    for iteration in range(max_iter):
        U_m = U ** m
        centers = (U_m.T @ X) / np.sum(U_m.T, axis=1,
                                          keepdims=True)

        dist = np.zeros((N, c))
        for j in range(c):
            dist[:, j] = np.linalg.norm(X - centers[j],
                                         axis=1)

        dist = np.fmax(dist, 1e-6)
        U_new = np.zeros_like(U)
```

```

    for i in range(N):
        for j in range(c):
            denom = sum((dist[i, j] / dist[i, k]) **
                        (2 / (m - 1)) for k in range(c))
            U_new[i, j] = 1.0 / denom

        if np.linalg.norm(U_new - U) < epsilon:
            break
    U = U_new

    return U, centers

# === BACKPROPAGATION MANUAL MULTI-OUTPUT ===
def backpropagation(X, Y, hidden_neurons=8, epochs=10000,
                    lr=0.1):
    input_neurons = X.shape[1]
    output_neurons = Y.shape[1]

    W1 = np.random.randn(input_neurons, hidden_neurons)
    b1 = np.zeros((1, hidden_neurons))
    W2 = np.random.randn(hidden_neurons, output_neurons)
    b2 = np.zeros((1, output_neurons))

    losses = []

    for epoch in range(epochs):
        Z1 = X @ W1 + b1
        A1 = sigmoid(Z1)
        Z2 = A1 @ W2 + b2
        A2 = sigmoid(Z2)

        error = Y - A2
        loss = np.mean(error ** 2)
        losses.append(loss)

```

```

if loss < 1e-6:
    # Memakai 1e-6 atau lebih kecil untuk
    #menghasilkan output sangat presisi,
    break

    dZ2 = error * sigmoid_derivative(A2)
    dW2 = A1.T @ dZ2
    db2 = np.sum(dZ2, axis=0, keepdims=True)

    dA1 = dZ2 @ W2.T
    dZ1 = dA1 * sigmoid_derivative(A1)
    dW1 = X.T @ dZ1
    db1 = np.sum(dZ1, axis=0, keepdims=True)

    W2 += lr * dW2
    b2 += lr * db2
    W1 += lr * dW1
    b1 += lr * db1

return W1, b1, W2, b2, epoch + 1, losses

def predict(X, W1, b1, W2, b2):
    A1 = sigmoid(X @ W1 + b1)
    A2 = sigmoid(A1 @ W2 + b2)
    return A2

def compute_U_test(X_test, centers, m=2):
    N = X_test.shape[0]
    c = centers.shape[0]
    dist = np.zeros((N, c))

    for j in range(c):
        dist[:, j] = np.linalg.norm(X_test - centers[j],
            axis=1)
    dist = np.fmax(dist, 1e-6)

```

```

U_test = np.zeros((N, c))
for i in range(N):
    for j in range(c):
        denom = sum((dist[i, j] / dist[i, k]) **
                     (2 / (m - 1)) for k in range(c))
        U_test[i, j] = 1.0 / denom
return U_test

# ===== DATA =====
X_all = pd.read_excel("input 2012-2023.xlsx").to_numpy()
Y_all = pd.read_excel("Target 2012-2023.xlsx").to_numpy()

# ===== SPLIT =====
n_total = len(X_all)
n_test = int(0.3 * n_total)
n_train = n_total - n_test

X_train = X_all[:n_train]
Y_train = Y_all[:n_train]
X_test = X_all[n_train:]
Y_test = Y_all[n_train:]

# ===== FUZZY =====
U_train, centers = fuzzy_cmeans(X_train, c=3)
X_train_fuzzy = np.hstack([X_train, U_train])

U_test = compute_U_test(X_test, centers)
X_test_fuzzy = np.hstack([X_test, U_test])

# ===== TRAINING =====
W1, b1, W2, b2, trained_epochs, losses =
backpropagation(X_train_fuzzy, Y_train)
Y_train_pred = predict(X_train_fuzzy, W1, b1, W2, b2)
# ===== PREDIKSI =====
Y_pred = predict(X_test_fuzzy, W1, b1, W2, b2)

```

```

# ===== METRIK =====
mse_train = np.mean((Y_train - Y_train_pred) ** 2)
mape_train = np.mean(np.abs((Y_train - Y_train_pred)
/ Y_train)) * 100
mse = np.mean((Y_test - Y_pred) ** 2)
mape = np.mean(np.abs((Y_test - Y_pred) / Y_test)) * 100

print("\n=== HASIL AKHIR ===")
print("\n=== HASIL FUZZY CMEANS TRAINING ===")
print(U_train)
print(X_train_fuzzy)
print("\n=== HASIL FUZZY CMEANS TRAINING ===")
print(U_test)
print(X_test_fuzzy)
print(f"Epochs      : {trained_epochs}")
print(f"MSE Training : {mse_train:.6f}")
print(f"MAPE Training: {mape_train:.2f}%")
print(f"MSE Testing  : {mse:.6f}")
print(f"MAPE Testing : {mape:.2f}%\n")

print("=== Prediksi Semua Sektor (Data Testing) ===")
for i in range(len(Y_test)):
    print(f"\nData {i+1}")
    print("Target   :", [{":.5f}".format(val) for val in
Y_test[i]])
    print("Prediksi :", [{":.5f}".format(val) for val in
Y_pred[i]])
    print("Error    :", [{":.5f}".format(Y_test[i][j] -
Y_pred[i][j]) for j in range(Y_test.shape[1])])

```

2. Script Untuk Prediksi Tahun 2024-2028

```
# ===== PARAMETER DENORMALISASI =====
x_min = 168.2          # Sesuaikan nilai minimum dari data asli
x_max = 12986.94       # Sesuaikan nilai maksimum dari data asli

# Fungsi denormalisasi min-max skala [0.1, 0.9]
def denormalisasi(x_norm, x_min, x_max):
    return ((x_norm - 0.1) / 0.8) * (x_max - x_min) + x_min

print("=== Prediksi Tahun 2024 2028 ===")
tahun_awal = 2023
input_prediksi = X_all[-1]
# Data tahun 2023 sebagai input awal

hasil_prediksi_2024_2028 = []

for tahun in range(2024, 2029):
    # Hitung jarak dari input ke setiap pusat cluster
    dist = np.linalg.norm(input_prediksi - centers, axis=1)
    dist = np.fmax(dist, 1e-6) # Hindari pembagian nol

    # Hitung derajat keanggotaan fuzzy (U_pred)
    U_pred = np.array([
        1.0 / sum((dist[j] / dist[k]) ** (2)
        for k in range(len(centers)))
        for j in range(len(centers))
    ])
    U_pred = U_pred.reshape(1, -1)

    # Gabungkan input dengan hasil clustering fuzzy
    input_fuzzy = np.hstack([input_prediksi.reshape(1, -1),
    U_pred])

    # Prediksi menggunakan model backpropagation
    output_pred = predict(input_fuzzy, W1, b1, W2, b2).
    flatten()
```

```

# Simpan hasil prediksi
hasil_prediksi_2024_2028.append((tahun, output_pred))

# Tampilkan hasil: normalisasi dan denormalisasi
pred_norm = ["{:.5f}".format(val) for val in output_pred]
pred_denorm = ["{:.2f}".format(denormalisasi(val,
x_min, x_max)) for val in output_pred]

print(f"\nTahun {tahun}")
print("  Normalisasi      :", pred_norm)
print("  Denormalisasi     :", pred_denorm)

input_prediksi = output_pred

```

Lampiran 5. Hasil Denormalisasi

1. Denormalisasi Hasil *Adaline*

$$\begin{aligned}
 x_1 &= \frac{0.87152-0.1}{0.8} (12986.94 - 168.2) + 168.2 = 12530.53 \\
 x_2 &= \frac{0.91618-0.1}{0.8} (12986.94 - 168.2) + 168.2 = 13246.18 \\
 x_3 &= \frac{0.93500-0.1}{0.8} (12986.94 - 168.2) + 168.2 = 13547.79 \\
 x_4 &= \frac{0.59765-0.1}{0.8} (12986.94 - 168.2) + 168.2 = 8142.29 \\
 x_5 &= \frac{0.61660-0.1}{0.8} (12986.94 - 168.2) + 168.2 = 8445.97 \\
 x_6 &= \frac{0.62578-0.1}{0.8} (12986.94 - 168.2) + 168.2 = 8593.01 \\
 x_7 &= \frac{0.30105-0.1}{0.8} (12986.94 - 168.2) + 168.2 = 3389.71 \\
 x_8 &= \frac{0.31618-0.1}{0.8} (12986.94 - 168.2) + 168.2 = 3632.21 \\
 x_9 &= \frac{0.32248-0.1}{0.8} (12986.94 - 168.2) + 168.2 = 3733.06 \\
 x_{10} &= \frac{0.15828-0.1}{0.8} (12986.94 - 168.2) + 168.2 = 1101.98 \\
 x_{11} &= \frac{0.16403-0.1}{0.8} (12986.94 - 168.2) + 168.2 = 1194.23 \\
 x_{12} &= \frac{0.16649-0.1}{0.8} (12986.94 - 168.2) + 168.2 = 1233.61 \\
 x_{13} &= \frac{0.10819-0.1}{0.8} (12986.94 - 168.2) + 168.2 = 299.42 \\
 x_{14} &= \frac{0.10957-0.1}{0.8} (12986.94 - 168.2) + 168.2 = 321.59 \\
 x_{15} &= \frac{0.11020-0.1}{0.8} (12986.94 - 168.2) + 168.2 = 331.69 \\
 x_{16} &= \frac{0.12079-0.1}{0.8} (12986.94 - 168.2) + 168.2 = 501.25 \\
 x_{17} &= \frac{0.12767-0.1}{0.8} (12986.94 - 168.2) + 168.2 = 611.50 \\
 x_{18} &= \frac{0.12211-0.1}{0.8} (12986.94 - 168.2) + 168.2 = 522.48
 \end{aligned}$$

Tabel 0.2. Hasil Denormalisasi Metode *Adaline*

Tahun	A	B	C	D	E	F
2021*	12530.53	8142.29	3389.71	1101.98	299.42	501.25
2022*	13246.18	8445.97	3632.21	1194.23	321.59	611.50
2023*	13547.79	8593.01	3733.06	1233.61	331.69	522.48

2. Denormalisasi Hasil *Backpropagation-Fuzzy Clustering*

$$\begin{aligned}
 x_1 &= \frac{0.81988-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 11703.14 \\
 x_2 &= \frac{0.81762-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 11666.93 \\
 x_3 &= \frac{0.81633-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 11646.26 \\
 x_4 &= \frac{0.59865-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 8158.28 \\
 x_5 &= \frac{0.60041-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 8186.48 \\
 x_6 &= \frac{0.59919-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 8166.93 \\
 x_7 &= \frac{0.30370-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 3432.17 \\
 x_8 &= \frac{0.30646-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 3476.40 \\
 x_9 &= \frac{0.30383-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 3434.25 \\
 x_{10} &= \frac{0.16113-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 1147.71 \\
 x_{11} &= \frac{0.16415-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 1196.10 \\
 x_{12} &= \frac{0.16473-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 1205.40 \\
 x_{13} &= \frac{0.10693-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 279.24 \\
 x_{14} &= \frac{0.10782-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 293.50 \\
 x_{15} &= \frac{0.10801-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 296.55 \\
 x_{16} &= \frac{0.12215-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 523.12 \\
 x_{17} &= \frac{0.12160-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 514.31 \\
 x_{18} &= \frac{0.12072-0.1}{0.8}(12986.94 - 168.2) + 168.2 = 500.21
 \end{aligned}$$

Tabel 0.3. Hasil Denormalisasi Metode *Backpropagation-Fuzzy Clustering*

Tahun	A	B	C	D	E	F
2021*	11703.14	8158.28	3432.17	1147.71	279.24	523.12
2022*	11666.93	8186.48	3476.40	1196.10	293.50	514.31
2023*	11646.26	8166.93	3434.25	1205.40	296.55	500.21

RIWAYAT HIDUP

A. Identitas Diri

1. Nama : Alvian Rizky Fajar Anggara
2. Tempat & Tgl. Lahir : Klaten, 13 November 2001
3. Alamat Rumah : Kp. Cipayung, RT. 002/ RW. 007,
Kel. Tengah Kec. Cibinong, Kab. Bogor
4. Hp : 089508770059
5. Email : alfianrizkycell123@gmail.com

B. Riwayat Pendidikan

1. MI Nurul Ikhsan
2. MTs Negeri Cibinong
3. SMK Negeri 1 Cibinong

Semarang, 14 Febuari 2025

Alvian Rizky Fajar Anggara

NIM : 2008046030