DETEKSI KONSUMSI ENERGI PADA NAVIGASI CERDAS KENDARAAN LISTRIK MENGGUNAKAN MODEL JARINGAN SYARAF TIRUAN BERBASIS MULTILAYER PERCEPTRON

SKRIPSI

Oleh :
HUSNUL KHATIMAH
NIM. 220605110169

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025

DETEKSI KONSUMSI ENERGI PADA NAVIGASI CERDAS KENDARAAN LISTRIK MENGGUNAKAN MODEL JARINGAN SYARAF TIRUAN BERBASIS MULTILAYER PERCEPTRON

SKRIPSI

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh :
HUSNUL KHATIMAH
NIM. 220605110169

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2025

HALAMAN PERSETUJUAN

DETEKSI KONSUMSI ENERGI PADA NAVIGASI CERDAS KENDARAAN LISTRIK MENGGUNAKAN MODEL JARINGAN SYARAF TIRUAN BERBASIS MULTILAYER PERCEPTRON

SKRIPSI

Oleh :
HUSNUL KHATIMAH
NIM. 220605110169

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 3 Desember 2025

Pembimbing I,	Pembimbing II,
Okta Qomaruddin Aziz, M.Kom	Dr. Zainal Abidin, M.Kom
NIP. 19911019 201903 1 013	NIP. 19760613 200501 1 004

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Supriyono, M.Kom
NIP. 19841010 201903 1 012

HALAMAN PENGESAHAN

DETEKSI KONSUMSI ENERGI PADA NAVIGASI CERDAS KENDARAAN LISTRIK MENGGUNAKAN MODEL JARINGAN SYARAF TIRUAN BERBASIS MULTILAYER PERCEPTRON

SKRIPSI

Oleh :
HUSNUL KHATIMAH
NIM. 220605110169

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 24 Desember 2025

Susunan Dewan Penguji

Ketua Penguji	: Dr. Cahyo Crysdian, M.Cs NIP. 19740424 200901 1 008	( )
Anggota Penguji I	: Khadijah Fahmi Hayati Holle, M.Kom NIP. 19900626 202203 2 002	( )
Anggota Penguji II	: Okta Qomaruddin Aziz, M.Kom NIP. 19911019 201903 1 013	( )
Anggota Penguji III	: Dr. Zainal Abidin, M.Kom NIP. 19760613 200501 1 004	( )

Mengetahui dan Mengesahkan,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Supriyono, M.Kom
NIP. 19841010 201903 1 012

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama	: Husnul Khatimah
NIM	: 220605110169
Fakultas / Program Studi	: Sains dan Teknologi / Teknik Informatika
Judul Skripsi	: Deteksi Konsumsi Energi Pada Navigasi Cerdas Kendaraan Listrik Menggunakan Model Jaringan Syaraf Tiruan Berbasis Multilayer Perceptron

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 24 Desember 2025
Yang membuat pernyataan,

Husnul Khatimah
NIM 220605110169

MOTTO

اِنَّ مَعَ الْعُسْرِ يُسْرًاۗ

"Sesungguhnya beserta kesulitan ada kemudahan."

HALAMAN PERSEMBAHAN

Alhamdulillahi rabbil alamin, segala puji bagi Allah Yang Maha Esa atas limpahan karunia dan hidayah-Nya sehingga penulis dapat merampungkan tugas akhir ini. Shalawat serta salam senantiasa tercurah kepada Nabi Muhammad ﷺ, pembawa risalah kebenaran yang menuntun umat manusia dari kegelapan menuju cahaya Islam.

Atas izin Allah ﷻ, karya ini penulis persembahkan kepada:

1. Ayah Jhon Kenedi dan Mamak (Almh) Morina yang telah membimbing, mendoakan, dan menyayangi saya dari awal saya berada di dunia hingga saat ini. Walaupun harus melalui berbagai luka dan penderitaan, keduanya tetap berupaya memberikan yang terbaik demi masa depan saya. Pencapaian saya hingga titik ini tidak terlepas dari doa-doa mereka yang senantiasa mengiringi siang dan malam.
2. Bang Firqan, bang Uud, bang Fiqri, dan kak Hanim yang senantiasa memberikan dukungan, baik secara moral maupun material, serta menghadirkan semangat, nasihat, dan perhatian dalam setiap langkah yang saya tempuh. Perhatian, kasih sayang, perlindungan, dan teladan mereka senantiasa mengiringi perjalanan saya hingga berada di titik ini.
3. Bapak Okta Qomaruddin Aziz, M.Kom. atas segala ketulusan dan keikhlasan dalam membimbing saya menyelesaikan skripsi ini.

4. Bapak Ahmad Fahmi Karami, M.Kom., selaku Dosen Wali, atas ketulusan, kesabaran, dan keikhlasan dalam membimbing saya sejak awal perkuliahan hingga tahap penyelesaian studi.
5. Hilya, Atsila, Silvi, Adi, dan Singgi yang telah memberikan dukungan moral dan kebersamaan selama proses penyusunan skripsi ini.
6. Hindia, Feast, Nadin Amizah, serta musisi lainnya dalam playlist saya, yang melalui karya-karyanya senantiasa menemani, menenangkan, dan memberi ruang bernapas bagi saya selama mengerjakan skripsi ini,

   ”Kita hampir mati, dan kau selamatkan aku”
7. Dan terakhir, untuk diriku sendiri, yang telah melalui berbagai rintangan dan tetap memilih untuk bertahan hingga karya ini terselesaikan. Hiduplah lebih lama, iim. Meski tidak semua perjalanan berjalan tanpa luka, masih banyak keindahan yang menanti untuk disaksikan dan disyukuri.

Penulis juga menyampaikan terima kasih yang tulus kepada semua pihak yang telah memberikan bantuan, dukungan, dan doa selama proses pengerjaan skripsi ini. Semoga segala kebaikan yang telah diberikan mendapatkan balasan terbaik dari Allah Subhanahu Wa Ta’ala.

KATA PENGANTAR

Bismillahirrahmaanirrahiim, Assalamu'alaikum Warahmatullahi Wabarakatuh

Puji syukur senantiasa terucapkan kehadirat Allah Subhanahu Wa Ta'ala atas limpahan nikmat sehingga skripsi yang berjudul "Deteksi Konsumsi Energi Pada Navigasi Cerdas Kendaraan Listrik Menggunakan Model Jaringan Syaraf Tiruan Berbasis Multilayer Perceptron" ini dapat diselesaikan sebagai salah satu persyaratan untuk menyelesaikan studi. Shalawat serta salam selalu tercurahkan kepada Nabi Muhammad ﷺ. Semoga kelak mendapatkan syafaat beliau di *yaumul qiyamah*.

Penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dorongan, semangat, bantuan dan ide-ide kreatif sehingga tahap demi tahap penyusunan skripsi ini telah selesai. Ucapan terima kasih tersebut secara khusus disampaikan kepada:

1. 1. Prof. Dr. Hj. Ilfi Nur Diana, M.Si, selaku Rektor Universitas Islam Negeri Malang.
2. Dr. Agus Mulyono, M.Kes, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Malang.
3. Supriyono, M. Kom, selaku Kepala Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Okta Qomaruddin Aziz, M.Kom. selaku dosen pembimbing I dan Dr. Zainal Abidin M.Kom. selaku dosen pembimbing II yang telah memberikan bantuan dan arahan kepada penulis, sehingga bisa menuntaskan skripsi ini.

5. Dr. Cahyo Crysdian, M.Cs. selaku dosen penguji I dan Khadijah Fahmi Hayati Holle, M.Kom. selaku dosen penguji II yang telah menguji serta memberikan masukan sehingga penulis dapat menuntaskan skripsi dengan baik.
6. Bapak Ahmad Fahmi Karami, M.Kom. atas segala perhatian, ketulusan dan keikhlasan yang telah membimbing selama masa studi.
7. Seluruh dosen dan Jajaran Staf Program Studi Teknik Informatika, atas ilmu, bimbingan, dan bantuan administratif yang telah diberikan selama penulis menempuh studi.
8. Kedua orang tua dan kakak, Alfirqan Anshari, Alkhudri Anshari, Alfiqri Anshari, dan Husnul Hanim yang senantiasa mengiringi langkah penulis dengan cinta doa, nasihat, dan dukungan yang tulus, sehingga penulis dapat menyelesaikan studi ini.
9. Hilya Zahra Alifa, sahabat sejak awal perkuliahan, yang dengan ketulusan telah menemani perjalanan dan perjuangan penulis melalui berbagai suka dan duka, serta menghadirkan kehangatan melalui keluarga yang selalu menerima penulis dengan baik.
10. Sahabat-sahabat tersayang, Madhina, Wulan, Singgi, Silvi, Atsila, Adi, serta teman seperjuangan Gavrila dan Yoza, terima kasih atas kebersamaan, dukungan, dan berbagai bantuan yang diberikan selama ini. Kehadiran kalian menjadi penguat bagi penulis dalam menjalani kehidupan di perantauan dan proses penyelesaian studi.

11. Teman-teman “Monero”, “Pandora”, dan “Quantum Spark”, khususnya Kak Yoga, Kak Tiara, Fais, Rahma, Icha, Adam, dan Ubay, yang telah memberikan dukungan, serta bantuan sehingga proses perkuliahan dan penyusunan skripsi ini dapat berjalan dengan baik.
12. Teman-teman “Esperanza Cielo” yang turut memberikan pengalaman nyata dalam pengabdian kepada masyarakat serta semangat dan motivasinya selama pengerjaan tugas akhir ini.
13. Seluruh keluarga besar Teknik Informatika terkhusus Angkatan 2022 “INFINITY”, atas segala ilmu, semangat, dan pengalaman berharga yang telah dibagikan.
14. Seluruh keluarga besar, teman, sahabat, dan kerabat penulis yang tidak dapat disebutkan satu per satu, atas doa, dukungan, dan bantuan yang telah diberikan.

Akhir kata, penulis menyadari bahwa skripsi ini masih memiliki keterbatasan. Oleh karena itu, saran dan kritik yang membangun sangat diharapkan. Semoga skripsi ini dapat memberikan manfaat bagi pembaca serta berkontribusi dalam pengembangan ilmu pengetahuan di masa mendatang dan diterima sebagai amal ibadah di sisi Allah Subhanahu Wa Ta’ala

Wassalamu’alaikum warahmatullahi wabarakatuh.

Malang, 24 Desember 2025

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Khatimah, Husnul. 2025. Model Jaringan Syaraf Tiruan Berbasis Multi-Layer Perceptron untuk Deteksi Konsumsi Energi pada Navigasi Cerdas Kendaraan Listrik. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Okta Qomaruddin Aziz, M.Kom. (II) Dr. Zainal Abidin, M.Kom.

Kata Kunci: Kendaraan Listrik, Konsumsi Energi, Multi-Layer Perceptron, SMOTE.

Keterbatasan kapasitas baterai kendaraan listrik menuntut sistem navigasi cerdas mempertimbangkan estimasi energi sepanjang rute guna menjamin efisiensi operasional. Penelitian ini bertujuan mengevaluasi performa model Multi-Layer Perceptron (MLP) dalam mengklasifikasikan konsumsi energi rute ke dalam kategori penggunaan konsumsi energi 0% dan konsumsi energi 1%. Data bersumber dari 159 perjalanan di wilayah Malang dengan lima fitur: waktu tempuh, jarak, kecepatan maksimum, perubahan elevasi, dan kepadatan lalu lintas. Pra-pemrosesan data dilakukan melalui normalisasi Z-Score dan teknik Synthetic Minority Over-sampling Technique (SMOTE) untuk mengatasi ketidakseimbangan kelas. Pengujian melibatkan variasi arsitektur jaringan dan nilai threshold, yang dievaluasi berdasarkan metrik *Accuracy*, *Precision*, *Recall*, serta *F1-Score*. Hasil menunjukkan model mencapai *Accuracy* tertinggi 96,15% pada dataset asli, namun konfigurasi tersebut cenderung bias terhadap kelas mayoritas. Oleh karena itu, arsitektur MLP 5-8-4-1 dengan penerapan SMOTE pada threshold 0,50 ditetapkan sebagai model optimal karena memberikan keseimbangan performa terbaik dengan *Accuracy* 92,31% dan *F1-Score* 95,24%. Temuan ini membuktikan bahwa implementasi komputasi cerdas memberikan kontribusi signifikan pada manajemen energi kendaraan listrik yang lebih presisi, terukur, dan bertanggung jawab.

ABSTRACT

Khatimah, Husnul. 2025. Multilayer Perceptron–Based Artificial Neural Network Model for Energy Consumption Detection in Intelligent Electric Vehicle Navigation. Undergraduate Thesis. Informatics Engineering Study Program, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University of Malang. Advisors: (I) Okta Qomaruddin Aziz, M.Kom. (II) Dr. Zainal Abidin, M.Kom.

Keywords: Electric Vehicle, Energy Consumption, Multi-Layer Perceptron, SMOTE.

The limited battery capacity of electric vehicles requires navigation systems to consider not only distance and travel time, but also the energy demand along the route. This study aims to measure the performance of a Multilayer Perceptron (MLP) model in classifying electric vehicle route energy consumption into two classes, namely 0% and 1% consumption. The dataset consists of 160 electric vehicle trips in the Malang area with five main features: travel time, distance, maximum speed, elevation change, and traffic density. The data are normalized using Z-Score, split in a stratified manner into training, validation, and test sets, and balanced using SMOTE on the training set. Several experimental scenarios are constructed by varying the hidden layer architectures and *threshold* values, and are evaluated using accuracy, *Precision*, *Recall*, and *F1-Score*. The results show that the highest accuracy of 96.15% is obtained on the original imbalanced data; however, this configuration tends to favor the majority class. The MLP configuration with a 5–8–4–1 architecture on SMOTE-balanced data and a *threshold* of 0.50 is selected as the main model because it provides the best trade-off, achieving an accuracy of 92.31% and an *F1-Score* of 95.24%.

الملخص

خاتمة، حسن. 2025. نموذج الشبكة العصبية الاصطناعية القائم على الإدراك المتعدد الطبقات للكشف عن استهلاك الطاقة في الملاحة الذكية للمركبات الكهربائية. بحث جامعي. قسم هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية بمالانج. المشرف: (1) أوكتا قمر الدين عزيز، الماجستير. (2) الدكتور زين العابدين، الماجستير.

الكلمات المفتاحية :المركبات الكهربائية، استهلاك الطاقة، الإدراك المتعدد الطبقات، تقنية ,SMOTE.

تتطلب سعة البطارية المحدودة في المركبات الكهربائية وجود أنظمة ملاحة ذكية تأخذ في الاعتبار تقدير استهلاك الطاقة لضمان كفاءة التشغيل. تهدف هذه الدراسة إلى تقييم أداء نموذج الشبكة العصبية الاصطناعية القائم على الإدراك المتعدد الطبقات (MLP) في تصنيف استهلاك الطاقة إلى فئتين: استهلاك الطاقة بنسبة 0% واستهلاك الطاقة بنسبة 1%. تم الحصول على البيانات من 159 رحلة في منطقة مالانج باستخدام خمس ميزات تشمل وقت السفر، والمسافة، والسرعة القصوى، وتغير الارتفاع، وكثافة المرور. تمت معالجة البيانات مسبقًا باستخدام توحيد الدرجة المعيارية (Z-Score) وتقنية الإفراط في أخذ عينات الأقلية الاصطناعية (SMOTE) لمعالجة عدم توازن الفئات. شملت الاختبارات تنويع هيكل الشبكة وقيم العتبة (Threshold) ، والتي تم تقييمها بناءً على مقاييس الدقة، والدقة المحددة، والاسترداد، ودرجة F1.تشير النتائج إلى أن النموذج حقق أعلى دقة بنسبة 96.15% على مجموعة البيانات الأصلية، ومع ذلك أظهر هذا التكوين انحيازًا نحو الفئة الكبرى. بناءً على ذلك، تم تحديد هيكل MLP 5-8-4-1 مع تطبيق تقنية SMOTE عند عتبة 0.50 كنموذج أمثل، حيث قدم أفضل توازن في الأداء بدقة بلغت 92.31% ودرجة F1 بلغت 95.24%. تثبت هذه النتائج أن تطبيق الحوسبة الذكية يساهم بشكل كبير في إدارة طاقة المركبات الكهربائية بطريقة أكثر دقة ومسؤولية.يه.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kendaraan Listrik (Electric Vehicle/EV) semakin menjadi pilihan utama dalam upaya mengurangi gas emisi karbon dan ketergantungan terhadap bahan bakar fosil. Fakta menunjukkan bahwa penggunaan kendaraan listrik mengalami perkembangan yang sangat cepat secara global, seiring dengan meningkatnya kesadaran terhadap lingkungan dan kebijakan pemerintah yang mendukung transportasi berkelanjutan. Menurut International Energy Agency (IEA), penjualan EV global meningkat dari 14 juta unit (2023) menjadi 17 juta unit (2024), tumbuh sekitar 25% *year-on-year*. China menyumbang lebih dari 60% penjualan, dan negara-negara maju seperti Norwegia telah mencatat pangsa pasar hingga 89%. Namun demikian, keterbatasan kapasitas baterai pada kendaraan listrik menjadi tantangan utama dalam pengoperasiannya. Hal ini berdampak pada perlunya strategi navigasi yang efisien dalam hal konsumsi energi untuk memperpanjang jarak tempuh dan mengoptimalkan penggunaan daya baterai.

Salah satu tantangan dalam sistem navigasi kendaraan listrik adalah bagaimana memilih jalur yang efisien dalam penggunaan energi (Baroughi dkk., 2023; Dania, 2024; Moulik dkk., 2025). Strategi navigasi tidak dapat hanya mengandalkan jarak atau waktu tempuh, tetapi juga perlu mempertimbangkan estimasi konsumsi energi secara *real-time* agar kendaraan tidak melalui rute yang menguras baterai secara signifikan. Beberapa penelitian menunjukkan bahwa

penambahan modul prediksi energi ke dalam sistem navigasi dapat membantu dalam pengambilan keputusan yang lebih hemat daya (Alateef & Thomas, 2023; Modi & Bhattacharya, 2022). Menurut Modi & Bhattacharya, 2022, konsumsi energi kendaraan dipengaruhi oleh berbagai faktor seperti waktu tempuh, kecepatan maksimum, perubahan elevasi, dan kepadatan lalu lintas, di mana keempat faktor ini memiliki hubungan yang tidak selalu linier terhadap jumlah energi yang digunakan. Waktu tempuh dan kecepatan maksimum, misalnya, berkaitan dengan durasi dan intensitas kendaraan bergerak. Perubahan elevasi, khususnya pada rute menanjak, menyebabkan kendaraan membutuhkan tenaga lebih besar dibandingkan pada jalan datar. Sementara itu, kondisi lalu lintas padat menyebabkan kendaraan sering berhenti dan kembali berakselerasi, yang turut meningkatkan konsumsi energi. Hubungan antar faktor-faktor tersebut bersifat kompleks dan saling memengaruhi, sehingga tidak dapat dimodelkan secara optimal oleh regresi linear atau metode statistik konvensional.

Solusi terhadap permasalahan konsumsi energi kendaraan listrik telah ditawarkan oleh sejumlah penelitian dalam beberapa tahun terakhir. Zhang dkk., 2024 mengembangkan model prediksi konsumsi energi bus listrik menggunakan data waktu nyata dari parameter rute, lingkungan, dan kendaraan. Pendekatan ini menunjukkan efektivitas integrasi big data dalam pemodelan konsumsi energi dan berhasil menurunkan tingkat kesalahan estimasi, meskipun masih bergantung pada ketersediaan data dalam jumlah besar. Chen dkk., 2024 mengusulkan pendekatan berbasis data-driven probabilistik yang menggabungkan pengenalan kondisi berkendara dan pemodelan berbasis fragmen kinetik. Model ini menunjukkan

peningkatan *Accuracy* estimasi terutama dalam kondisi lalu lintas kompleks, tetapi tetap menghasilkan *output* dalam bentuk numerik. Sementara itu, EV-PINN yang dikembangkan oleh (Lim dkk., 2024) memanfaatkan pendekatan Physics-Informed Neural Network untuk memprediksi konsumsi energi berdasarkan kecepatan dan waktu secara efisien tanpa memerlukan sensor tambahan. Model ini mencapai performa tinggi dengan nilai *error* sangat rendah , namun fokus utamanya masih pada prediksi kontinu.

Secara umum, hasil dari penelitian-penelitian tersebut telah menunjukkan performa yang baik dalam memprediksi konsumsi energi kendaraan listrik. Namun demikian, sebagian besar masih berorientasi pada estimasi nilai numerik (regresi) dan belum diarahkan pada penyederhanaan *output* untuk mendukung pengambilan keputusan secara langsung dalam sistem navigasi cerdas. Mengingat kebutuhan sistem navigasi yang bekerja secara *real-time*, penyajian hasil prediksi dalam bentuk yang lebih sederhana dan aplikatif seperti pengelompokan “hemat” atau “boros” energi menjadi nilai tambah. Hal ini juga relevan dengan karakteristik data konsumsi energi yang digunakan, di mana nilai-nilainya terbatas pada angka-angka diskrit seperti 0, 1, atau -0.5 dan 0.5, bukan dalam bentuk nilai kontinu. Kondisi ini menunjukkan bahwa hasil konsumsi energi dalam perjalanan pendek atau segmentasi tertentu tidak selalu berubah secara signifikan, sehingga lebih tepat jika ditangani sebagai klasifikasi ketimbang regresi (Ausri & Bigazzi, 2024). Oleh karena itu, pendekatan klasifikasi yang mengelompokkan tingkat konsumsi ke dalam kategori tertentu dinilai lebih praktis dan sesuai dengan kebutuhan sistem navigasi yang responsif terhadap perubahan kondisi rute secara langsung.

Konsumsi energi kendaraan listrik dipengaruhi oleh berbagai faktor seperti waktu tempuh, kecepatan maksimum, perubahan elevasi, dan kepadatan lalu lintas. Hubungan antar faktor tersebut bersifat kompleks dan tidak selalu linier, sehingga sulit ditangani dengan metode prediksi sederhana seperti regresi linear (Alateef & Thomas, 2023; Lim dkk., 2024). Untuk itu, dibutuhkan pendekatan yang mampu mengenali pola-pola kompleks dan menangani interaksi antar fitur yang tidak terstruktur, tanpa bergantung pada asumsi distribusi data tertentu. Jaringan Syaraf Tiruan (JST) menjadi salah satu pendekatan yang sesuai karena mampu mempelajari pola hubungan non-linear antar fitur melalui proses pembelajaran yang adaptif. JST juga fleksibel dalam menghadapi jenis data yang bervariasi dan dapat digunakan untuk berbagai tujuan prediksi. Dalam penelitian ini, digunakan arsitektur Multi-Layer Perceptron (MLP) karena bentuknya yang sederhana namun efektif untuk data tabular (Ikwen dkk., 2024). Selain itu, MLP dapat memberikan hasil prediksi yang cepat dan stabil, menjadikannya cocok untuk konteks prediksi dua kelas seperti “konsumsi 0%” dan “konsumsi 1%” konsumsi energi dalam sistem navigasi kendaraan listrik.

Dari berbagai jenis arsitektur JST yang tersedia, MLP dianggap paling sesuai untuk data tabular dan non-sekuensial. Tidak seperti Convolutional Neural Network (CNN) yang digunakan untuk data visual, atau Recurrent Neural Network (RNN) yang digunakan untuk data urut waktu, MLP dapat memproses fitur statis secara langsung dan efisien (Pauli dkk., 2021). Selain itu, arsitektur MLP relatif ringan dan mudah diimplementasikan, sehingga cocok untuk diterapkan dalam sistem navigasi cerdas yang membutuhkan hasil prediksi cepat dan stabil.

Efektivitas MLP dalam konteks estimasi konsumsi energi juga ditunjukkan oleh (Chen dkk., 2024), yang menerapkan berbagai arsitektur JST, termasuk MLP, dalam pendekatan probabilistik berbasis kondisi berkendara. Penelitian tersebut menunjukkan bahwa MLP memberikan performa yang stabil dalam kondisi lalu lintas yang dinamis, serta mampu menangani input multivariabel yang kompleks tanpa memerlukan struktur model yang berat. Dengan mempertimbangkan karakteristik data dan kebutuhan sistem yang responsif, MLP dinilai sebagai pendekatan yang tepat dan praktis untuk digunakan dalam penelitian ini.

Evaluasi performa model merupakan tahap penting dalam pengembangan sistem berbasis *machine learning*, terutama ketika model digunakan sebagai dasar pengambilan keputusan dalam konteks yang sensitif, seperti sistem navigasi cerdas kendaraan listrik. Ketepatan prediksi terhadap kondisi yang berpotensi menyebabkan konsumsi energi tinggi atau rendah akan berdampak langsung pada efisiensi daya, jarak tempuh, dan efektivitas sistem. Tanpa proses evaluasi yang menyeluruh, model dapat menghasilkan prediksi yang bias, khususnya ketika distribusi data tidak seimbang. Oleh karena itu, pemilihan metrik evaluasi seperti *Precision*, *Recall*, dan *F1-Score* menjadi krusial untuk menilai performa model secara adil dan komprehensif (Riyanto dkk., 2023).

Penelitian ini akan menggunakan metrik evaluasi berupa *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. *Accuracy* mengukur proporsi prediksi yang benar dari seluruh data. *Precision* mengukur ketepatan model dalam mengidentifikasi kasus konsumsi energi tinggi, sedangkan *Recall* mengukur sejauh mana model mampu mendeteksi seluruh kasus tersebut. *F1-Score* digunakan sebagai metrik

gabungan yang memperhatikan keseimbangan antara *Precision* dan *Recall*, terutama penting saat distribusi data tidak merata.

Pentingnya pemilihan metrik evaluasi juga ditegaskan dalam beberapa studi terkini yang meneliti konsumsi energi pada kendaraan listrik. Studi oleh (Alateef & Thomas, 2023) menunjukkan bahwa *Accuracy* saja tidak cukup untuk menilai keandalan model, karena model masih dapat mengabaikan kondisi boros energi jika data tidak seimbang. Hal ini menjadi krusial karena kesalahan dalam mengklasifikasikan rute yang mengonsumsi daya besar dapat berdampak langsung pada jarak tempuh kendaraan. (Chen dkk., 2024) menunjukkan bahwa *F1-Score* memberikan hasil yang lebih stabil dibanding *Accuracy* dalam memprediksi konsumsi energi kendaraan, terutama dalam skenario lalu lintas yang berubah-ubah.

Sementara itu, (Lim dkk., 2024) menekankan pentingnya *Recall* dalam konteks navigasi cerdas, karena kegagalan mendeteksi konsumsi energi tinggi dapat menyebabkan sistem menyarankan rute yang tampaknya efisien padahal menguras daya. *Precision* juga menjadi fokus utama dalam studi tersebut karena prediksi positif palsu (*False positive*) dapat menyebabkan sistem menghindari rute yang sebenarnya hemat energi. Penggunaan metrik seperti *Precision*, *Recall*, dan *F1-Score* dinilai lebih tepat dalam evaluasi performa model prediksi konsumsi energi karena mempertimbangkan dampak langsung terhadap efisiensi baterai dalam pengambilan keputusan sistem navigasi.

Dalam perspektif Islam, upaya mengelola energi dan menjaga keseimbangan sumber daya merupakan bentuk tanggung jawab manusia sebagai khalifah di bumi. Hal ini sejalan dengan prinsip identifikasi dan pengelompokan

berdasarkan karakteristik tertentu agar tercipta ketepatan dalam bertindak, sebagaimana diisyaratkan dalam Al-Qur'an Surah Al-Hujurat ayat 13:

يٰٓاَيُّهَا النَّاسُ اِنَّا خَلَقْنٰكُمْ مِّنْ ذَكَرٍ وَّاُنْثٰى وَجَعَلْنٰكُمْ شُعُوْبًا وَّقَبَاۤىِٕلَ لِتَعَارَفُوْا ۚ اِنَّ اَكْرَمَكُمْ عِنْدَ اللّٰهِ اَتْقٰىكُمْ ۗاِنَّ اللّٰهَ عَلِيْمٌ خَبِيْرٌ

"Wahai manusia, sesungguhnya Kami telah menciptakan kamu dari seorang laki-laki dan perempuan. Kemudian, Kami menjadikan kamu berbangsa-bangsa dan bersuku-suku agar kamu saling mengenal. Sesungguhnya yang paling mulia di antara kamu di sisi Allah adalah orang yang paling bertakwa. Sesungguhnya Allah Maha Mengetahui lagi Maha Teliti." (Q.S. Al-Hujurat [49]:13)

Ayat ini menegaskan pentingnya pengelolaan waktu dan sumber daya secara seimbang sebagai bentuk kesyukuran. Dalam konteks penelitian ini, pengembangan sistem navigasi kendaraan listrik yang lebih hemat energi dapat dipandang sebagai upaya memanfaatkan teknologi dan energi secara bertanggung jawab.

Dengan demikian, penerapan teknologi klasifikasi ini tidak hanya menjadi solusi teknis dalam pengembangan sistem navigasi cerdas, tetapi juga menjadi manifestasi dari nilai-nilai Islam dalam menjaga keseimbangan pemanfaatan sumber daya serta menghindari penggunaan yang melampaui batas (*israf*) melalui pengelolaan energi yang lebih terukur, teliti, dan bertanggung jawab.

1.2 Pernyataan Masalah

Bagaimana performa model MLP dalam mendeteksi konsumsi energi berdasarkan metrik evaluasi *Accuracy*, *Precision*, *Recall*, dan *F1-Score* dengan mempertimbangkan pengaruh arsitektur MLP, kondisi data, dan *threshold*?

1.3 Batasan Masalah

Penelitian ini dibatasi pada penggunaan dataset yang diperoleh dari penelitian Bapak Dr. M. Amin Hariyadi, M.T., dkk (Hariyadi dkk., 2025). Atribut yang digunakan adalah waktu tempuh (t), jarak tempuh (s), kecepatan maksimum (topv), perubahan elevasi (*delta altitude*), kepadatan lalu lintas (*crowd*), dan konsumsi energi (*consum*) dengan rute kendaraan di sekitar Malang.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini untuk mengukur performa MLP dalam melakukan deteksi konsumsi energi berdasarkan metrik *Accuracy*, *Precision*, *Recall*, dan *F1-Score*.

1.5 Manfaat Penelitian

Beberapa manfaat yang diharapkan dari penelitian ini kepada berbagai pihak, diantaranya sebagai berikut:

1. Bagi Pengguna/Pengemudi Kendaraan Listrik, penelitian ini diharapkan dapat memberikan informasi prediksi yang akurat sehingga memungkinkan memberi rute yang paling efisien, mengoptimalkan titik pengisian daya, dan mengurangi kekhawatiran terhadap kehabisan baterai (*range anxiety*).
2. Bagi Industri Otomotif dan Pengembang Teknologi, penelitian ini diharapkan dapat menjadi dasar untuk peningkatan dan penyempurnaan algoritma pada Baterai Management System (BMS) dan sistem navigasi cerdas kendaraan listrik, sehingga dapat menghasilkan produk yang lebih efisien dan kompetitif di pasar.

3. Bagi Kementerian/Badan yang Mengurus Lingkungan Hidup, penelitian ini diharapkan dapat mendukung upaya pengendalian emisi dan promosi teknologi ramah lingkungan dengan memberikan basis data dan model prediksi yang membuktikan efisiensi energi yang lebih tinggi pada kendaraan listrik. Hasil ini dapat digunakan sebagai landasan kuat dalam merumuskan regulasi dan insentif yang mendorong adopsi kendaraan listrik sebagai solusi transportasi berkelanjutan.

BAB II

STUDI PUSTAKA

2.1 Deteksi Konsumsi Energi Listrik

Kendaraan listrik (Electric Vehicle/EV) merupakan salah satu inovasi transportasi modern yang berperan penting dalam mendukung pengurangan emisi karbon dan ketergantungan terhadap bahan bakar fosil (Jiang dkk., 2024). EV menggunakan motor listrik yang ditenagai oleh baterai sebagai sumber energi utama, sehingga lebih ramah lingkungan dibandingkan kendaraan konvensional berbasis mesin pembakaran dalam (Ranjan dkk., 2022). Namun demikian, konsumsi energi EV masih menjadi isu penting karena dipengaruhi oleh banyak faktor, baik dari sisi teknis kendaraan maupun kondisi lingkungan (Marchiori, 2022). Akibatnya, sustem navigasi tidak boleh hanya mengandalkan jarak dan waktu perjalanan saja. Ia juga harus dapat memperkirakan kebutuhan energi rute sebelum merekomendasikan rute yang akan dilalui oleh pengemudi.

Pendekatan berbasis klasifikasi ini telah divalidasi oleh Moulik dkk., (2025). Dalam penelitiannya, mereka menggunakan model *Multilayer Perceptron* (MLP) untuk melakukan *Driving Pattern Recognition*. Sistem tersebut tidak berfokus pada prediksi angka tunggal, melainkan mengklasifikasikan input berkendara ke dalam kategori kondisi jalan dan kepadatan lalu lintas. Hasilnya menunjukkan *Accuracy* klasifikasi mencapai 97,45%, membuktikan bahwa MLP sangat efektif untuk memetakan karakteristik data EV yang kompleks menjadi kelas-kelas keputusan yang tegas. Hal ini menjadi landasan bahwa data konsumsi

energi juga dapat diperlakukan sebagai masalah klasifikasi untuk membedakan efisiensi.

Dukungan terhadap pendekatan klasifikasi juga diperkuat oleh Padmavathy dkk., (2023) melalui pengembangan sistem optimasi energi *SmartEV*. Kinerja sistem ini dievaluasi secara spesifik menggunakan metrik klasifikasi, yaitu *Accuracy* dan *Precision*, bukan metrik *error* regresi. Dengan pencapaian *Accuracy* hingga 98,21% dalam menentukan strategi energi, penelitian ini menegaskan bahwa permasalahan manajemen energi EV dapat diselesaikan dengan memisahkan data ke dalam kategori keputusan biner (kondisi optimal vs tidak optimal), yang selaras dengan konsep deteksi kelas energi.

Lebih lanjut, Chen dkk., (2024) menekankan bahwa estimasi energi yang akurat harus diawali dengan *driving condition recognition*. Mereka membuktikan bahwa mengenali label atau kategori kondisi berkendara adalah tahap krusial, karena setiap kondisi perjalanan memiliki karakteristik profil energi yang unik. Temuan ini menyiratkan bahwa pengelompokan kondisi (klasifikasi) adalah langkah fundamental dalam memahami perilaku konsumsi energi EV.

Sementara itu, pemilihan variabel input untuk model deteksi ini didasarkan pada validasi fisik dari penelitian terdahulu. Modi & Bhattacharya, (2022) memvalidasi bahwa fitur seperti kecepatan, perubahan elevasi, dan temperatur lingkungan adalah variabel diskriminan utama yang wajib ada dalam model deteksi energi. Hal ini diperkuat oleh Zhang dkk., (2024) yang menunjukkan bahwa hubungan kecepatan terhadap energi bersifat non-linear dan pengaruh temperatur membentuk kurva U (optimal pada 20–25 °C). Temuan ini menegaskan bahwa

penggunaan model jaringan saraf (*Neural Network*) lebih tepat digunakan dibandingkan regresi linier sederhana untuk menangkap kompleksitas fitur-fitur tersebut. Rangkuman penelitian terkait mengenai electric vehicle tersebut disajikan pada Tabel 2.1.

Dari Tabel 2.1, dapat disimpulkan bahwa tren penelitian kendaraan listrik (EV) secara umum kini bergerak melampaui sekadar estimasi numerik (regresi) menuju pendekatan berbasis klasifikasi dan pengenalan pola (*pattern recognition*). Pergeseran ini menegaskan bahwa sistem manajemen energi pada EV modern tidak hanya dituntut memiliki presisi hitungan teknis, tetapi juga kemampuan cerdas untuk mengidentifikasi tingkat konsumsi energi secara akurat untuk memudahkan pengguna. Dengan memanfaatkan kemampuan *machine learning* dalam mengenali karakteristik operasional kendaraan, data energi yang kompleks dapat diterjemahkan menjadi informasi status yang lebih intuitif, sehingga mendukung efisiensi dan pengambilan keputusan yang lebih baik dalam ekosistem kendaraan listrik.

Tabel 2.1 Penelitian mengenai Electric Vehicle

Referensi Penelitian	Input	Metode yang Digunakan	Hasil
A system for electric vehicle's energy-aware routing in a transportation network through real-time prediction of energy consumption (Modi & Bhattacharya, 2022)	Profil kecepatan masa depan hasil prediksi lalu lintas; fitur: kecepatan kendaraan, akselerasi, kecepatan angin, elevasi jalan, suhu lingkungan, tingkat pengisian baterai (SOC), beban bantu.	*Deep Auto-Encoder* digunakan untuk memprediksi kecepatan masa depan. Estimasi energi dilakukan dengan multi-channel CNN dan disempurnakan dengan Bagged Decision Tree. Sistem bersifat *real-time* serta independen dari parameter internal kendaraan.	Kinerja terbaik vs pembanding; MAPE minimum 1,57% untuk estimasi konsumsi. Dapat dipakai untuk energy-aware routing (pemilihan rute hemat energi).

Referensi Penelitian	Input	Metode yang Digunakan	Hasil
Optimized Energy Consumption of Electric Vehicles with Driving Pattern Recognition for Real Driving Scenarios (Moulik dkk., 2025)	Kecepatan rata-rata, akselerasi, jumlah berhenti.	Klasifikasi Pola Mengemudi (Pattern Recognition) berbasis MLP.	*Output*: Kelas Pola (Urban/Highway). *Accuracy*: 97,45%. Membuktikan MLP efektif membedakan kategori kondisi berkendara.
A Machine learning-Based Energy Optimization System for Electric Vehicles (Padmavathy dkk., 2023)	Pola mengemudi, lalu lintas, cuaca, baterai.	*Machine Learning untuk optimasi strategi (SmartEV).*	*Output*: Strategi Keputusan. *Accuracy*: 98,21%. Menggunakan metrik klasifikasi untuk mengukur ketepatan strategi energi.
Energy Consumption Estimation of Electric Vehicles for Future Driving Process Based on Driving Conditions Recognition (Chen dkk., 2024)	Kecepatan, percepatan, jarak, waktu, tegangan & arus baterai.	*Unsupervised Learning (SOM)* + Klasifikasi Kondisi.	*Output*: Label Kondisi (Driving Condition Labels). Validasi bahwa pengelompokan kondisi perjalanan adalah tahap krusial untuk *Accuracy* prediksi.
A system for electric vehicle's energy-aware routing (Modi & Bhattacharya, 2022)	Kecepatan, elevasi, suhu, SOC.	Deep Auto-Encoder (DAE) dan CNN.	*Output*: Estimasi Energi (MAPE 1.57%). Validasi bahwa Suhu & Elevasi adalah fitur diskriminan utama yang wajib ada dalam model deteksi energi.
Electric Vehicle Power Consumption Modelling Method Based on Improved ACO–SVR. (Zhang dkk., 2024)	Kecepatan, temperatur, arus, tegangan.	Support Vector Regression (SVR).	*Output*: Model Konsumsi Daya (R^2 = 0.892). Validasi bahwa hubungan kecepatan terhadap energi bersifat Non-Linear, sehingga cocok menggunakan Neural Network.

2.2 Multilayer Perceptron (MLP)

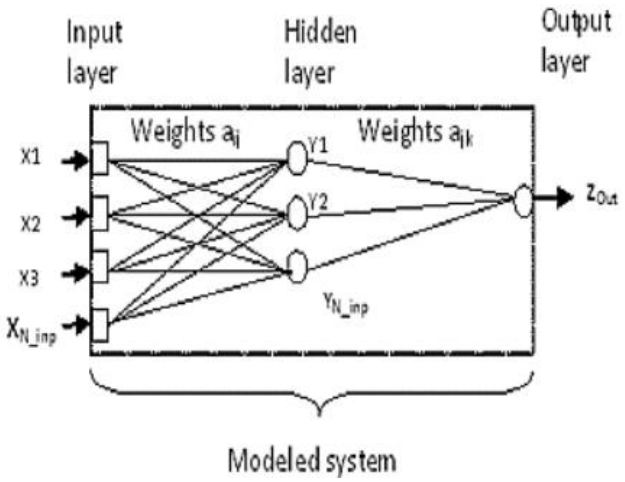
Jaringan Syaraf Tiruan atau yang biasa dikenal dengan Artificial Neural Network (ANN) merupakan metode *machine learning* yang terkenal dan sering digunakan dalam berbagai penelitian. ANN terinspirasi dari cara kerja otak manusia dan terdiri dari prosesor-prosesor seperti neuron biologis yang saling terhubung (Qamar & Zardari, 2023). ANN berfungsi untuk mempelajari hubungan kompleks antara input dan *output* sehingga dapat digunakan dalam tugas klasifikasi, prediksi, pengenalan pola, maupun regresi pada data yang bersifat non-linear ("*Application* ANN: *A Review*," 2018).

Salah satu artsitektur ANN yang populer dan banyak digunakan dalam penelitian *machine learning* adalah Multilayer Perceptron (MLP). MLP merupakan jenis jaringan syaraf feed-forward, artinya data hanya bergerak maju menuju dari lapisan input menuju lapisan tersembunyi (*hidden layer*), lalu lapisan *output* tanpa adanya umpan balik (*recurrent connections*) di dalam jaringan (Markov, 2023). Arsitektur MLP secara umum dapat dilihat pada gambar 2.1.

1. *Input Layer*, lapisan yang berperan sebagai pintu masuk data mentah yang akan diproses oleh model. Pada penelitian ini, fitur-fitur seperti jarak tempuh, waktu tempuh, kecepatan maksimum, perubahan elevasi dan kepadatan lalu lintas menjadi inputan utama (*variabel prediktor*). Seluruh data tersebut direpresentasikan dalam bentuk numerik yang akan diproses lebih lanjut oleh *hidden layer*.
2. *Hidden layer*, lapisan yang berada di antara input dan *output* yang berfungsi untuk mengekstraksi pola non-linier dan tidak terlihat secara langsung dari data.

Pada tahap ini, data diteruskan dari *input layer* dikalikan dengan bobot, ditambahkan bias, lalu dilewatkan ke dalam fungsi aktivasi non-linier. Proses ini memungkinkan jaringan memahami hubungan kompleks antar variabel yang tidak dapat ditangkap oleh metode.

3. *Output Layer*, tahapan terakhir dari pemrosesan data, dimana hasil yang diperoleh dari *hidden layer* diubah menjadi nilai prediksi. *Output layer* akan menghasilkan estimasi konsumsi energi kendaraan listrik tergolong konsumsi 0% atau konsumsi 1% berdasarkan variabel-variabel yang telah dimasukkan pada proses-proses sebelumnya.



Gambar 2.1 Arsitektur Model MLP 1 Layer
Sumber: (Gulo & Lubis, 2024)

Proses pembelajaran pada Multilayer Perceptron (MLP) dilakukan melalui dua tahap utama, yaitu *forward propagation* dan *backward propagation*. Pada tahap *forward propagation*, data inputan diproses secara berlapis mulai dari *input layer*, *hidden layer*, hingga *output layer* (Pauli dkk., 2021). Setiap neuron akan menghitung nilai berdasarkan kombinasi linier antara bobot (*weight*) dan bias, kemudian hasilnya dilewatkan ke fungsi aktivasi untuk menghasilkan *output* non-

linier. Setelah *output* akhir terbentuk, dilakukan perhitungan *error* dengan membandingkan hasil prediksi terhadap target.

Selanjutnya, pada tahap *backward propagation*, *error* yang dihasilkan akan dipropagasikan kembali ke lapisan sebelumnya untuk menghitung kontribusi masing-masing bobot terhadap kesalahan. Nilai gradien *error* ini digunakan untuk memperbarui bobot dengan algoritma optimasi, sehingga model dapat memperbaiki diri secara iteratif hingga mencapai *Accuracy* yang optimal.

Tahapan-tahapan tersebut dapat dijelaskan melalui beberapa rumus utama berikut:

1. Kombinasi Linear Input dan Bobot

$$z^{(l)} = W^{(l)}a^{(l-1)} + b^{(l)} \tag{2.1}$$

Keterangan:
$z^{(l)}$: input bersih (pre-activation) ke lapisan l
$W^{(l)}$: bobot antar neuron pada lapisan l
$a^{(l-1)}$: *output* dari lapisan sebelumnya
$b(l)$: bias pada lapisan l

Tahap pertama dalam forward propagation adalah menghitung kombinasi linear dari input, bobot, dan bias, sebagaimana dirumuskan pada Persamaan 2.1. Hasil perhitungan ini disebut sebagai *pre-activation* yang nantinya akan diproses oleh fungsi aktivasi.

2. Fungsi Aktivasi

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

$$f(x) = max(0, x) \tag{2.3}$$

Keterangan:
$f(x)$: fungsi aktivasi
x : input yang akan diproses

e : konstanta Euler (~2.718)

Fungsi aktivasi digunakan untuk menambahkan sifat non-linear ke dalam jaringan. Hal ini penting agar MLP dapat mengenali pola yang rumit, bukan hanya hubungan linier sederhana. Pada penelitian ini, fungsi aktivasi yang digunakan adalah fungsi *sigmoid* dan *Rectified Linear Unit* (ReLU), yang masing-masing dinyatakan pada Persamaan 2.2 dan Persamaan 2.3.

3. Aktivasi Neuron pada Tiap Layer

$$a^{(l)} = f\left(z^{(l)}\right) \tag{2.4}$$

Keterangan:
$a^{(l)}$: *output* lapisan l
$f\left(z^{(l)}\right)$: hasil transformasi non-linier dari input bersih

Nilai pre-activation $z^{(l)}$ kemudian dilewatkan ke fungsi aktivasi f. Hasilnya adalah *output* $a^{(l)}$, yaitu informasi yang siap diteruskan ke lapisan berikutnya, sebagaimana ditunjukkan pada Persamaan 2.4.

4. Fungsi *Loss* (Binary Cross-Entropy/BCE)

$$\mathrm{BCE}(y, \hat{p}) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\widehat{p_i}) + (1 - y_i)\log(1 - \widehat{p_i})] \tag{2.5}$$

Keterangan:
N : jumlah sampel
y_i : label sebenarnya untuk sampel ke-i
$\widehat{p_i}$: probabilitas prediksi model bahwa sampe ke-i termasuk Kelas 1 (konsumsi 1%)

Loss function digunakan untuk mengukur selisih antara hasil prediksi dan nilai target sebenarnya. Semakin kecil nilai *Loss*, semakin baik kinerja model. Pada penelitian ini, fungsi *Loss* yang digunakan adalah *Binary Cross-Entropy* (BCE) yang dirumuskan pada Persamaan 2.5.

5. Backpropagation (*Delta Rule*)

$$\delta^{(l)} = \left(W^{(l+1)}\right)^T \delta^{(l+1)} \odot f'\left(z^{(l)}\right) \tag{2.6}$$

Keterangan:
$\delta^{(l)}$: *error* pada lapisan l
$\left(W^{(l+1)}\right)^T$: transpose bobot lapisan berikutnya
$\delta^{(l+1)}$: *error* pada lapisan berikutnya
$f'\left(z^{(l)}\right)$: turunan fungsi aktivasi pada lapisan l
$\odot$: perkalian elemen demi elemen (*Hadamard product*)

Backpropagation menghitung seberapa besar kontribusi *error* dari tiap neuron. *Pada* tahap ini, *error* pada setiap lapisan dihitung secara mundur dari lapisan output menuju lapisan input menggunakan aturan delta (*delta rule*), sebagaimana ditunjukkan pada Persamaan 2.6. Dengan cara ini, bobot bisa diperbarui agar prediksi semakin mendekati target.

6. Pembaruan Bobot (*Gradient Descent*)

$$W^{(l)} = W^{(l)} - \eta \left(\frac{\partial J}{\partial W^{(l)}}\right) \tag{2.7}$$

Keterangan:
$W^{(l)}$: *error* pada lapisan l
η : *learning rate*
$\left(\frac{\partial J}{\partial W^{(l)}}\right)$: *error* pada lapisan berikutnya

Setelah *error* dihitung, langkah terakhir adalah memperbarui bobot agar *error* menurun, sebagaimana dirumuskan pada Persamaan 2.7. Proses ini dilakukan dengan algoritma *gradient descent*.

Sejumlah penelitian terdahulu telah menunjukkan efektivitas MLP dalam berbagai konteks, khususnya dalam bidang energi dan kendaraan listrik. Misalnya, penelitian yang dilakukan oleh Moulik dkk., (2025) melalui penelitian mengenai

Driving Pattern Recognition. Dalam studi ini, MLP digunakan untuk mengolah input dinamis seperti kecepatan rata-rata, akselerasi, dan jumlah berhenti guna mengklasifikasikan pola berkendara. Hasil penelitian mencatat *Accuracy* klasifikasi mencapai 97,45%, yang membuktikan bahwa MLP sangat andal dalam membedakan kategori data EV yang kompleks dan memisahkan pola berkendara ke dalam kelas-kelas yang spesifik.

Selain kemampuannya dalam klasifikasi, MLP juga sering digunakan sebagai standar atau *baseline* yang kuat dalam pengembangan model prediksi energi. Hal ini terlihat dalam penelitian Padmavathy dkk., (2023) yang mengembangkan sistem optimasi *SmartEV*. Meskipun penelitian tersebut mengusulkan metode optimasi baru, MLP (sebagai *Artificial Neural Network*) tetap menunjukkan kinerja prediksi yang kokoh sebagai model pembanding dengan *Accuracy* berkisar antara 81% hingga 85%. Hal ini menegaskan bahwa tanpa modifikasi yang rumit sekalipun, struktur dasar MLP sudah mampu menangkap pola konsumsi energi dengan cukup baik.

Posisi MLP sebagai metode yang kompetitif diperkuat oleh studi komparatif yang dilakukan oleh Duarte & Yoshizaki (2025). Mereka membandingkan kinerja MLP dengan berbagai algoritma *machine learning* modern seperti *Support Vector Regression* (SVR), *Gradient Boosting*, dan *Extra Trees* menggunakan dataset perjalanan EV. Meskipun metode *ensemble* seperti *Extra Trees* memberikan hasil terbaik dalam studi tersebut, MLP terbukti memberikan performa prediksi yang konsisten dan kompetitif dibandingkan metode regresi lainnya. Temuan ini

menyiratkan bahwa MLP tetap menjadi pilihan relevan karena keseimbangannya antara *Accuracy* dan efisiensi komputasi.

Meskipun penelitian-penelitian di atas menunjukkan keandalan MLP, terdapat celah penelitian (*research gap*) yang signifikan. Moulik dkk. (2025) berfokus pada klasifikasi pola mengemudi (*input*), bukan klasifikasi langsung terhadap status energi (*output*). Di sisi lain, Duarte & Yoshizaki (2025) berfokus pada prediksi nilai numerik yang sering kali sulit diterjemahkan secara cepat oleh pengemudi awam. Belum banyak penelitian yang secara spesifik memanfaatkan MLP untuk langsung mengklasifikasikan *output* energi menjadi label biner sebagai sistem pendukung keputusan yang sederhana namun akurat.

Tabel 2.2 Penelitian Mengenai Multi-Layer Perceptron

Referensi Penelitian	Input	Metode yang Digunakan	Hasil
Optimized Energy Consumption of Electric Vehicles with Driving Pattern Recognition (Moulik dkk., 2025)	Kecepatan rata-rata, akselerasi positif/negatif, jumlah berhenti.	Multilayer Perceptron (MLP) untuk klasifikasi pola.	MLP berhasil mengklasifikasikan pola berkendara dengan *Accuracy* 97,45%, membuktikan keandalannya dalam mengenali kategori data EV yang kompleks.
A Machine Learning-Based Energy Optimization System (Padmavathy dkk., 2023)	Pola mengemudi, kondisi lalu lintas, cuaca, status baterai.	Artificial Neural Network (MLP) sebagai model pembanding/baseline.	MLP menunjukkan kemampuan prediksi yang kuat sebagai model dasar (*Accuracy* ~81-85%) sebelum dilakukan optimasi lanjutan dengan metode SEV.
Comparative Study of Machine learning Models for EV Energy Consumption (Duarte & Yoshizaki, 2025)	Data perjalanan EV (Colorado dataset)	Multilayer Perceptron, Support Vector Regression, Gradient Boosting, XGBoost, Extra Trees	Extra Trees memberikan hasil terbaik, namun MLP terbukti memberikan performa prediksi yang kompetitif dan konsisten dibandingkan metode regresi lainnya.

Oleh karena itu, penelitian ini dilakukan untuk mengisi celah tersebut dengan mengadaptasi kekuatan klasifikasi MLP yang dibuktikan oleh Moulik dkk. (2025), namun diterapkan pada target *output* berupa status konsumsi energi. Pendekatan ini menawarkan solusi jalan tengah: memanfaatkan kemampuan non-linier MLP (seperti yang divalidasi Duarte & Yoshizaki (2025)) tetapi menyajikan hasil yang lebih intuitif bagi pengguna (seperti filosofi *decision support* pada Padmavathy dkk.), sehingga sistem navigasi tidak hanya cerdas secara matematis tetapi juga fungsional secara operasional.

Berdasarkan literatur pada Tabel 2.2, disimpulkan bahwa MLP memiliki peran strategis sebagai *universal approximator* yang andal. Penelitian ini akan mengoptimalkan potensi tersebut dengan menggeser paradigma penggunaan MLP dari sekadar prediksi angka (regresi) atau pengenalan pola input, menjadi alat klasifikasi status energi yang berorientasi pada pengambilan keputusan navigasi.

BAB III

DESAIN DAN IMPLEMENTASI

3.1 Pengambilan Data

Data yang digunakan pada penelitian ini merupakan data sekunder. Data ini diambil dari penelitian oleh Hariyadi dkk. (2025). Dataset ini memiliki 159 data dan 6 atribut yang sejalan dengan analisis konsumsi energi kendaraan listrik. Pada penelitian ini digunakan 5 fitur dengan 1 atribut yaitu *consum* merupakan variabel *output* dalam penelitian ini. Atribut yang digunakan meliputi:

Tabel 3.1 Penjelasan Fitur

No.	Fitur	Simbol	Satuan	Tipe Data	Deskripsi singkat	Domain/ Nilai
1	Waktu tempuh	t	menit	Numerik kontinu	Lama perjalanan pada rute yang ditempuh	$t > 0$
2	Jarak tempuh	s	km	Numerik kontinu	Panjang rute dari titik awal ke tujuan	$s > 0$
3	Kecepatan maksimum	topv	km/ jam	Numerik kontinu	Kecepatan tertinggi yang tercapai pada rute	$topv \geq 0$
4	Perubahan elevasi	delta altitude	meter	Numerik kontinu	Selisih ketinggian (naik/turun) di sepanjang rute	Δelevasi merupakan nilai negatif, nol, nilai positif
5	Kepadatan lalu lintas	crowd	—	Kategorikal ordinal	Kondisi lalu lintas pada rute	Crowd = {0,1,2}
6	Konsumsi energi (label)	*consum*	—	Kategorikal biner (target)	Konsumsi energi yang terpakai	Consum = {0, 1}

Tabel 3.2 Contoh Dataset Konsumsi Energi Kendaraan Listrik

Waktu tempuh (menit)	Jarak tempuh (km)	Kecepatan Max (km/jam)	Perubahan elevasi (m)	Kepadatan Lalu Lintas	Konsumsi (%)
7	3	20	16	1	1
3	2	33	60	2	1
6	2	40	7	1	0
4	1	35	-46	1	0
3	1	35	92	2	1

Dari Tabel 3.1 dan 3.2 dapat dilihat bahwa dataset memiliki lima fitur input (t, s, topv, delta altitude, dan crowd) serta satu variabel *output*, yaitu konsumsi energi (*consum*). Data ini menjadi dasar utama dalam proses pelatihan model MLP untuk estimasi konsumsi energi kendaraan listrik. Setelah pengumpulan data, tahap berikutnya adalah *Exploratory Data Analysis* (EDA) untuk meninjau distribusi label, sebaran serta korelasi fitur.

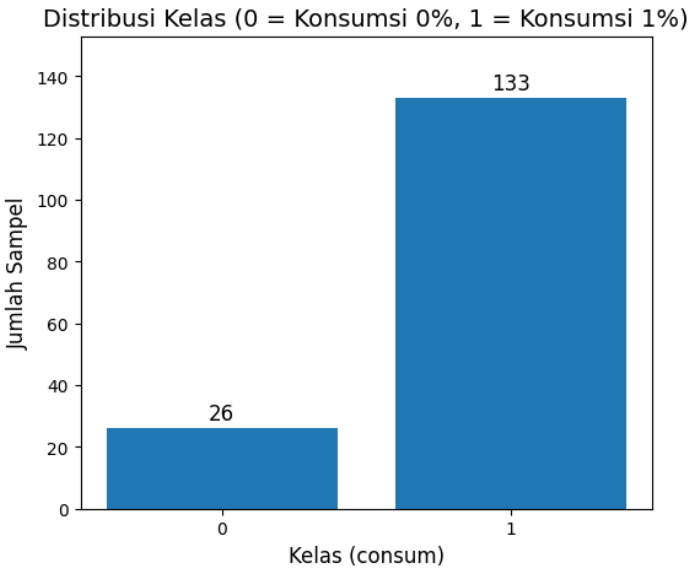
3.2 Exploratory Data Analysis (EDA)

3.2.1 Distribusi Label

Penetapan kategori variabel target *consum* (Kelas 0 dan Kelas 1) dilakukan secara manual dengan mengacu pada panel navigasi cerdas mobil listrik. Dari total 159 sampel, Kelas 1 (konsumsi 1%) berjumlah sekitar 133 sampel, sedangkan Kelas 0 (konsumsi 0%) berjumlah sekitar 26 sampel. Hal ini menunjukkan bahwa adanya ketidakseimbangan kelas (*class Imbalanced*) yang signifikan pada variabel target (label) *consum* seperti pada Gambar 3.1.

Ketidakseimbangan ini mengindikasikan bahwa *Accuracy* saja berpotensi menyesatkan (*misleading*) karena model dapat mencapai nilai *Accuracy* tinggi

dengan lebih sering memprediksi kelas mayoritas. Oleh karena itu, pemodelan akan menggunakan *stratified split* (*train*/*Validation*/*test*) agar proporsi kelas terjaga. Kinerja model akan dievaluasi menggunakan metrik *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. Selain itu, untuk mengurangi bias model terhadap kelas mayoritas, dilakukan penyesuaian ambang batas keputusan (*Threshold tuning*) dan/atau penerapan teknik *oversampling* SMOTE (*Synthetic Minority Over-sampling Technique*) pada data latih. Dengan demikian, implementasi strategi-strategi tersebut bertujuan agar hasil evaluasi yang diperoleh mampu memberikan penilaian yang objektif dan imparsial terhadap kinerja model dalam mengklasifikasikan kedua kelas.

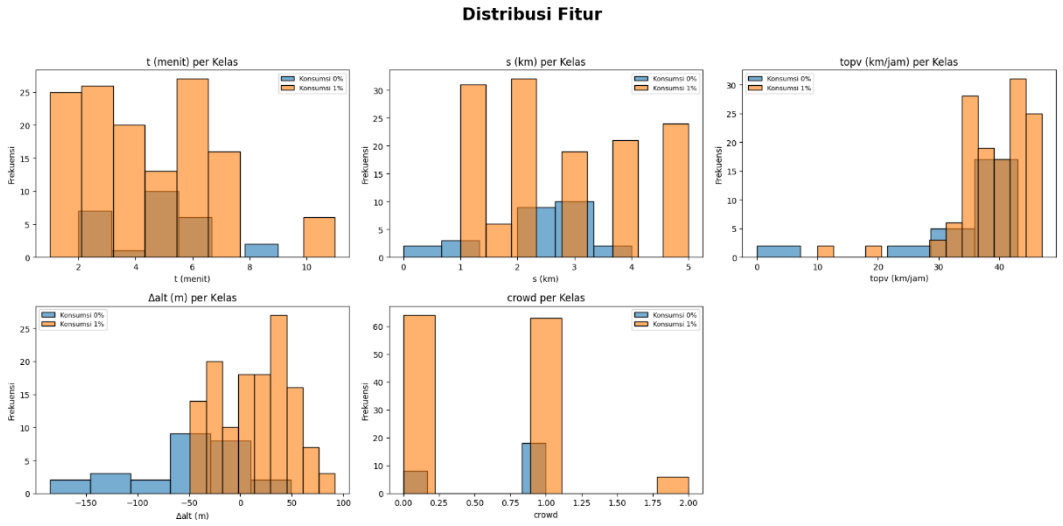


Gambar 3.1 Distribusi Label

3.2.2 Distribusi Fitur

Analisis distribusi fitur pada Gambar 3.2 dilakukan dengan membandingkan sebaran nilai untuk setiap variabel antara Kelas 0 (konsumsi 0%) dan Kelas 1 (konsumsi 1%) pada variabel target. Secara umum, distribusi fitur

kontinu (t, s, topv, Δalt) menunjukkan variasi yang signifikan dan tingkat tumpang tindih (*overlap*) yang tinggi antara Kelas 0 (konsumsi 0%) dan Kelas 1 (konsumsi 1%). Kondisi ini mengindikasikan bahwa kemampuan pemisah (*separability*) setiap fitur secara individual untuk membedakan kedua kelas adalah terbatas.



Gambar 3.2 Distribusi Fitur

Secara lebih spesifik, analisis menunjukkan bahwa fitur t (menit) dan s (km) memiliki konsentrasi sebaran yang signifikan pada rentang nilai rendah baik untuk Kelas 0 (konsumsi 0%) maupun Kelas 1 (konsumsi 1%). Berbeda halnya, fitur topv (kecepatan maksimum) memperlihatkan kecenderungan distribusi yang lebih menonjol pada nilai tinggi, terutama di antara sampel yang termasuk dalam Kelas 1 (konsumsi 1%). Fitur Δalt (perubahan elevasi) menampilkan rentang sebaran yang luas, mencakup nilai dari negatif hingga positif, yang mengindikasikan adanya variasi elevasi yang besar antar-sampel. Selain itu, pada fitur topv dan Δalt, teramati adanya kecondongan sebaran yang signifikan dan rentang nilai yang lebar. Sementara itu, fitur crowd yang merupakan variabel kategorikal ordinal,

menunjukkan konsentrasi sebaran yang kuat pada nilai diskrit 1.00 dan 1.25 untuk Kelas 1 (konsumsi 1%), dan kategori 0.00 serta 1.25 untuk Kelas 0 (konsumsi 0%).

Secara keseluruhan, karakteristik distribusi fitur ditandai dengan tumpang tindih yang tinggi dan kecondongan sebaran pada variabel kontinu, menegaskan pentingnya implementasi strategi pra-pemrosesan data yang cermat. Strategi ini mencakup penerapan Normalisasi Z-Score pada fitur kontinu untuk menyamakan skala dan mengurangi potensi bias dari sebaran data yang tidak seragam, di mana penyesuaian ini krusial demi tercapainya kualitas dan keandalan model yang optimal pada proses berikutnya.

3.2.3 Korelasi Antar Fitur

Analisis korelasi antar fitur dilakukan menggunakan koefisien Spearman's Rho pada Gambar 3.3 dan didukung oleh visualisasi Scatterplot Matrix pada Gambar 3.4 untuk mendeteksi hubungan monotonik antar variabel. Secara umum, hasil analisis statistik menunjukkan bahwa tidak ditemukan bukti adanya korelasi yang kuat antar fitur prediktor, sehingga mengindikasikan bahwa masalah multikolinearitas dapat dihindari. Koefisien Spearman's Rho yang mayoritas rendah menunjukkan bahwa sebagian besar fitur prediktor bersifat independen satu sama lain. Kondisi ini secara kuat mengindikasikan bahwa pola hubungan data bersifat non-linear, yang menjustifikasi penggunaan model yang mampu menangkap struktur kompleks.

Korelasi signifikan yang teridentifikasi berada pada tingkat sedang hingga rendah (*moderate to low*), dengan contoh utama adalah hubungan negatif antara fitur s (jarak) dengan crowd (kepadatan lalu lintas), dengan koefisien $\rho = -0.399$

dan p-value < 0.001. Temuan bahwa korelasi antar fitur yang tinggi tidak ditemukan memberikan keuntungan metodologis ganda: (1) masalah multikolinearitas dalam proses pemodelan dapat dihindari, dan (2) rendahnya hubungan monotonik antar fitur memberikan justifikasi kuat untuk memilih dan mengimplementasikan arsitektur model yang mampu menangkap pola non-linear yang kompleks.

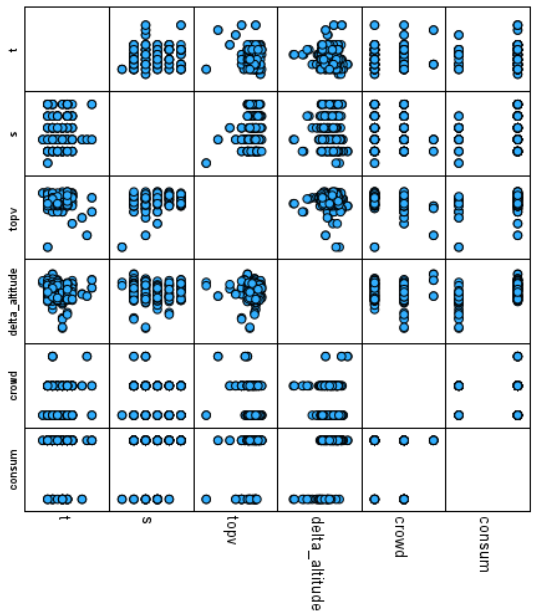
Correlations

			t	s	topv	delta_altitude	crowd
Spearman's rho	t	Correlation Coefficient	--				
		Sig. (2-tailed)	.				
		N	159				
	s	Correlation Coefficient	.136	--			
		Sig. (2-tailed)	.093	.			
		N	153	153			
	topv	Correlation Coefficient	-.036	.111	--		
		Sig. (2-tailed)	.651	.172	.		
		N	159	153	159		
	delta_altitude	Correlation Coefficient	-.164*	-.094	-.058	--	
		Sig. (2-tailed)	.039	.246	.467	.	
		N	159	153	159	159	
	crowd	Correlation Coefficient	.125	-.049	-.399***	-.051	--
		Sig. (2-tailed)	.116	.547	<.001	.521	.
		N	159	153	159	159	159

*. Correlation is significant at the 0.05 level (2-tailed).

***. Correlation is significant at the 0.001 level (2-tailed).

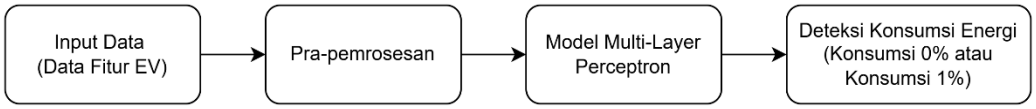
Gambar 3.3 Korelasi Antar Fitur



Gambar 3.4 Scatterplot Matrix Antar Fitur

3.3 Desain Sistem

Pada penelitian ini, Gambar 3.5 menampilkan tahapan dari sistem deteksi konsumsi energi pada navigasi cerdas kendaraan listrik. Tahap-tahapan tersebut, meliputi:



Gambar 3.5 Desain Penelitian

3.3.1 Pra-pemrosesan Data

Pra-pemrosesan data bertujuan untuk memastikan bahwa data yang digunakan merupakan data yang memiliki kualitas baik dan siap diproses dalam penelitian. Fokus pra-pemrosesan adalah menormalisasi fitur agar berada pada skala yang seragam serta pembagian dataset untuk evaluasi yang adil dan dapat direproduksi.

3.3.2 Normalisasi Data (Z-Score)

Tahapan pertama pada pra-pemrosesan data yaitu normalisasi atau standarisasi data, dimana proses normalisasi menggunakan Z-Score. Proses ini menormalkan data dengan cara mengurangkan rata-rata (*mean*) dan membaginya dengan simpangan baku (*standar deviation*). Pendekatan ini bertujuan untuk menyamakan skala antar fitur dan umumnya mempercepat serta menstabilkan proses pembelajaran jaringan syaraf (Faye dkk., 2023).

Normalisasi data bertujuan untuk menyamakan skala antar fitur setiap variabel input seragam dan proses pelatihan MLP lebih stabil. Hal ini perlu dilakukan agar fitur seperti jarak tempuh tidak mendominasi fitur lain seperti waktu

tempuh. Adapun contoh data sebelum dinormalisasi disajikan pada Tabel 3.3 dan data setelah dinormalisasi disajikan pada Tabel 3.4.

Tabel 3.3 Contoh data sebelum dinormalisasi

t	s	topv	delta altitude	crowd
7	3	20	16	1
3	2	35	36	1
5	2	40	38	1
6	1.5	37	47	1
3	1	35	92	2

Tabel 3.4 Contoh data ternormalisasi (Z-Score)

normt	norms	normtopv	normdelta	normcrowd
2	0.238788	−2.48563	0.273898	0.73579
−0.75671	−0.47757	−0.45680	0.701276	0.73579
0.183265	−0.47757	0.219471	0.744014	0.73579
0.653252	−0.83576	−0.18629	0.936333	0.73579
−0.75670	−1.19394	−0.45680	1.897933	2.508375

Hasil dari data ternormalisasi pada Tabel 3.4, diperoleh dari Persamaan 3.1 berikut:

$$x_{norm} = \frac{x-\mu}{\sigma} \tag{3.1}$$

3.3.3 Pembagian Dataset

Setelah data telah dinormalisasi, maka dilanjutkan dengan tahap pembagian dataset. Dataset dibagi menjadi tiga bagian, yaitu data pelatihan (*data Training*), data validasi (*data Validation*), dan data pengujian (*data test*). Pembagian data dilakukan untuk mengevaluasi serta mencegah kebocoran data (*data leakage*)

terhadap sistem. Pada penelitian ini, data akan dibagi dengan 70% data dialokasikan untuk data pelatihan, 15% untuk data validasi, dan 15% untuk data pengujian.

3.4 Implementasi Multi-Layer Perceptron (MLP)

Pada tahap ini, model MLP dibangun untuk mengklasifikasikan tingkat konsumsi energi menjadi dua kelas yaitu konsumsi 0% dan konsumsi 1% berdasarkan lima fitur terstandarisasi yakni waktu tempuh, jarak tempuh, kecepatan maksimum, perubahan elevasi, dan kepadatan lalu lintas. Imlpementasi mencakup penetapan arsitektur, proses pelatihan (*Training*), dan aturan keputusan (*Thresholding*).

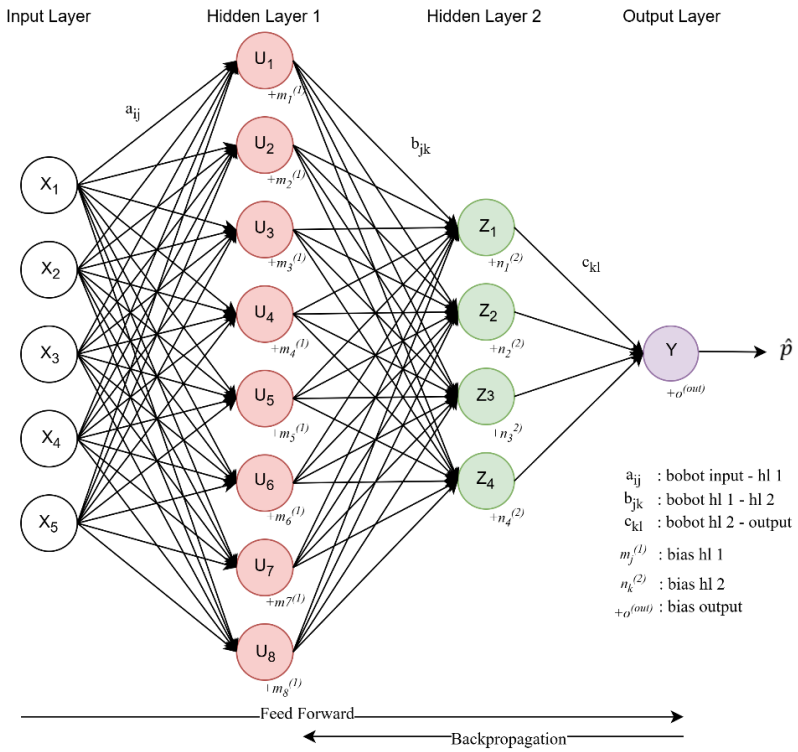
3.4.1 Arsitektur Model

Pada Gambar 3.6 ditunjukkan skema arsitektur Multilayer Perceptron (MLP) yang mempresentasikan salah satu konfigurasi model yang digunakan dalam penelitian ini. Secara spesifik, konfigurasi lapisan dalam arsitektur ini dibagi menjadi empat bagian dengan fungsi sebagai berikut:

1. *Input Layer*: Terdiri atas 5 neuron yang disesuaikan dengan dimensi fitur navigasi yang telah dinormalisasi menggunakan Z-score. Fitur-fitur tersebut meliputi waktu tempuh (t), jarak (s), kecepatan maksimum (topv), perubahan elevasi (*delta altitude*), dan kepadatan lalu lintas (*crowd*).
2. *Hidden Layer* 1: Jumlah neuron ditentukan berdasarkan pada skenario pengujian.

3. *Hidden Layer* 2: Menggunakan jumlah neuron yang lebih kecil dibandingkan lapisan pertama.
4. *Output Layer*: Menggunakan 1 neuron dengan fungsi aktivasi Sigmoid untuk menghasilkan probabilitas tunggal yang merepresentasikan prediksi kelas target (”Konsumsi 1%” atau Kelas 1).

Meskipun jumlah neuron pada lapisan tersembunyi berubah-ubah sesuai skenario, seluruh konfigurasi tetap konsisten menggunakan fungsi aktivasi ReLU pada kedua *hidden layer*. Penerapan ReLU bertujuan untuk menjaga efisiensi komputasi dan mencegah masalah gradien yang menghilang (*vanishing gradient*), sehingga model dapat belajar secara stabil meskipun menggunakan jumlah parameter yang berbeda-beda (Jacot dkk., 2022).



Gambar 3.6 Arsitektur MLP

3.4.2 Forward Pass

Tahap ini bertujuan untuk menghitung keluaran model dari fitur yang sudah dinormalisasi, tanpa melakukan pembaruan bobot. Secara ringkas, vektor input yang telah dinormalisasi dimasukkan ke jaringan. Nilai tersebut akan diproses berurutan melalui dua lapisan tersembunyi (*hidden layer*), dengan mengkombinasikan secara linier dengan bobot dan bias, lalu dilewatkan ke fungsi aktivasi (ReLU) untuk menghasilkan aktivasi pada *hidden layer* 1. Aktivasi ini yang kemudian akan menjadi input ke *hidden layer* 2 dan kembali mengalami proses yang sama hingga diperoleh hasil pada lapisan *output*.

Untuk perhitungan pra-aktivasi pada *hidden layer*-1 menggunakan Persamaan 3.2, dengan:

$$U_{(j)} = a_{ij}\, x + m_j^{(1)} \tag{3.2}$$

Aktivasi ReLu layer-1 sesuai dengan Persamaan 3.3:

$$U_{(j)} = \mathrm{ReLU}\big(U_{(j)}\big) \tag{3.3}$$

Aktivasi digunakan setelah kombinasi linear (bobot + bias) untuk menambahkan non-linearitas, sehingga jaringan mulai menangkap pola yang tidak linier. Proses pada *hidden layer* kedua mengikuti langkah yang sama seperti *hidden layer* pertama. Contoh perhitungan pra-aktivasi neuron pada *hidden layer*-2 menggunakan Persamaan 3.4 dan 3.5:

$$Z_{(k)} = b_{jk}\, U_{(j)} + n_k^{(2)} \tag{3.4}$$

$$Z_{(k)} = \mathrm{ReLU}\big(Z_{(k)}\big) \tag{3.5}$$

Selanjutnya, lapisan *output* mengubah hasil dari *hidden layer* kedua menjadi probabilitas kelas dengan Persamaan 3.6 dan 3.7.

$$Y = c_{kl}\, Z_{(k)} + o^{(out)} \tag{3.6}$$

$$\hat{p} = \sigma(Y) = \frac{1}{1+e^{-Y}} \tag{3.7}$$

Secara umum, keluaran pada lapisan akhir berupa nilai probabilitas di rentang 0 samppai 1. Probabilitas ini kemudian dipetakan menjadi kelas dengan aturan ambang sederhana: nilai yang sama dengan atau di atas 0,5 diklasifikasikan sebagai Kelas 1 (konsumsi 1%), sedangkan nilai di bawah 0,5 sebagai Kelas 0 (konsumsi 0%). Nilai ambang ini juga dapat disesuaikan pada tahap validasi untuk menyeimbangkan *Precision* dan *Recall* sesuai kebutuhan.

3.4.3 Fungsi Rugi (*Loss*)

Pada klasifikasi biner, fungsi rugi yang digunakan adalah Binary Cross-Entropy (BCE) karena secara langsung membandingkan probabilitas prediksi model dengan label target 0 atau 1. Tujuan dari fungsi rugi adalah membandingkan nilai BCE pada data latih sekaligus memantau BCE pada data validasi untuk mencegah *overfitting*. Persamaan yang digunakan untuk BCE ditunjukkan pada Persamaan 3.8.

$$\text{BCE}(y, \hat{p}) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\widehat{p_i}) + (1 - y_i)\log(1 - \widehat{p_i})] \tag{3.8}$$

Keterangan:
N : jumlah sampel
y_i : label sebenarnya untuk sampel ke-i
$\widehat{p_i}$: probabilitas prediksi model bahwa sampe ke-i termasuk Kelas 1 (konsumsi 1%)

Nilai probabilitas yang telah diperoleh sebelumnya, kemudian dimasukkan ke fungsi rugi Binary Cross-Entropy (BCE) pada Persamaan 3.9. Untuk satu sampel $(N = 1)$, maka Persamaan sederhananya menjadi:

$$\mathrm{BCE}(y, \hat{p}) = -[y \ln(\hat{p}) + (1 - y) \ln(1 - \hat{p})] \quad (3.9)$$

Untuk label taget $y = 1$, maka menggunakan Persamaan 3.10:

$$\mathrm{BCE}(1, \hat{p}) = -\ln(\hat{p}) \quad (3.10)$$

Untuk label target $y = 0$, maka 3.11:

$$\mathrm{BCE}(0, \hat{p}) = -\ln(1 - \hat{p}) \quad (3.11)$$

Setelah nilai probabilitas keluaran model diperoleh dari lapisan *output*, kesalahan prediksi diukur menggunakan Binary Cross-Entropy (BCE). Pemilihan BCE sejalan dengan keluaran sigmoid yang berbentuk probabilitas, sehingga perbedaan antara probabilitas prediksi dan label biner (0/1) dapat dihitung secara langsung. Dibandingkan MSE, BCE umumnya lebih peka terhadap kesalahan pada tugas klasifikasi, sehingga gradien yang dihasilkan lebih informatif dan proses pelatihan cenderung lebih cepat konvergen.

Dalam implementasi, perhitungan BCE dilakukan pada data latih untuk memperbarui bobot, sambil dipantau pada data validasi guna mendeteksi tanda-tanda *overfitting*. Banyak pustaka juga menyediakan bentuk perhitungan berbasis logit (menggunakan nilai sebelum sigmoid) untuk meningkatkan stabilitas numerik ketika probabilitas sangat mendekati 0 atau 1.

3.4.4 Backpropagation

Backpropagation digunakan untuk menghitung gradien *Loss* terhadap setiap bobot dan bias, lalu memperbarui parameter agar nilai *Loss* menurun dari *epoch* ke *epoch*. Prosesnya berlangsung per mini-batch dan berulang hingga memenuhi kriteria berhenti.

Untuk klasifikasi biner dengan aktivasi sigmoid dan *Loss* BCE, turunan *Loss* menjadi sederhana. Jika p adalah probabilitas keluaran dan y adalah label, maka *error* pada neuron *output* dengan Persamaan 3.12 dan 3.13:

$$\delta_{(Y)} = \hat{p} - y \tag{3.12}$$

$$\tilde{\delta}_{(Y)} = (c_{kl})^{\top} \delta_{(Y)} \tag{3.13}$$

Error di setiap *hidden layer* dihitung dari *error* layer setelahnya yang "ditarik" melalui bobot, lalu dikalikan turunan fungsi aktivasi. Tahap selanjutnya, menurunkan *error* ke *hidden layer*-2 menggunakan Persamaan 3.14:

$$\delta_{(Z)} = \tilde{\delta}_{(Z)} \odot \mathrm{ReLU}'\left(Z_{(k)}\right) \tag{3.14}$$

Setelah itu, menurunkan *error* ke *hidden layer*-1 menggunakan Persamaan yang sama dengan Persamaan 3.14. Dari langkah forward, yang menhasilkan $ReLU^{U_{(j)}}$. Dilanjut dengan menghitung $\tilde{\delta}^{(1)}$ dengan cara yang sama seperti menghitung $\tilde{\delta}^{(2)}$ menggunakan Persamaan 3.15.

$$\delta_{(U)} = \tilde{\delta}_{(U)} \odot \mathrm{ReLU}'\left(U_{(j)}\right) \tag{3.15}$$

Gradien dihitung dari aktivasi masukan ke layer tersebut dan *error* layer yang bersangkutan. Maka gradien *output* (*hidden layer*-2 → *output*) dihitung menggunakan Persamaan 3.16,

$$\frac{\partial J}{\partial c_{kl}} = \delta_{(Y)}\left(Z_{(k)}\right)^{\top} \tag{3.16}$$

Maka gradien layer ke-2 (*hidden layer*-1 → *hidden layer*-2) dihitung menggunakan Persamaan 3.17 dan 3.18,

$$\frac{\partial J}{\partial b_{jk}} = \delta_{(Z)}\left(U_{(j)}\right)^{\top} \tag{3.17}$$

$$\frac{\partial J}{\partial b_{jk}} = \delta_{(Z)} \tag{3.18}$$

Maka gradien layer ke-1 (input → *hidden layer*-1) dihitung menggunakan Persamaan 3.19 dan 3.20,

$$\frac{\partial J}{\partial a_{ij}} = \delta_{(U)} x^{\top} \tag{3.19}$$

$$\frac{\partial J}{\partial a_{ij}} = \delta_{(U)} \tag{3.20}$$

Setelah gradien tiap layer diperoleh, parameter diperbarui menggunakan *learning rate* yang ditetapkan. Proses ini diulang per *mini-batch* sepanjang *epoch*, sementara nilai *Loss* pada *Validation set* dipantau untuk menerapkan *early stopping* agar mencegah *overfitting*.

3.4.5 Pembaruan Parameter

Setelah gradien terhadap sistem parameter diperoleh melalui proses *backpropagation*, langkah berikutnya adalah memperbarui bobot dan bias agar nilai fungsi rugi menurun. Prinsip umumnya mengikuti *gradien descent*: parameter digeser berlawanan dengan arah gradien sebesar laju besaran (η).

Update parameter lapisan *output* (*hidden layer* 2 → *output*), dihitung menggunakan Persamaan 3.21 dan 3.22,

$$c_{kl\ baru}^{(Y)} = c_{kl\ ama}^{(Y)} - \eta \frac{\partial J}{\partial c_{kl}} = \delta^{(out)} \left(Z_{(k)}\right)^{\top} \tag{3.21}$$

$$o_{baru}^{(out)} = o_{lama}^{(out)} - \eta\, \delta^{(out)} \tag{3.22}$$

Update parameter layer ke-2 (*hidden layer* 1 → *hidden layer* 2), dihitung menggunakan Persamaan 3.23 dan 3.24,

$$b_{jk\ baru}^{(Z)} = b_{jk\ lama}^{(Z)} - \frac{\partial J}{\partial b_{jk}} \tag{3.23}$$

$$n_{j\ baru}^{(Z)} = n_{j\ lama}^{Z} - \eta\, \delta^{(2)} \tag{3.24}$$

Update parameter layer ke-1 (input → *hidden layer* 1), dihitung menggunakan Persamaan 3.25 dan 3.26,

$$a_{ij\ baru}^{(U)} = a_{ij\ lama}^{(U)} - \eta \frac{\partial J}{\partial a_{ij}} \tag{3.25}$$

$$m_{i\ baru}^{(U)} = m_{ij\ lama}^{(U)} - \eta\, \delta^{(1)} \tag{3.26}$$

3.4.6 Aturan Keputusan (*Threshold*)

Model menghasilkan probabilitas $\hat{p} = P(y = 1 \mid x)$ untuk kelas ”konsumsi 1%”. Aturan keputusan didefinisikan dengan Persamaan 3.27.

$$\hat{y} = \begin{cases} 1, jika\ \hat{p} \geq \tau \\ 0, jika\ \hat{p} < \tau \end{cases} \tag{3.27}$$

dengan τ adalah ambang keputusan. Nilai τ = 0.5 digunakan sebagai acuan awal, tetapi penetapan akhir τ dilakukan berdasarkan kinerja pada *Validation set* agar selaras dengan tujuan evaluasi dan karakter data.

BAB IV

UJI COBA DAN PEMBAHASAN

4.1 Skenario Uji Coba

Tahap ini menyajikan serangkaian skenario pengujian untuk mengevaluasi performa model dalam berbagai variasi parameter. Tujuannya adalah menemukan kombinasi parameter yang paling optimal untuk mendeteksi konsumsi energi pada navigasi cerdas kendaraan listrik. Evaluasi dilakukan dengan melihat *Accuracy* serta *Loss* konvergensi. Adapun parameter utama yang diuji ialah:

1. Arsitektur *Hidden Layer* dan Jumlah Neuron: Parameter ini memengaruhi kapasitas pembelajaran model. *Hidden layer* merupakan penghubung antar *input* dan *output* yang mampu memodelkan hubungan *non-linear* (Da Silva dkk., 2017). Sementara itu, jumlah neuron merepresentasikan banyaknya unit pada setiap *hidden layer*. Jumlah neuron yang lebih banyak dapat meningkatkan kemampuan model dalam mempelajari pola yang kompleks, namun juga menaikkan risiko *overfitting* dan biaya komputasi. Sebaliknya, jumlah neuron yang terlalu sedikit dapat menyebabkan model mengalami *underfitting* dan gagal menangkap hubungan non-linear (Diukarev & Starukhin, 2024).
2. Kondisi Data: Pengujian dilakukan pada dua kondisi data, yaitu *Imbalanced* (tidak seimbang) dan *Balanced* yang menggunakan teknik SMOTE.
3. Ambang Batas Keputusan (*Threshold*): Nilai ambang batas probabilitas yang menentukan hasil akhir deteksi dari *layer output*. Variasi *threshold* diuji untuk melihat dampaknya terhadap *trade-off* antara *Precision* dan *Recall*.

Variasi parameter yang digunakan pada pengujian dirangkum pada Tabel 4.1 berikut:

Tabel 4.1 Skenario Pengujian MLP

Uji Coba	Hidden Layer	Dataset	Jumlah Neuron	*Threshold*
1	2 Hidden Layer	*Imbalanced*	5-8-4-1	0.5
2				0.6
3				0.7
4			5-4-2-1	0.5
5				0.6
6				0.7
7	3 Hidden Layer		5-8-6-4-1	0.5
8				0.6
9				0.7
10			5-6-4-2-1	0.5
11				0.6
12				0.7
13	2 Hidden Layer	*Balanced* Data menggunakan SMOTE	5-8-4-1	0.5
14				0.6
15				0.7
16			5-4-2-1	0.5
17				0.6
18				0.7
19	3 Hidden Layer		5-8-6-4-1	0.5
20				0.6
21				0.7
22			5-6-4-2-1	0.5
23				0.6
24				0.7

Tabel 4.1 merangkum seluruh kombinasi skenario pengujian yang disusun berdasarkan tiga sumbu variasi utama, yakni arsitektur model (mencakup kedalaman *hidden layer* dan lebar neuron), kondisi keseimbangan data, serta aturan keputusan (*threshold*). Di samping variabel-variabel tersebut, sejumlah parameter dan prosedur pelatihan ditetapkan secara seragam untuk menjaga konsistensi eksperimen.

Selama proses pelatihan, model menggunakan *Learning rate* sebesar 0.001 dan *Batch Size* 32. Guna menjamin validitas hasil dan meminimalkan bias akibat pembagian data statis, seluruh skenario dievaluasi menggunakan metode *K-Fold Cross Validation* dengan nilai $k = 5$. Pendekatan ini memastikan bahwa setiap data memiliki kesempatan yang sama untuk menjadi data latih maupun data validasi, sehingga performa yang dihasilkan lebih merepresentasikan kemampuan generalisasi model secara objektif. Selain itu, mekanisme *Early Stopping* diterapkan dengan nilai *patience* 15 untuk mencegah *overfitting*, serta *Random Seed* dikunci pada nilai 42.

Pemilihan variasi jumlah *hidden layer* dan neuron didasarkan pada kebutuhan untuk menyeimbangkan kapasitas model dengan ketersediaan data. Arsitektur dengan neuron yang lebih lebar (seperti 5-8-6-4-1) diuji untuk meningkatkan kemampuan representasi model dalam menangkap hubungan non-linear yang kompleks antar fitur navigasi. Sebaliknya, arsitektur yang lebih ramping (seperti 5-4-2-1) disertakan untuk mengevaluasi apakah model yang lebih sederhana sudah cukup memadai dan lebih efisien secara komputasi. Penentuan jumlah neuron ini krusial karena jumlah yang terlalu besar pada data yang sedikit dapat memicu *overfitting*, sedangkan jumlah yang terlalu sedikit berisiko menyebabkan *underfitting* di mana model gagal menangkap pola data yang sebenarnya.

Selain arsitektur, untuk strategi penanganan data serta mencegah bias model terhadap kelas mayoritas, metode *Synthetic Minority Oversampling Technique* (SMOTE) diterapkan (Adamu-Fika dkk., 2023). Secara prosedural, SMOTE

bekerja dengan cara menciptakan sampel sintetis baru untuk kelas minoritas, yang bertujuan untuk menyeimbangkan distribusi kelas pada data pelatihan (Călin, 2025). Penerapan ini hanya dilakukan pada data pelatihan (*data Training*) untuk menghindari kebocoran data (*data leakage*) ke data validasi dan pengujian. Dengan menyeimbangkan data, tujuannya adalah agar model dapat mempelajari batas keputusan (*decision boundary*) kelas minoritas secara lebih efektif, yang krusial untuk meningkatkan metrik *Recall* dan *F1-Score* dalam evaluasi kinerja. Efektivitas teknik ini kemudian diuji bersamaan dengan variasi nilai *threshold* (0.5, 0.6, dan 0.7). Nilai ambang batas 0.5 digunakan sebagai acuan dasar, sementara nilai yang lebih tinggi diuji untuk melihat apakah memperketat kriteria klasifikasi kelas positif dapat mengurangi kesalahan prediksi (*False positive*) pada kondisi data yang bias.

Setelah model dilatih, evaluasi kinerja dilakukan secara menyeluruh untuk menilai kemampuan generalisasi model terhadap data baru. Metode evaluasi utama yang digunakan dalam tahap ini adalah *Confusion Matrix*. Pemilihan metode ini sangat krusial mengingat adanya ketidakseimbangan distribusi kelas pada dataset, kondisi di mana penggunaan metrik *Accuracy* semata berpotensi menghasilkan bias terhadap kelas mayoritas. Dengan memetakan hasil prediksi ke dalam empat kuadran kinerja, *Confusion Matrix* memfasilitasi analisis yang lebih rinci mengenai karakteristik kesalahan model, baik itu berupa kegagalan deteksi (*False Negative*) maupun kesalahan peringatan (*False positive*) (Owusu-Adjei dkk., 2023).

Tabel 4.2 merepresentasikan *Confusion Matrix* yang digunakan untuk mengukur performa model klasifikasi konsumsi energi. Dalam Tabel ini, A merepresentasikan Konsumsi 1% dan B merepresentasikan Konsumsi 0%.

Tabel 4.2 *Confusion Matrix*

Kelas Aktual	**Kelas Prediksi**	
	Konsumsi 1% (A)	Konsumsi 0% (B)
Konsumsi 1% (A)	True Positive (AA)	False Negative (AB)
Konsumsi 0% (B)	*False positive* (BA)	True Negative (BB)

Merujuk pada ilustrasi Tabel 4.2, hasil klasifikasi model dapat diuraikan ke dalam empat komponen evaluasi utama yang merepresentasikan jenis keputusan yang diambil oleh model. Keempat komponen tersebut didefinisikan secara spesifik dalam konteks deteksi konsumsi energi sebagai berikut:

1. ***True Positive* (TP)**: Kondisi di mana model dengan tepat memprediksi data sebagai Kelas 1, dan pada kenyataannya data aktual memang menunjukkan Kelas 1.
2. ***True Negative* (TN)**: Kondisi di mana model berhasil mengidentifikasi data sebagai Kelas 0 secara akurat sesuai dengan data aktual. Kemampuan ini vital agar sistem dapat merekomendasikan efisiensi yang valid.
3. ***False positive* (FP)**: Terjadi ketika model salah memprediksi data sebagai Kelas 1, padahal sebenarnya data tersebut adalah Kelas 0. Kesalahan ini disebut Kesalahan Tipe I, yang dapat menyebabkan sistem mengabaikan potensi efisiensi yang sebenarnya ada.
4. ***False Negative* (FN)**: Terjadi ketika model memprediksi data sebagai Kelas 0, padahal data aktual menunjukkan Kelas 1. Kesalahan Tipe II ini dianggap lebih berisiko dalam konteks kendaraan listrik, karena prediksi Kelas 0 yang meleset dapat menyebabkan estimasi daya yang terlalu optimis dan memicu risiko kehabisan baterai (*range anxiety*).

Berdasarkan komponen-komponen dalam *Confusion Matrix* tersebut, kinerja model selanjutnya diukur menggunakan metrik statistik yang ditunjukkan pada persamaan-persamaan sebagai berikut:

1. ***Accuracy* (*Accuracy*)**: Mengukur proporsi prediksi yang benar secara keseluruhan (baik Kelas 0 maupun Kelas 1) dibandingkan total data. Adapun perhitungan *Accuracy* ditunjukkan pada Persamaan 4.1.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

2. ***Precision***: Mengukur tingkat ketepatan model dalam memprediksi Kelas 1, sehingga meminimalkan data Kelas 0 yang salah dideteksi sebagai Kelas 1. Perhitungan *Precision* ditunjukkan pada Persamaan 4.2.

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

3. ***Recall* (Sensitivitas)**: Mengukur kemampuan model untuk mendeteksi seluruh data Kelas 1 yang sebenarnya, guna meminimalkan risiko kesalahan prediksi menjadi Kelas 0. Untuk perhitungan *Recall* ditunjukkan pada Persamaan 4.3.

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

4. ***F1-Score***: Merupakan rata-rata harmonis dari *Precision* dan *Recall*, yang memberikan gambaran keseimbangan performa model, terutama ketika distribusi data antar kelas tidak seimbang. Untuk perhitungan *F1-Score* ditunjukkan pada Persamaan 4.4.

$$F1 - score = \frac{2\ x\ Precision\ x\ Recall}{Precision + Recall} \tag{4.4}$$

4.2 Hasil Uji Coba Data *Imbalanced*

Bagian ini memaparkan hasil pengujian model MLP pada kondisi data tidak seimbang (*Imbalanced*), di mana dataset hanya melalui tahap pra-pemrosesan normalisasi *Z-Score* tanpa penerapan SMOTE. Penilaian keberhasilan model didasarkan pada tingkat kesesuaian antara kelas hasil prediksi dengan data aktual pada tahap pengujian. Kinerja model selanjutnya dianalisis secara mendalam melalui perhitungan empat indikator utama, yaitu *Accuracy*, *Precision*, *Recall*, dan *F1-Score*.

4.2.1 Skenario 1 (5-8-4-1)

Model pada Skenario Uji Coba 1 menggunakan arsitektur dua *hidden layer* dengan jumlah neuron 5-8-4-1 dan *Threshold* 0.50. Hasil evaluasi performa model pada masing-masing fold disajikan pada Tabel 4.3.

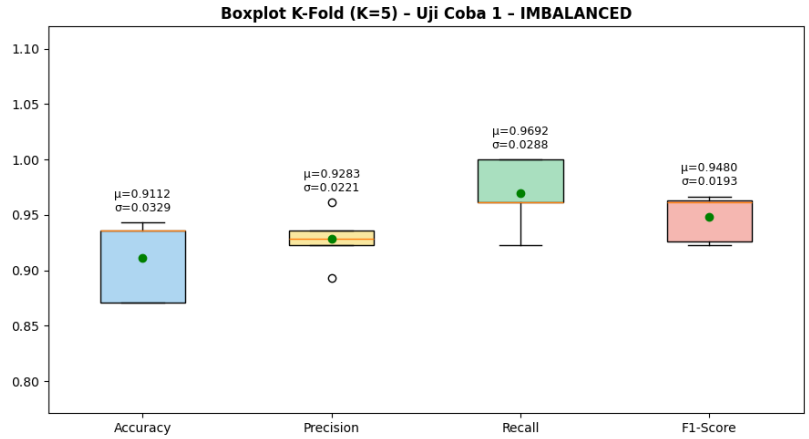
Tabel 4.3 *Evaluation Matrix* K-Fold Skenario 1

Fold	*Accuracy*	*Precision*	*Recall*	*F1-Score*
1	93.55%	96.15%	96.15%	96.15%
2	87.10%	89.29%	96.15%	92.59%
3	93.55%	92.86%	100.00%	96.30%
4	87.10%	92.31%	92.31%	92.31%
5	94.29%	93.55%	100.00%	96.67%
Rata-rata	**91.12%**	**92.83%**	**96.92%**	**94.80%**
Standard Deviasi	0.0329	0.0221	0.0288	0.0193

Berdasarkan Tabel 4.3, model menghasilkan rata-rata *Accuracy* sebesar 91,12%, *Precision* sebesar 92,83%, *Recall* sebesar 96,92%, dan *F1-Score* sebesar 94,80%. Nilai *Recall* yang tinggi menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam mendeteksi kelas 1 (konsumsi energi 1%), bahkan pada Fold ke-3 dan Fold ke-5 model mencapai *Recall* sebesar 100%, yang berarti seluruh

sampel kelas 1 pada fold tersebut berhasil diklasifikasikan dengan benar. Namun demikian, nilai *Precision* yang lebih rendah dibandingkan *Recall* mengindikasikan masih terdapat sejumlah *false positive*, yaitu sampel kelas 0 (konsumsi energi 0%) yang diprediksi sebagai kelas 1.

Selanjutnya distribusi statistik metrik evaluasi divisualisasikan pada Gambar 4.1.



Gambar 4.1 Boxplot Rata-rata Evaluation Matrix Skenario 1 (5-8-4-1)

Visualisasi boxplot pada Gambar 4.1 memperlihatkan bahwa distribusi metrik *Recall* dan *F1-Score* relatif lebih rapat dan stabil dibandingkan metrik *Accuracy*. Hal ini tercermin dari nilai standar deviasi *Recall* sebesar 0,0288 dan *F1-Score* sebesar 0,0193, yang menunjukkan konsistensi performa model antar fold. Sebaliknya, *Accuracy* memiliki standar deviasi tertinggi sebesar 0,0329, yang mengindikasikan adanya variasi performa model akibat perbedaan distribusi data pada masing-masing partisi K-Fold.

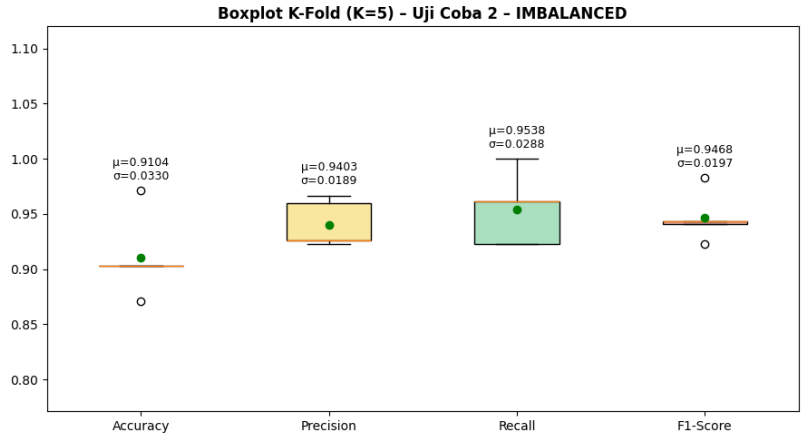
4.2.2 Skenario 2 (5-8-4-1)

Model pada Skenario Uji Coba 2 menggunakan arsitektur dan jumlah neuron yang sama dengan skenario 1, namun *Threshold* ditingkatkan menjadi 0.60. Hasil evaluasi performa model pada masing-masing fold disajikan pada Tabel 4.4.

Berdasarkan hasil evaluasi pada Tabel 4.4, model menghasilkan rata-rata *Accuracy* sebesar 91,04%, *Precision* sebesar 94,03%, *Recall* sebesar 95,38%, dan *F1-Score* sebesar 94,68%. Dibandingkan dengan Skenario 1, peningkatan nilai *Precision* menunjukkan bahwa penyesuaian *threshold* menjadi 0,60 membuat model lebih selektif dalam memprediksi kelas 1 (konsumsi energi 1%), sehingga jumlah prediksi positif palsu dapat dikurangi tanpa menurunkan kemampuan deteksi secara signifikan. Selanjutnya distribusi statistik metrik evaluasi divisualisasikan pada Gambar 4.2.

Tabel 4.4 *Evaluation Matrix* K-Fold Skenario 2

Fold	***Accuracy***	***Precision***	***Recall***	***F1-Score***
1	90.32%	96.00%	92.31%	94.12%
2	90.32%	92.59%	96.15%	94.34%
3	90.32%	92.59%	96.15%	94.34%
4	87.10%	92.31%	92.31%	92.31%
5	97.14%	96.67%	100.00%	98.31%
Rata-rata	**91.04%**	**94.03%**	**95.38%**	**94.68%**
Standard Deviasi	0.033	0.0189	0.0288	0.0197



Gambar 4.2 Boxplot Rata-rata Evaluation Matrix Skenario 2 (5-8-4-1)

Visualisasi boxplot pada Gambar 4.2 memperlihatkan bahwa distribusi metrik *Precision* dan *F1-Score* cenderung lebih rapat, yang ditunjukkan oleh nilai standar deviasi *Precision* sebesar 0,0189 dan *F1-Score* sebesar 0,0197. Hal ini mengindikasikan bahwa performa model pada skenario ini relatif stabil antar fold. Sementara itu, *Accuracy* memiliki standar deviasi sebesar 0,033, yang menunjukkan adanya variasi performa akibat perbedaan distribusi data pada masing-masing partisi K-Fold.

4.2.3 Skenario 3 (5-8-4-1)

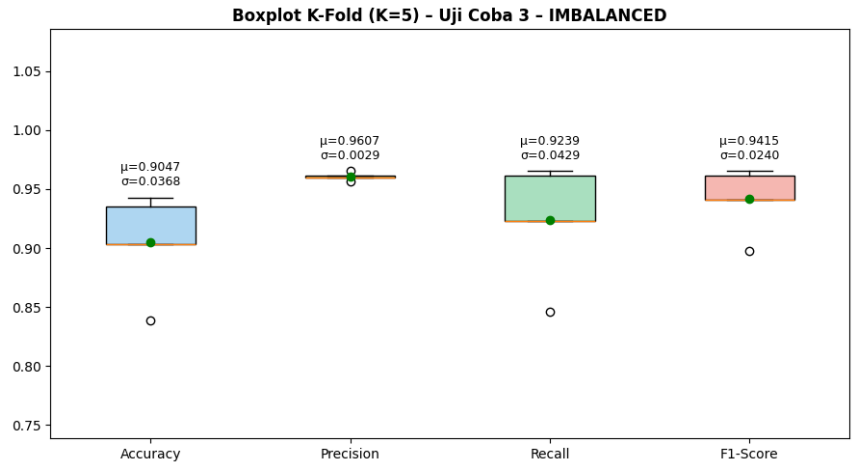
Model pada Skenario Uji Coba 3 menggunakan arsitektur dan jumlah neuron yang sama dengan skenario sebelumnya, dengan *threshold* ditingkatkan menjadi 0.70. Evaluasi performa model dilakukan menggunakan K-Fold Cross-Validation dengan K = 5, dan hasil pengujian pada masing-masing fold disajikan pada Tabel 4.5.

Tabel 4.5 *Evaluation Matrix* K-Fold Skenario 3

Fold	***Accuracy***	***Precision***	***Recall***	***F1-Score***
1	90.32%	96.00%	92.31%	94.12%

2	90.32%	96.00%	92.31%	94.12%
3	93.55%	96.15%	96.15%	96.15%
4	83.87%	95.65%	84.62%	89.80%
5	94.29%	96.55%	96.55%	96.55%
Rata-rata	**90.47%**	**96.07%**	**92.39%**	**94.15%**
Standard Deviasi	0.0368	0.0029	0.0429	0.024

Berdasarkan Tabel 4.5, model menghasilkan rata-rata *Accuracy* sebesar 90,47%, *Precision* sebesar 96,07%, *Recall* sebesar 92,39%, dan *F1-Score* sebesar 94,15%. Nilai *Precision* yang tinggi menunjukkan bahwa sebagian besar prediksi kelas 1 (konsumsi energi 1%) dilakukan dengan tepat, sedangkan nilai *Recall* yang lebih rendah mengindikasikan masih terdapat beberapa sampel kelas 1 yang tidak terdeteksi. Peningkatan *threshold* menjadi 0,70 menyebabkan model menjadi lebih selektif dalam memprediksi kelas 1, sehingga *Precision* meningkat dengan konsekuensi penurunan *Recall*. Selanjutnya visualisasi distribusi metrik evaluasi ditunjukkan pada Gambar 4.3.



Gambar 4.3 Boxplot Rata-rata Evaluation Matrix Skenario 3 (5-8-4-1)

Berdasarkan Gambar 4.3, boxplot K-Fold pada Skenario 3 menunjukkan bahwa metrik *Precision* memiliki distribusi yang sangat rapat dan stabil, sedangkan

variasi yang lebih besar terlihat pada metrik *Recall* dan *Accuracy*. Hal ini mengindikasikan bahwa model pada skenario ini lebih konsisten dalam menghasilkan prediksi kelas 1 dibandingkan dalam mendeteksi seluruh sampel kelas tersebut pada setiap fold.

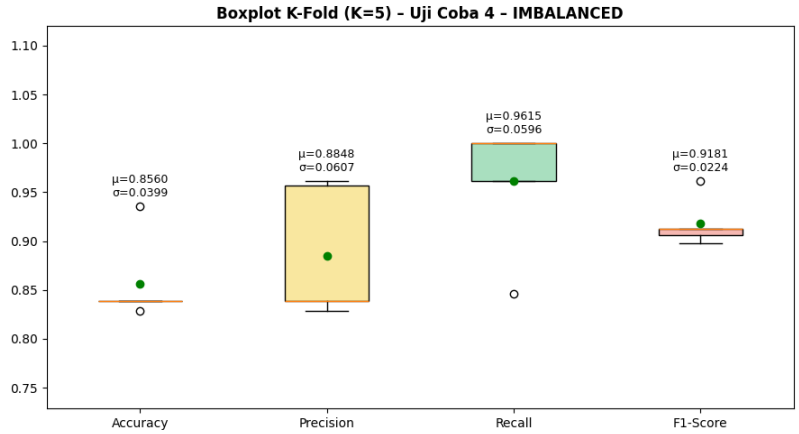
Secara keseluruhan, hasil pengujian pada arsitektur MLP 5–8–4–1 dengan variasi *threshold* 0,50 hingga 0,70 menunjukkan bahwa model memiliki performa yang konsisten dalam mendeteksi Kelas 1 (konsumsi energi 1%). Peningkatan nilai *threshold* cenderung meningkatkan nilai *Precision* dengan penurunan *Recall* yang relatif terbatas dan tidak signifikan, sehingga trade-off antara ketepatan dan sensitivitas deteksi masih berada dalam rentang yang terkendali. Temuan ini menunjukkan bahwa arsitektur 5–8–4–1 cukup stabil terhadap variasi *threshold*.

4.2.4 Skenario 4 (5-4-2-1)

Pada Skenario Uji Coba 4, dilakukan pengujian model menggunakan arsitektur tiga hidden layer dengan jumlah neuron 5–4–2–1 sebagai bentuk penyederhanaan struktur jaringan dibandingkan skenario sebelumnya. Hasil evaluasi performa model pada masing-masing fold dirangkum pada Tabel 4.6.

Tabel 4.6 *Evaluation Matrix* K-Fold Skenario 4

Fold	***Accuracy***	***Precision***	***Recall***	***F1-Score***
1	93.55%	96.15%	96.15%	96.15%
2	83.87%	83.87%	100.00%	91.23%
3	83.87%	83.87%	100.00%	91.23%
4	83.87%	95.65%	84.62%	89.80%
5	82.86%	82.86%	100.00%	90.62%
Rata-rata	**85.60%**	**88.48%**	**96.15%**	**91.81%**
Standard Deviasi	0.0399	0.0607	0.0596	0.0224



Gambar 4.4 Boxplot Rata-rata Evaluation Matrix Skenario 4 (5-4-2-1)

Berdasarkan Tabel 4.6, diperoleh rata-rata *Accuracy* sebesar 85,60%, *Precision* sebesar 88,48%, *Recall* sebesar 96,15%, dan *F1-Score* sebesar 91,81%. Nilai *Recall* yang tetap tinggi menunjukkan bahwa model masih mampu mendeteksi sebagian besar sampel kelas 1 (konsumsi energi 1%), namun penurunan nilai *Precision* mengindikasikan meningkatnya jumlah prediksi positif palsu, sehingga berdampak pada penurunan *Accuracy* dan *F1-Score* secara keseluruhan. Adapun visualisasi distribusi metrik evaluasi ditunjukkan pada Gambar 4.4.

Visualisasi pada Gambar 4.4 memperlihatkan bahwa variasi performa terbesar terjadi pada metrik *Precision* dan *Recall*, yang ditunjukkan oleh nilai standar deviasi masing-masing sebesar 0,0607 dan 0,0596. Sebaliknya, metrik *F1-Score* menunjukkan distribusi yang relatif lebih stabil dengan standar deviasi sebesar 0,0224, menandakan bahwa performa agregat model masih berada dalam rentang yang konsisten meskipun terdapat fluktuasi pada metrik individual.

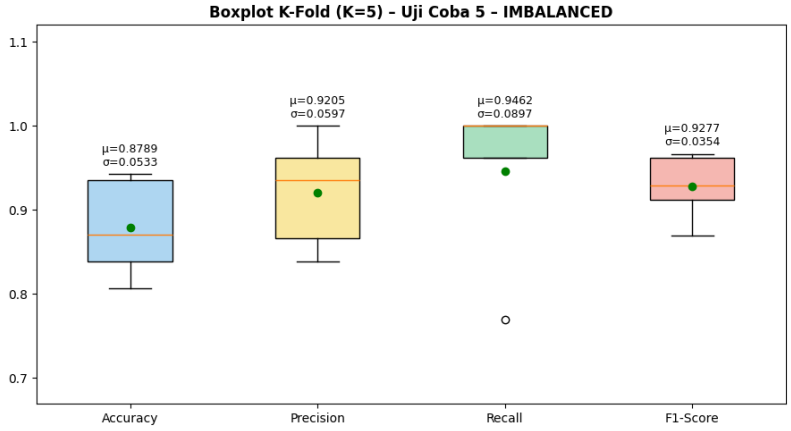
4.2.5 Skenario 5 (5-4-2-1)

Pada skenario ini, konfigurasi arsitektur 5-4-2-1 dipertahankan, namun *Threshold* ditingkatkan menjadi 0.60. Ringkasan hasil evaluasi performa model pada masing-masing fold disajikan pada Tabel 4.7.

Tabel 4.7 *Evaluation Matrix* K-Fold Skenario 5

Fold	***Accuracy***	***Precision***	***Recall***	***F1-Score***
1	93.55%	96.15%	96.15%	96.15%
2	87.10%	86.67%	100.00%	92.86%
3	83.87%	83.87%	100.00%	91.23%
4	80.65%	100.00%	76.92%	86.96%
5	94.29%	93.55%	100.00%	96.67%
Rata-rata	**87.89%**	**92.05%**	**94.62%**	**92.77%**
Standard Deviasi	0.0533	0.0597	0.0897	0.0354

Berdasarkan Tabel 4.7, model menghasilkan rata-rata *Accuracy* sebesar 87,89%, *Precision* sebesar 92,05%, *Recall* sebesar 94,62%, dan *F1-Score* sebesar 92,77%. Nilai *Recall* yang relatif tinggi menunjukkan bahwa model masih mampu mendeteksi sebagian besar sampel kelas 1 (konsumsi energi 1%), namun fluktuasi nilai *Precision* dan *Recall* pada beberapa fold mengindikasikan adanya ketidakkonsistenan performa antar partisi data.



Gambar 4.5 Boxplot Rata-rata Evaluation Matrix Skenario 5 (5-4-2-1)

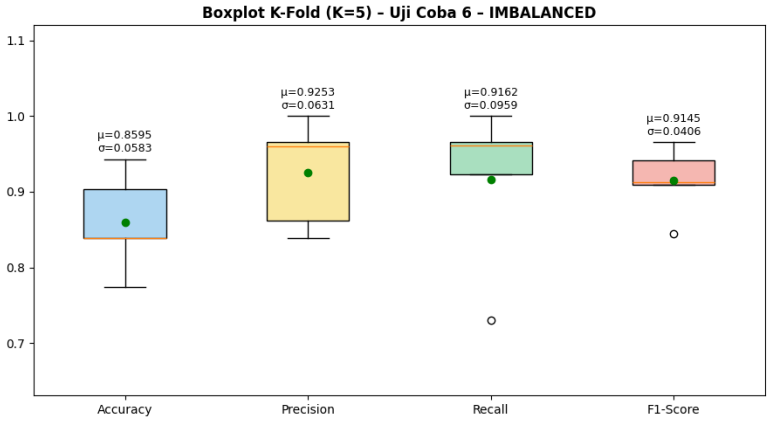
Visualisasi pada Gambar 4.5 menunjukkan bahwa variasi performa paling besar terjadi pada metrik *Recall*, yang tercermin dari nilai standar deviasi sebesar 0,0897, diikuti oleh *Precision* dengan standar deviasi sebesar 0,0597. Sebaliknya, metrik *F1-Score* memiliki sebaran yang relatif lebih stabil, menandakan bahwa performa keseluruhan model masih berada dalam rentang yang dapat diterima meskipun terdapat variasi pada metrik individual.

4.2.6 Skenario 6 (5-4-2-1)

Pada skenario 6 dengan arsitektur *hidden layer* dan neuron yang sama dengan skenario 4 dan 5 dengan meningkatkan *Threshold* menjadi 0.70. Ringkasan hasil evaluasi performa model pada masing-masing fold disajikan pada Tabel 4.8. Mengacu pada Tabel 4.8, model menghasilkan rata-rata *Accuracy* sebesar 85,95%, *Precision* sebesar 92,53%, *Recall* sebesar 91,62%, dan *F1-Score* sebesar 91,45%. Nilai *Recall* yang lebih rendah dibandingkan skenario sebelumnya menunjukkan bahwa model tidak selalu berhasil mendeteksi seluruh sampel kelas 1 (konsumsi energi 1%), khususnya pada beberapa fold, yang berdampak pada penurunan nilai *F1-Score*.

Tabel 4.8 *Evaluation Matrix* K-Fold Skenario 6

Fold	***Accuracy***	***Precision***	***Recall***	***F1-Score***
1	90.32%	96.00%	92.31%	94.12%
2	83.87%	86.21%	96.15%	90.91%
3	83.87%	83.87%	100.00%	91.23%
4	77.42%	100.00%	73.08%	84.44%
5	94.29%	96.55%	96.55%	96.55%
Rata-rata	**85.95%**	**92.53%**	**91.62%**	**91.45%**
Standard Deviasi	0.0583	0.0631	0.0959	0.0406



Gambar 4.6 Boxplot Rata-rata Evaluation Matrix Skenario 6 (5-4-2-1)

Berdasarkan visualisasi pada Gambar 4.6, terlihat bahwa sebaran nilai *Recall* memiliki variasi paling besar dibandingkan metrik lainnya, yang tercermin dari nilai standar deviasi sebesar 0,0959. Sementara itu, metrik *Precision* dan *F1-Score* menunjukkan distribusi yang relatif lebih rapat, menandakan bahwa model masih cukup konsisten dalam menghasilkan prediksi positif meskipun sensitivitas deteksinya menurun.

Secara keseluruhan, hasil pengujian pada arsitektur 5–4–2–1 dengan variasi *threshold* 0,50 hingga 0,70 menunjukkan bahwa model tetap memiliki performa yang tangguh dalam mendeteksi kelas 1 (konsumsi energi 1%), meskipun menggunakan struktur jaringan yang lebih sederhana. Peningkatan nilai *threshold* terbukti secara konsisten meningkatkan nilai *Precision*, yang merepresentasikan kemampuan model dalam meminimalkan kesalahan prediksi positif palsu (*false positive*). Meskipun kenaikan *threshold* memicu penurunan nilai *Recall*, kompensasi peningkatan pada *Precision* menjaga nilai *F1-Score* tetap stabil di rentang 91% hingga 92%. Temuan ini mengindikasikan bahwa arsitektur 5–4–2–1

mencapai titik keseimbangan (*trade-off*) optimal pada *threshold* 0,60, yang menghasilkan rata-rata *Accuracy* tertinggi sebesar 87,89%. Hal ini membuktikan bahwa penyederhanaan struktur jaringan tetap mampu mempertahankan reliabilitas deteksi melalui penyesuaian parameter *threshold* yang tepat.

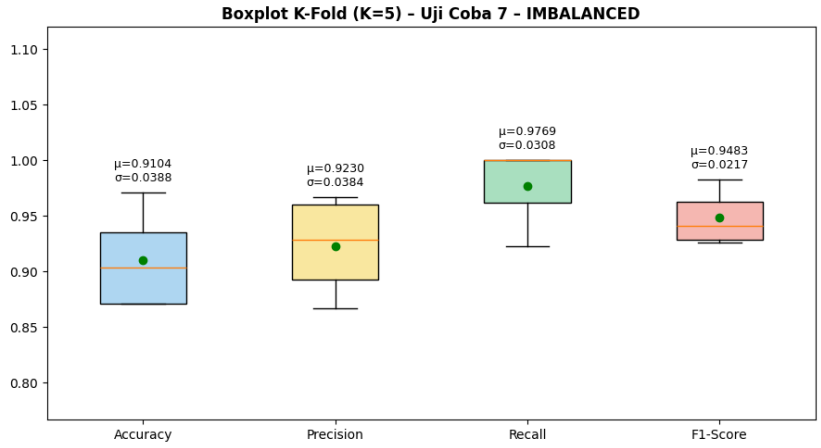
4.2.7 Skenario 7 (5-8-6-4-1)

Pada skenario 7, arsitektur ditingkatkan menjadi tiga *hidden layer* yang lebih kompleks dengan jumlah neuron 5-8-6-4-1 dan *Threshold* ditetapkan pada 0.50. Peningkatan *hidden layer* bertujuan untuk membandingkan hasil evaluasi dengan arsitektur dua *hidden layer*. Ringkasan hasil evaluasi performa model pada masing-masing fold disajikan pada Tabel 4.9.

Mengacu pada Tabel 4.9, model menghasilkan rata-rata *Accuracy* sebesar 91,04%, *Precision* sebesar 92,30%, *Recall* sebesar 97,69%, dan *F1-Score* sebesar 94,83%. Nilai *Recall* yang sangat tinggi menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam mendeteksi kelas 1 (konsumsi energi 1%), di mana pada beberapa fold model berhasil mencapai *Recall* sebesar 100%, menandakan tidak adanya kesalahan false negative pada fold tersebut.Visualisasi distribusi metrik evaluasi untuk skenario ini dapat dilihat pada Gambar 4.7.

Tabel 4.9 *Evaluation Matrix* K-Fold Skenario 7

Fold	***Accuracy***	***Precision***	***Recall***	***F1-Score***
1	87.10%	89.29%	96.15%	92.59%
2	93.55%	92.86%	100.00%	96.30%
3	87.10%	86.67%	100.00%	92.86%
4	90.32%	96.00%	92.31%	94.12%
5	97.14%	96.67%	100.00%	98.31%
Rata-rata	**91.04%**	**92.30%**	**97.69%**	**94.83%**
Standard Deviasi	0.0388	0.0384	0.0308	0.0217



Gambar 4.7 Boxplot Rata-rata Evaluation Matrix Skenario 7 (5-8-6-4-1)

Berdasarkan visualisasi pada Gambar 4.7, boxplot K-Fold pada Skenario 7 memperlihatkan bahwa metrik *Recall* dan *F1-Score* memiliki distribusi yang relatif rapat dan stabil, yang tercermin dari nilai standar deviasi masing-masing sebesar 0,0308 dan 0,0217. Sementara itu, metrik *Accuracy* dan *Precision* menunjukkan variasi yang sedikit lebih besar, namun masih berada dalam rentang yang konsisten antar fold.

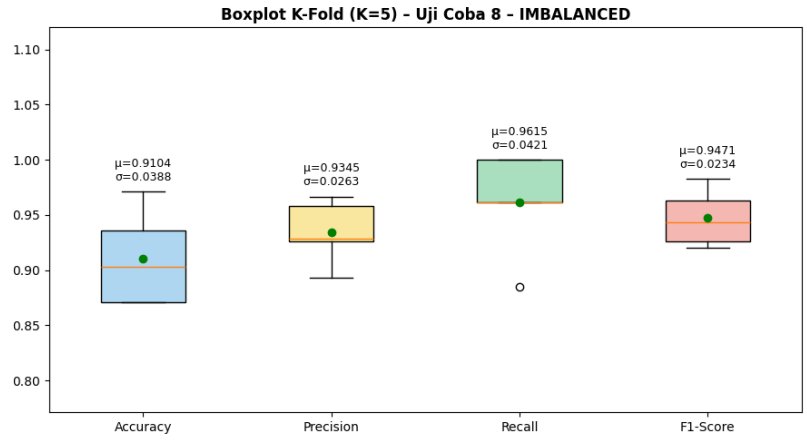
4.2.8 Skenario 8 (5-8-6-4-1)

Pada skenario ini, arsitektur tiga *hidden layer* dan konfigurasi yang sama seperti Skenario 7 dengan menaikkan *threshold* menjadi 0.6. Penyesuaian ini bertujuan untuk mengamati pengaruh peningkatan ambang batas keputusan terhadap ketepatan prediksi model pada struktur jaringan yang lebih dalam. Hasil evaluasi performa model untuk setiap *fold* pada skenario ini dirangkum dalam Tabel 4.10.

Tabel 4.10 *Evaluation Matrix* K-Fold Skenario 8

Fold	*Accuracy*	*Precision*	*Recall*	*F1-Score*
1	87.10%	89.29%	96.15%	92.59%
2	90.32%	92.59%	96.15%	94.34%
3	93.55%	92.86%	100.00%	96.30%
4	87.10%	95.83%	88.46%	92.00%
5	97.14%	96.67%	100.00%	98.31%
Rata-rata	**91.04%**	**93.45%**	**96.15%**	**94.71%**
Standard Deviasi	0.0388	0.0263	0.0421	0.0234

Berdasarkan Tabel 4.10, model menghasilkan rata-rata *Accuracy* sebesar 91,04%, *Precision* sebesar 93,45%, *Recall* sebesar 96,15%, dan *F1-Score* sebesar 94,71%. Nilai *Recall* yang tinggi menunjukkan bahwa model tetap mampu mendeteksi sebagian besar sampel kelas 1 (konsumsi energi 1%), sementara peningkatan nilai *Precision* dibandingkan Skenario 7 mengindikasikan berkurangnya jumlah prediksi positif palsu pada konfigurasi ini.



Gambar 4.8 Boxplot Rata-rata Evaluation Matrix Skenario 8 (5-8-6-4-1)

Dari Gambar 4.8, boxplot K-Fold pada Skenario 8 memperlihatkan bahwa metrik *Precision* dan *F1-Score* memiliki distribusi yang relatif rapat dan stabil, dengan nilai standar deviasi masing-masing sebesar 0,0263 dan 0,0234. Meskipun

metrik *Recall* menunjukkan variasi yang sedikit lebih besar (standar deviasi 0,0421), performa deteksi kelas 1 tetap berada pada tingkat yang tinggi dan konsisten antar fold.

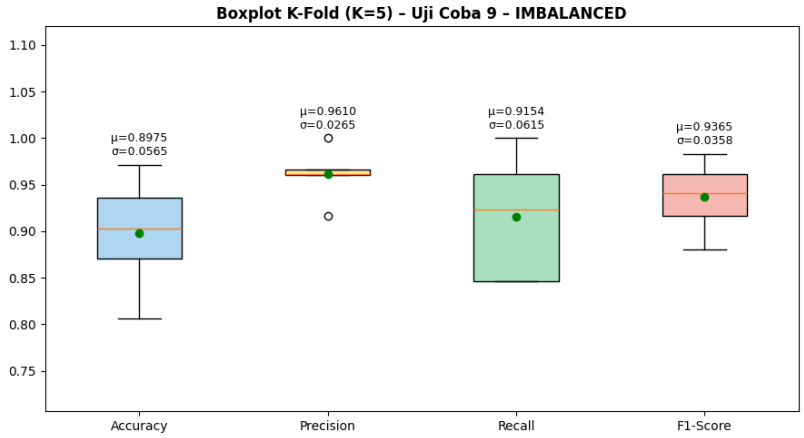
4.2.9 Skenario 9 (5-8-6-4-1)

Pada model konfigurasi yang digunakan tetap menggunakan arsitektur serta jumlah neuron yang sama seperti Skenario 7 dan 8 dengan peningkatan *Threshold* yang ketat menjadi 0.70. Ringkasan hasil evaluasi performa model pada masing-masing fold disajikan pada Tabel 4.11

Hasil evaluasi pada Tabel 4.11, model menghasilkan rata-rata *Accuracy* sebesar 89,75%, *Precision* sebesar 96,10%, *Recall* sebesar 91,54%, dan *F1-Score* sebesar 93,65%. Nilai *Precision* yang sangat tinggi menunjukkan bahwa model sangat selektif dalam memprediksi kelas 1 (konsumsi energi 1%), namun nilai *Recall* yang lebih rendah dibandingkan skenario sebelumnya mengindikasikan bahwa masih terdapat sebagian sampel kelas 1 yang tidak terdeteksi, khususnya pada beberapa fold.

Tabel 4.11 *Evaluation Matrix* K-Fold Skenario 9

Fold	***Accuracy***	***Precision***	***Recall***	***F1-Score***
1	90.32%	96.00%	92.31%	94.12%
2	80.65%	91.67%	84.62%	88.00%
3	93.55%	96.15%	96.15%	96.15%
4	87.10%	100.00%	84.62%	91.67%
5	97.14%	96.67%	100.00%	98.31%
Rata-rata	**89.75%**	**96.10%**	**91.54%**	**93.65%**
Standard Deviasi	0.0565	0.0265	0.0615	0.0358



Gambar 4.9 Boxplot Rata-rata Evaluation Matrix Skenario 9 (5-8-6-4-1)

Berdasarkan visualisasi pada Gambar 4.9, terlihat bahwa metrik *Precision* memiliki distribusi yang paling rapat dan stabil, dengan nilai standar deviasi sebesar 0,0265, sedangkan metrik *Recall* menunjukkan variasi yang lebih besar dengan standar deviasi 0,0615. Kondisi ini mengindikasikan bahwa peningkatan ketepatan prediksi kelas 1 pada skenario ini dicapai dengan konsekuensi penurunan sensitivitas deteksi yang lebih bervariasi antar fold.

Berdasarkan hasil pengujian pada Skenario 7, 8, dan 9 yang menggunakan arsitektur 5–8–6–4–1, dapat disimpulkan bahwa penambahan jumlah hidden layer dan neuron mampu meningkatkan kemampuan model dalam mendeteksi kelas 1 (konsumsi energi 1%), yang tercermin dari nilai *Recall* yang tinggi dan relatif konsisten pada sebagian besar skenario. Meskipun demikian, variasi konfigurasi parameter menunjukkan adanya pergeseran keseimbangan antara *Precision* dan *Recall*, di mana peningkatan ketepatan prediksi pada beberapa skenario diikuti oleh penurunan sensitivitas deteksi yang tidak signifikan. Secara keseluruhan, konfigurasi 5–8–6–4–1 menunjukkan performa yang stabil dan lebih adaptif dalam menangkap pola data yang kompleks dibandingkan arsitektur yang lebih sederhana,

sehingga layak dipertimbangkan sebagai arsitektur unggulan pada kondisi dataset yang tidak seimbang.

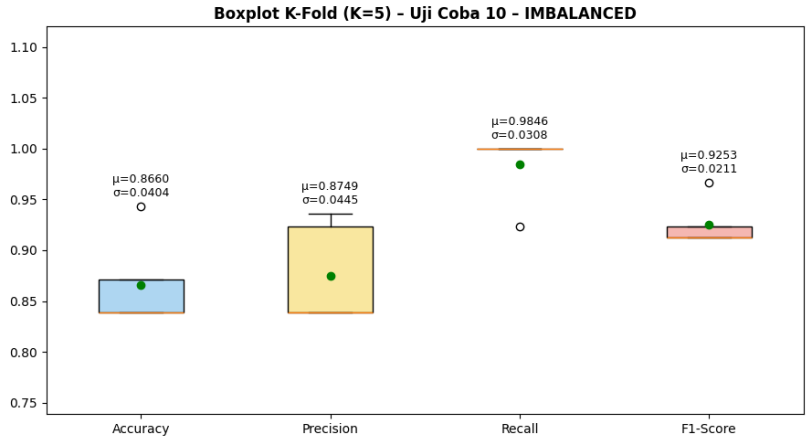
4.2.10 Skenario 10 (5-6-4-2-1)

Kelompok pengujian terakhir untuk data *Imbalanced* menerapkan arsitektur tiga *hidden layer* namun dengan jumlah neuron yang lebih sederhana, yaitu 5-6-4-2-1 dan penetapan *Threshold* 0.50. Hal ini bertujuan untuk mengevaluasi apakah kedalaman jaringan dapat mengkompensasi keterbatasan kapasitas neuron dalam mempelajari fitur. Ringkasan hasil evaluasi performa model pada masing-masing fold disajikan pada Tabel 4.12.

Berdasarkan hasil pengujian yang dirangkum dalam Tabel 4.12, model mencapai rata-rata *Accuracy* sebesar 86,60%, *Precision* sebesar 87,49%, *Recall* sebesar 98,46%, dan *F1-Score* sebesar 92,53%. Tingginya nilai *Recall* menunjukkan bahwa model sangat sensitif dalam mendeteksi kelas 1 (konsumsi energi 1%), di mana sebagian besar fold menunjukkan nilai *Recall* mencapai 100%. Namun demikian, ketepatan prediksi terhadap kelas tersebut relatif lebih rendah, yang tercermin dari nilai *Precision* yang menurun.

Tabel 4.12 *Evaluation Matrix* K-Fold Skenario 10

Fold	***Accuracy***	***Precision***	***Recall***	***F1-Score***
1	83.87%	83.87%	100.00%	91.23%
2	83.87%	83.87%	100.00%	91.23%
3	83.87%	83.87%	100.00%	91.23%
4	87.10%	92.31%	92.31%	92.31%
5	94.29%	93.55%	100.00%	96.67%
Rata-rata	**86.60%**	**87.49%**	**98.46%**	**92.53%**
Standard Deviasi	0.0404	0.0445	0.0308	0.0211



Gambar 4.10 Boxplot Rata-rata Evaluation Matrix Skenario 10 (5-6-4-2-1)

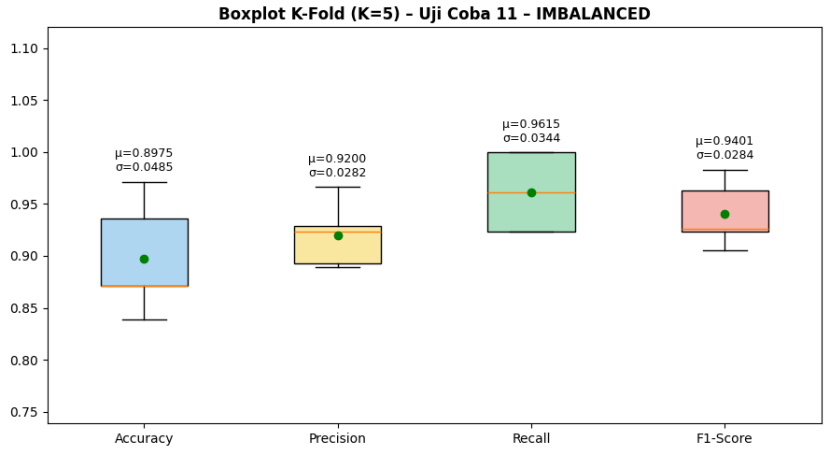
Apabila ditinjau dari Gambar 4.10, terlihat bahwa distribusi nilai *Recall* sangat rapat dan stabil dibandingkan metrik lainnya, yang tercermin dari nilai standar deviasi yang relatif kecil. Sebaliknya, metrik *Accuracy* dan *Precision* menunjukkan variasi yang lebih besar antar fold, menandakan bahwa performa model pada aspek ketepatan prediksi masih dipengaruhi oleh pembagian data.

4.2.11 Skenario 11 (5-6-4-2-1)

Skenario 11 menggunakan arsitektur serta jumlah neuron yang sama dengan skenario 10 dengan meningkatkan nilai *Threshold* menjadi 0.60. Ringkasan hasil evaluasi pada masing-masing fold disajikan pada Tabel 4.12.

Tabel 4.13 *Evaluation Matrix* K-Fold Skenario 11

Fold	***Accuracy***	***Precision***	***Recall***	***F1-Score***
1	87.10%	89.29%	96.15%	92.59%
2	93.55%	92.86%	100.00%	96.30%
3	83.87%	88.89%	92.31%	90.57%
4	87.10%	92.31%	92.31%	92.31%
5	97.14%	96.67%	100.00%	98.31%
Rata-rata	**89.75%**	**92.00%**	**96.15%**	**94.01%**
Standard Deviasi	0.0485	0.0282	0.0344	0.0284



Gambar 4.11 Boxplot Rata-rata Evaluation Matrix Skenario 11 (5-6-4-2-1)

Hasil pengujian pada Tabel 4.13 menunjukkan bahwa model mencapai *Accuracy* rata-rata sebesar 89,75%, dengan *Precision* sebesar 92,00%, *Recall* sebesar 96,15%, dan *F1-Score* sebesar 94,01%. Nilai *Recall* yang tinggi mengindikasikan bahwa model tetap memiliki sensitivitas yang baik dalam mendeteksi kelas 1 (konsumsi energi 1%), sementara peningkatan nilai *Precision* dibandingkan skenario sebelumnya menunjukkan berkurangnya jumlah prediksi positif palsu (*false positive*).

Jika ditinjau melalui Gambar 4.11, terlihat bahwa distribusi metrik *Precision* dan *F1-Score* relatif rapat dan stabil, yang tercermin dari nilai standar deviasi masing-masing sebesar 0,0282 dan 0,0284. Meskipun metrik *Accuracy* menunjukkan variasi yang sedikit lebih besar, performa keseluruhan model tetap berada dalam rentang yang konsisten antar fold.

4.2.12 Skenario 12 (5-6-4-2-1)

Skenario 12 yang merupakan skenario terakhir yang menggunakan arsitektur yang sama dengan Skenario 10 dan 11 menggunakan *Threshold* paling

ketat yaitu 0.70. Rekapitulasi hasil evaluasi performa model pada masing-masing fold ditampilkan pada Tabel 4.12.

Dari hasil pengujian pada Tabel 4.12, model memperoleh *Accuracy* rata-rata sebesar 88,46%, *Precision* sebesar 92,50%, *Recall* sebesar 93,85%, dan *F1-Score* sebesar 93,11%. Nilai *Recall* yang tetap tinggi menunjukkan bahwa model masih mampu mendeteksi sebagian besar sampel kelas 1 (konsumsi energi 1%), sementara nilai *Precision* yang stabil mengindikasikan bahwa ketepatan prediksi kelas 1 dapat dipertahankan meskipun konfigurasi parameter diubah.

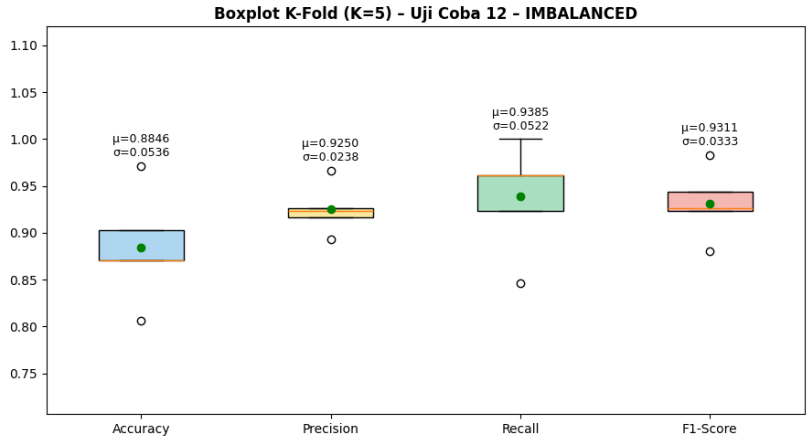
Apabila ditinjau melalui Gambar 4.12, terlihat bahwa distribusi metrik *Precision* memiliki sebaran yang relatif paling rapat dibandingkan metrik lainnya, dengan nilai standar deviasi sebesar 0,0238. Sebaliknya, metrik *Accuracy* dan *Recall* menunjukkan variasi yang lebih besar, yang mencerminkan adanya fluktuasi performa antar fold akibat perbedaan pembagian data.

Tabel 4.14 *Evaluation Matrix* K-Fold Skenario 12

Fold	***Accuracy***	***Precision***	***Recall***	***F1-Score***
1	87.10%	89.29%	96.15%	92.59%
2	90.32%	92.59%	96.15%	94.34%
3	87.10%	92.31%	92.31%	92.31%
4	80.65%	91.67%	84.62%	88.00%
5	97.14%	96.67%	100.00%	98.31%
Rata-rata	**88.46%**	**92.50%**	**93.85%**	**93.11%**
Standard Deviasi	0.0536	0.0238	0.0522	0.0333

Berdasarkan hasil pengujian pada Skenario 10, 11, dan 12, arsitektur 5-6-4-2-1 menunjukkan kemampuan deteksi kelas 1 yang sangat baik pada data imbalanced, yang tercermin dari nilai *Recall* tinggi dan relatif konsisten pada seluruh variasi *threshold*. Peningkatan *threshold* dari 0,50 hingga 0,70 terbukti

mampu meningkatkan nilai *Precision* tanpa menyebabkan penurunan *Recall* yang signifikan, sehingga keseimbangan antara ketepatan dan sensitivitas deteksi tetap terjaga. Secara keseluruhan, hasil ini mengindikasikan bahwa kedalaman jaringan mampu mengimbangi keterbatasan jumlah neuron, menjadikan arsitektur 5-6-4-2-1 cukup stabil dan efektif dalam menangani ketidakseimbangan data.



Gambar 4.12 Boxplot Rata-rata Evaluation Matrix Skenario 12 (5-6-4-2-1)

4.3 Hasil Uji Coba Data *Balanced*

Pada tahap pengujian ini, evaluasi kinerja model dilanjutkan dengan menerapkan teknik *Synthetic Minority Over-sampling Technique* (SMOTE) pada data pelatihan. Langkah ini diimplementasikan sebagai upaya mitigasi terhadap kendala ketidakseimbangan kelas yang teridentifikasi pada pengujian sebelumnya, dengan tujuan menyeimbangkan distribusi data agar model dapat mempelajari pola kelas minoritas secara lebih proporsional. Prosedur evaluasi tetap mempertahankan konsistensi konfigurasi, di mana model diuji kembali menggunakan variasi arsitektur dan *Threshold* yang sama seperti pada tahap sebelumnya.

4.3.1 Skenario 13 (5-8-4-1)

Skenario ini merupakan uji coba pertama yang menerapkan SMOTE dengan *threshold* standar dan arsitektur yang sama dengan Skenario 1 yaitu 5-8-4-1. Konfigurasi arsitektur dan *threshold* yang digunakan pada skenario ini disamakan dengan pengujian sebelumnya yang menggunakan data *imbalanced*, sehingga pengaruh penerapan SMOTE terhadap kinerja model dapat diamati secara lebih objektif. Ringkasan hasil evaluasi performa model pada masing-masing fold disajikan pada Tabel 4.15.

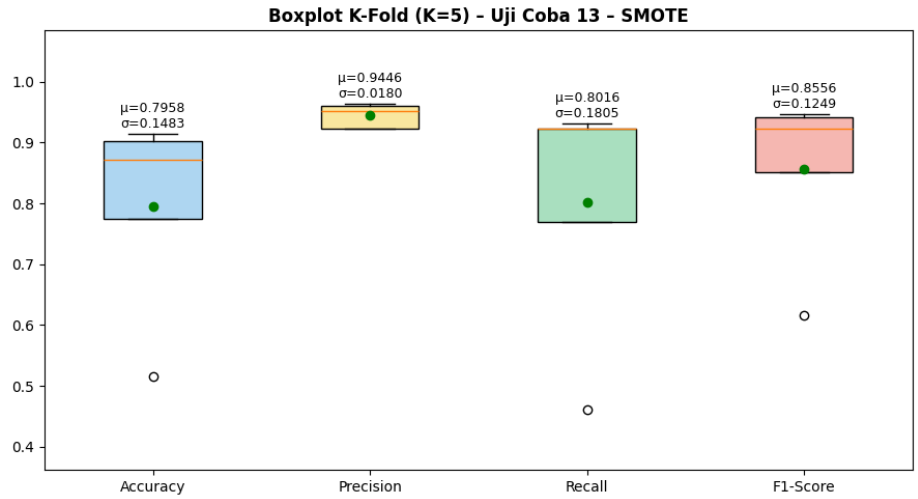
Berdasarkan Tabel 4.15, model menghasilkan rata-rata *Accuracy* sebesar 79,58%, *Precision* sebesar 94,46%, *Recall* sebesar 80,16%, dan *F1-Score* sebesar 85,56%. Nilai *Precision* yang tinggi menunjukkan bahwa model tetap mampu melakukan prediksi kelas positif secara tepat setelah penerapan SMOTE. Namun demikian, nilai *Recall* yang relatif lebih rendah dan variasi performa antar fold mengindikasikan bahwa model belum sepenuhnya konsisten dalam mendeteksi seluruh sampel kelas minoritas, meskipun distribusi data telah diseimbangkan.

Tabel 4.15 *Evaluation Matrix* K-Fold Skenario 13 dengan SMOTE

Fold	***Accuracy***	***Precision***	***Recall***	***F1-Score***
1	77.42%	95.24%	76.92%	85.11%
2	87.10%	92.31%	92.31%	92.31%
3	90.32%	96.00%	92.31%	94.12%
4	51.61%	92.31%	46.15%	61.54%
5	91.43%	96.43%	93.10%	94.74%
Rata-rata	**79.58%**	**94.46%**	**80.16%**	**85.56%**
Standard Deviasi	0.1483	0.018	0.1805	0.1249

Visualisasi pada Gambar 4.13 memperlihatkan bahwa variasi performa cukup besar terjadi pada metrik *Accuracy* dan *Recall*, yang tercermin dari nilai

standar deviasi masing-masing sebesar 0,1483 dan 0,1805. Sebaliknya, metrik *Precision* menunjukkan distribusi yang relatif lebih rapat dengan standar deviasi sebesar 0,018, menandakan bahwa ketepatan prediksi kelas positif tetap terjaga secara konsisten pada setiap fold.



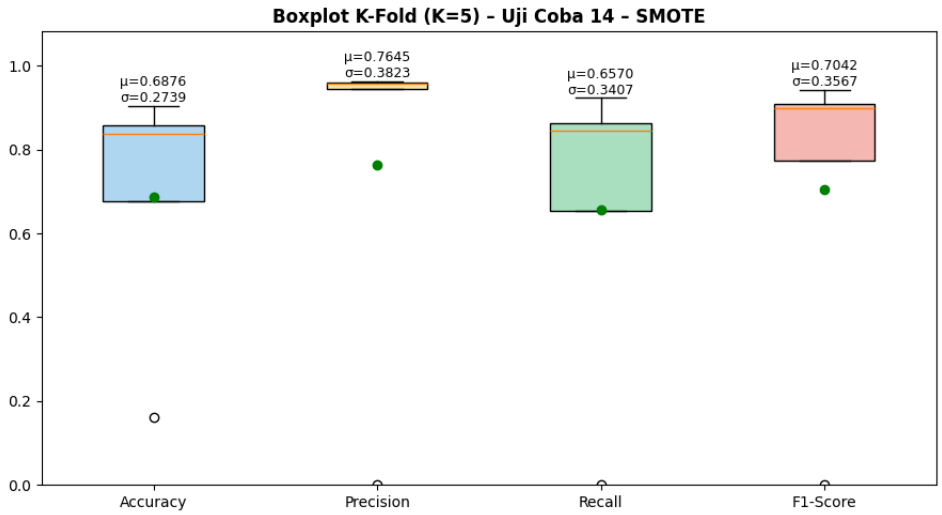
Gambar 4.13 Boxplot Rata-rata Evaluation Matrix Skenario 13 dengan SMOTE (5-8-4-1)

4.3.2 Skenario 14 (5-8-4-1)

Pada skenario ini, konfigurasi arsitektur 5-8-4-1 pada data *Balanced* diuji dengan menaikkan *threshold* menjadi 0.6. Rekapitulasi kinerja model pada masing-masing fold ditampilkan pada Tabel 4.16.

Tabel 4.16 *Evaluation Matrix* K-Fold Skenario 14 dengan SMOTE

Fold	***Accuracy***	***Precision***	***Recall***	***F1-Score***
1	67.74%	94.44%	65.38%	77.27%
2	83.87%	95.65%	84.62%	89.80%
3	90.32%	96.00%	92.31%	94.12%
4	16.13%	0.00%	0.00%	0.00%
5	85.71%	96.15%	86.21%	90.91%
Rata-rata	**68.76%**	**76.45%**	**65.70%**	**70.42%**
Standard Deviasi	0.2739	0.3823	0.3407	0.3567



Gambar 4.14 Boxplot Rata-rata Evaluation Matrix Skenario 14 dengan SMOTE (5-8-4-1)

Hasil evaluasi pada Tabel 4.16 menunjukkan bahwa model memperoleh *Accuracy* rata-rata sebesar 68,76%, *Precision* sebesar 76,45%, *Recall* sebesar 65,70%, dan *F1-Score* sebesar 70,42%. Nilai-nilai tersebut menunjukkan penurunan performa yang cukup signifikan dibandingkan Skenario 13. Penurunan ini terutama dipengaruhi oleh kegagalan model pada salah satu fold, yang menghasilkan nilai *Precision*, *Recall*, dan *F1-Score* sebesar 0%, sehingga berdampak besar terhadap nilai rata-rata keseluruhan.

Apabila ditinjau melalui Gambar 4.14, terlihat bahwa sebaran metrik *Accuracy*, *Precision*, dan *Recall* memiliki variasi yang sangat besar, yang tercermin dari nilai standar deviasi yang tinggi, masing-masing sebesar 0,2739, 0,3823, dan 0,3407. Kondisi ini mengindikasikan bahwa performa model pada skenario ini sangat tidak stabil antar fold, meskipun data telah diseimbangkan menggunakan SMOTE.

4.3.3 Skenario 15 (5-8-4-1)

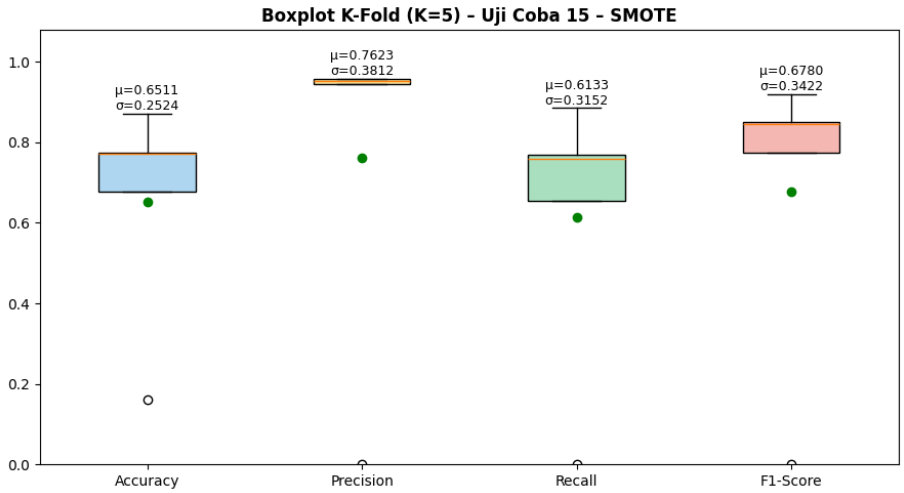
Skenario 15 merupakan skenario terakhir untuk arsitektur 5-8-4-1 dengan penerapan SMOTE, dimana *threshold* dinaikkan ke level yang paling ketat, yaitu 0.7. Rekapitulasi kinerja model pada setiap fold pengujian ditampilkan pada Tabel 4.17, sementara sebaran metrik evaluasi divisualisasikan pada Gambar 4.15.

Hasil pengujian pada Tabel 4.17 menunjukkan bahwa model memperoleh *Accuracy* rata-rata sebesar 65,11%, *Precision* sebesar 76,23%, *Recall* sebesar 61,33%, dan *F1-Score* sebesar 67,80%. Nilai-nilai tersebut memperlihatkan penurunan performa yang lebih lanjut dibandingkan skenario SMOTE sebelumnya. Penurunan ini dipengaruhi oleh kegagalan klasifikasi pada salah satu fold yang menghasilkan nilai *Precision*, *Recall*, dan *F1-Score* sebesar 0%, sehingga berdampak signifikan terhadap nilai rata-rata keseluruhan.

Tabel 4.17 *Evaluation Matrix* K-Fold Skenario 15 dengan SMOTE

Fold	***Accuracy***	***Precision***	***Recall***	***F1-Score***
1	67.74%	94.44%	65.38%	77.27%
2	77.42%	95.24%	76.92%	85.11%
3	87.10%	95.83%	88.46%	92.00%
4	16.13%	0.00%	0.00%	0.00%
5	77.14%	95.65%	75.86%	84.62%
Rata-rata	**65.11%**	**76.23%**	**61.33%**	**67.80%**
Standard Deviasi	0.2524	0.3812	0.3152	0.3422

Jika ditinjau dari Gambar 4.15, terlihat bahwa variasi performa antar fold sangat besar, khususnya pada metrik *Precision* dan *Recall*, yang tercermin dari nilai standar deviasi masing-masing sebesar 0,3812 dan 0,3152. Sebaran yang lebar ini mengindikasikan bahwa proses pembelajaran model pada konfigurasi ini belum stabil meskipun distribusi data telah diseimbangkan.



Gambar 4.15 Boxplot Rata-rata Evaluation Matrix Skenario 15 dengan SMOTE (5-8-4-1)

Berdasarkan hasil pengujian pada Skenario 13–15, arsitektur 5-8-4-1 menunjukkan performa terbaik pada *threshold* standar, dengan nilai *Precision* yang tinggi dan keseimbangan yang relatif baik antara *Precision* dan *Recall*. Peningkatan *threshold* pada skenario berikutnya justru menyebabkan penurunan performa, terutama pada *Recall* dan *F1-Score*, serta meningkatkan variasi hasil antar fold. Hal ini mengindikasikan bahwa arsitektur 5-8-4-1 kurang stabil ketika *threshold* diperketat, meskipun data telah diseimbangkan menggunakan SMOTE.

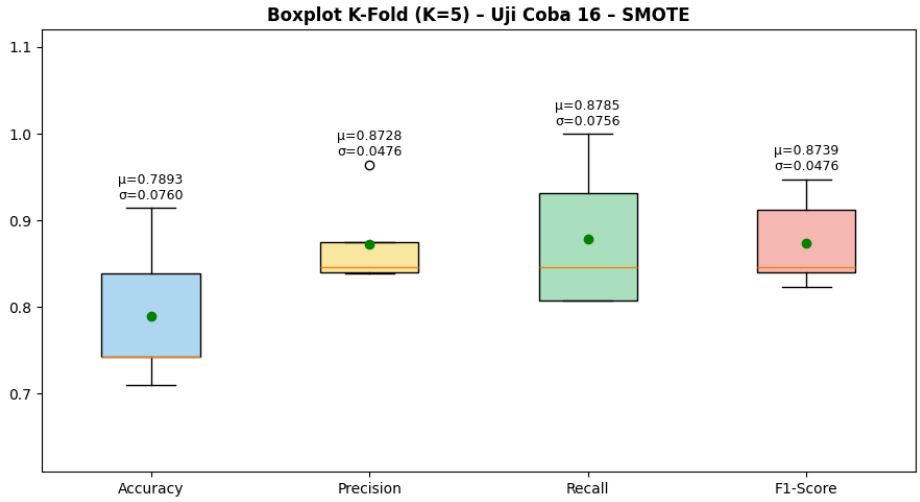
4.3.4 Skenario 16 (5-4-2-1)

Kelompok pengujian berikutnya beralih ke arsitektur yang lebih sederhana (5-4-2-1) dengan penerapan SMOTE dan *threshold* standar 0.5. Konfigurasi arsitektur ini digunakan untuk mengamati pengaruh penyederhanaan jaringan terhadap kinerja model setelah distribusi kelas diseimbangkan. Hasil evaluasi performa model pada setiap fold dirangkum dalam Tabel 4.18, sedangkan pola sebaran metrik evaluasi ditunjukkan pada Gambar 4.16.

Hasil pengujian memperlihatkan bahwa model mencapai *Accuracy* rata-rata sebesar 78,93%, dengan *Precision* sebesar 87,28%, *Recall* sebesar 87,85%, dan *F1-Score* sebesar 87,39%. Nilai *Precision* dan *Recall* yang relatif seimbang menunjukkan bahwa model mampu menjaga ketepatan prediksi sekaligus sensitivitas deteksi kelas 1 (konsumsi energi 1%) setelah penerapan SMOTE, meskipun nilai *Accuracy* masih berada pada tingkat sedang.

Tabel 4.18 *Evaluation Matrix* K-Fold Skenario 16 dengan SMOTE

Fold	***Accuracy***	***Precision***	***Recall***	***F1-Score***
1	74.19%	84.62%	84.62%	84.62%
2	70.97%	84.00%	80.77%	82.35%
3	74.19%	87.50%	80.77%	84.00%
4	83.87%	83.87%	100.00%	91.23%
5	91.43%	96.43%	93.10%	94.74%
Rata-rata	**78.93%**	**87.28%**	**87.85%**	**87.39%**
Standard Deviasi	0.076	0.0476	0.0756	0.0476



Gambar 4.16 Boxplot Rata-rata Evaluation Matrix Skenario 16 dengan SMOTE (5-4-2-1)

Analisis visual pada Gambar 4.16 menunjukkan bahwa metrik *Precision* dan *F1-Score* memiliki tingkat konsistensi yang lebih baik dibandingkan metrik

Accuracy dan *Recall*. Hal ini tercermin dari nilai standar deviasi *Precision* dan *F1-Score* yang lebih rendah, sementara variasi yang lebih tinggi pada *Accuracy* dan *Recall* mengindikasikan ketidakstabilan performa akibat variasi data uji antar fold.

4.3.5 Skenario 17 (5-4-2-1)

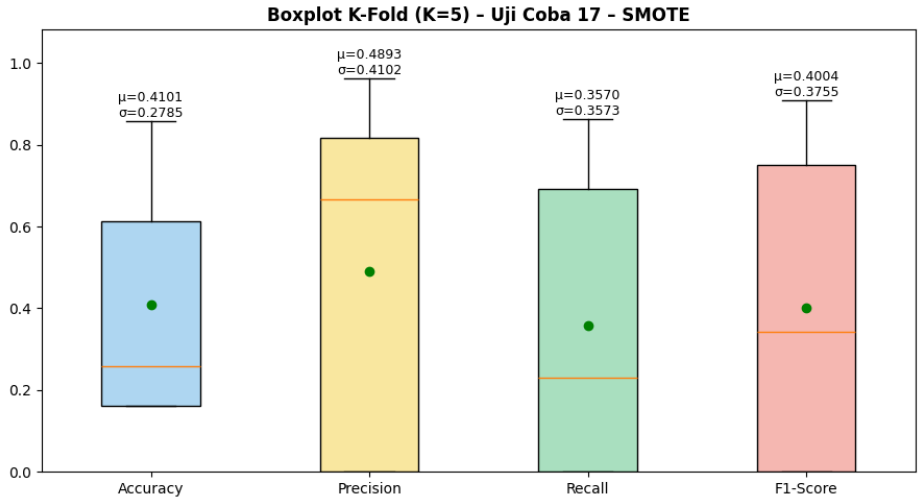
Pada skenario 17, evaluasi dilakukan pada arsitektur yang tetap dengan penerapan SMOTE, namun dengan *threshold* yang ditingkatkan menjadi 0.6. Konfigurasi ini bertujuan untuk mengamati pengaruh pengetatan *threshold* terhadap stabilitas dan performa model setelah sebelumnya arsitektur yang sama pada Skenario 16 menunjukkan hasil yang relatif baik pada *threshold* standar. Ringkasan hasil evaluasi performa model pada masing-masing fold disajikan pada Tabel 4.19.

Tabel 4.19 *Evaluation Matrix* K-Fold Skenario 17 dengan SMOTE

Fold	***Accuracy***	***Precision***	***Recall***	***F1-Score***
1	16.13%	0.00%	0.00%	0.00%
2	25.81%	66.67%	23.08%	34.29%
3	16.13%	0.00%	0.00%	0.00%
4	61.29%	81.82%	69.23%	75.00%
5	85.71%	96.15%	86.21%	90.91%
Rata-rata	**41.01%**	**48.93%**	**35.70%**	**40.04%**
Standard Deviasi	0.2785	0.4102	0.3573	0.3755

Berdasarkan Tabel 4.19, model menghasilkan *Accuracy* rata-rata sebesar 41,01%, *Precision* sebesar 48,93%, *Recall* sebesar 35,70%, dan *F1-Score* sebesar 40,04%. Nilai-nilai tersebut menunjukkan penurunan performa yang signifikan dibandingkan dengan Skenario 16. Penurunan ini dipengaruhi oleh kegagalan

klasifikasi pada beberapa fold yang menghasilkan nilai *Precision*, *Recall*, dan *F1-Score* sebesar 0%, sehingga berdampak besar terhadap nilai rata-rata keseluruhan.



Gambar 4.17 Boxplot Rata-rata Evaluation Matrix Skenario 17 dengan SMOTE (5-4-2-1)

Berdasarkan visualisasi pada Gambar 4.17, terlihat bahwa variasi performa antar fold cukup besar, terutama pada metrik *Precision* dan *F1-Score*, yang tercermin dari nilai standar deviasi masing-masing sebesar 0,4102 dan 0,3755. Hal ini mengindikasikan bahwa peningkatan *threshold* pada arsitektur 5-4-2-1 menyebabkan performa model menjadi kurang stabil, meskipun data telah diseimbangkan menggunakan SMOTE.

4.3.6 Skenario 18 (5-4-2-1)

Skenario 18 merupakan pengujian terakhir untuk arsitektur sederhana (5-4-2-1) dengan data SMOTE, di mana *threshold* dinaikkan ke level maksimum 0.7. Pengujian ini bertujuan untuk mengevaluasi dampak pengetatan *threshold* yang lebih ketat terhadap kinerja model setelah pada Skenario 17 terjadi penurunan

performa. Ringkasan hasil evaluasi performa model pada masing-masing fold disajikan pada Tabel 4.20.

Berdasarkan Tabel 4.20, model menghasilkan *Accuracy* rata-rata sebesar 33,99%, *Precision* sebesar 54,20%, *Recall* sebesar 24,24%, dan *F1-Score* sebesar 28,73%. Hasil ini menunjukkan penurunan performa yang lebih lanjut dibandingkan dengan Skenario 17. Rendahnya nilai *Recall* mengindikasikan bahwa model semakin gagal dalam mendeteksi sampel kelas positif, meskipun pada salah satu fold diperoleh nilai *Precision* yang sangat tinggi.

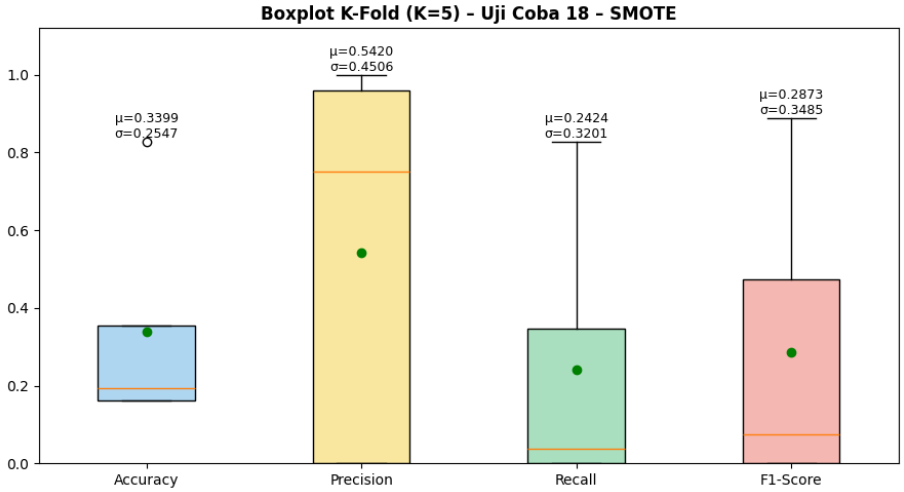
Visualisasi pada Gambar 4.18 menunjukkan bahwa variasi performa antar fold masih tergolong tinggi, khususnya pada metrik *Precision* dan *F1-Score*, yang tercermin dari nilai standar deviasi masing-masing sebesar 0,4506 dan 0,3485. Hal ini mengindikasikan bahwa peningkatan *threshold* hingga 0,7 semakin menurunkan stabilitas dan sensitivitas model, meskipun distribusi data telah diseimbangkan menggunakan SMOTE.

Tabel 4.20 *Evaluation Matrix* K-Fold Skenario 18 dengan SMOTE

Fold	***Accuracy***	***Precision***	***Recall***	***F1-Score***
1	16.13%	0.00%	0.00%	0.00%
2	19.35%	100.00%	3.85%	7.41%
3	16.13%	0.00%	0.00%	0.00%
4	35.48%	75.00%	34.62%	47.37%
5	82.86%	96.00%	82.76%	88.89%
Rata-rata	**33.99%**	**54.20%**	**24.24%**	**28.73%**
Standard Deviasi	0.2547	0.4506	0.3201	0.3485

Berdasarkan hasil pengujian pada Skenario 16, 17, dan 18, arsitektur 5-4-2-1 dengan penerapan SMOTE menunjukkan performa terbaik pada threshold standar (0,5), dengan keseimbangan yang relatif baik antara Precision dan Recall.

Peningkatan threshold menjadi 0,6 dan 0,7 justru menyebabkan penurunan performa yang signifikan, terutama pada metrik *Recall* dan *F1-Score*, serta meningkatkan variasi hasil antar fold. Hal ini mengindikasikan bahwa pada arsitektur yang lebih sederhana, pengetatan *threshold* membuat model terlalu konservatif dalam mendeteksi kelas positif, meskipun distribusi data telah diseimbangkan. Dengan demikian, arsitektur 5-4-2-1 lebih optimal digunakan pada *threshold* standar setelah penerapan SMOTE.



Gambar 4.18 Boxplot Rata-rata Evaluation Matrix Skenario 18 dengan SMOTE (5-4-2-1)

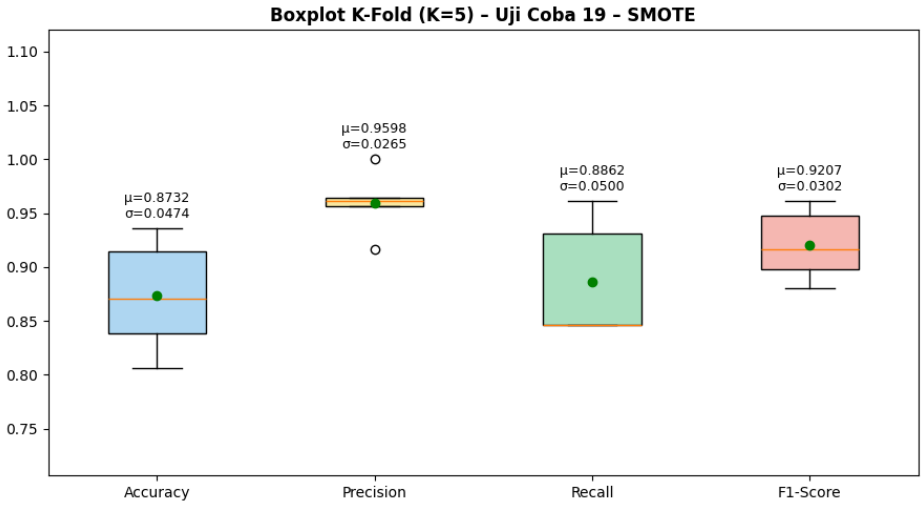
4.3.7 Skenario 19 (5-8-6-4-1)

Skenario 19 merupakan pengujian lanjutan menggunakan arsitektur 5-8-6-4-1 dengan penerapan SMOTE dan *threshold* standar 0,5. Pengujian ini bertujuan untuk mengevaluasi kinerja arsitektur yang lebih kompleks setelah distribusi kelas diseimbangkan, serta membandingkannya dengan arsitektur yang lebih sederhana pada skenario sebelumnya. Ringkasan hasil evaluasi performa model pada masing-masing fold disajikan pada Tabel 4.21.

Tabel 4.21 *Evaluation Matrix* K-Fold Skenario 19 dengan SMOTE

Fold	*Accuracy*	*Precision*	*Recall*	*F1-Score*
1	83.87%	95.65%	84.62%	89.80%
2	80.65%	91.67%	84.62%	88.00%
3	93.55%	96.15%	96.15%	96.15%
4	87.10%	100.00%	84.62%	91.67%
5	91.43%	96.43%	93.10%	94.74%
Rata-rata	**87.32%**	**95.98%**	**88.62%**	**92.07%**
Standard Deviasi	0.0474	0.0265	0.05	0.0302

Berdasarkan Tabel 4.21, model menghasilkan *Accuracy* rata-rata sebesar 87,32%, *Precision* sebesar 95,98%, *Recall* sebesar 88,62%, dan *F1-Score* sebesar 92,07%. Hasil ini menunjukkan bahwa arsitektur 5-8-6-4-1 mampu menjaga keseimbangan yang baik antara ketepatan dan sensitivitas deteksi setelah penerapan SMOTE. Nilai *Precision* yang tinggi mengindikasikan bahwa model sangat efektif dalam meminimalkan prediksi positif palsu, sementara nilai *Recall* yang relatif tinggi menunjukkan kemampuan deteksi kelas positif yang tetap optimal.



Gambar 4.19 Boxplot Rata-rata Evaluation Matrix Skenario 19 dengan SMOTE (5-8-6-4-1)

Berdasarkan visualisasi pada Gambar 4.19, terlihat bahwa sebaran metrik evaluasi relatif rapat dan stabil, khususnya pada metrik *Precision* dan *F1-Score*, yang tercermin dari nilai standar deviasi masing-masing sebesar 0,0265 dan 0,0302. Kondisi ini menunjukkan bahwa arsitektur 5-8-6-4-1 memiliki stabilitas performa yang baik antar fold setelah distribusi data diseimbangkan menggunakan SMOTE.

4.3.8 Skenario 20 (5-8-6-4-1)

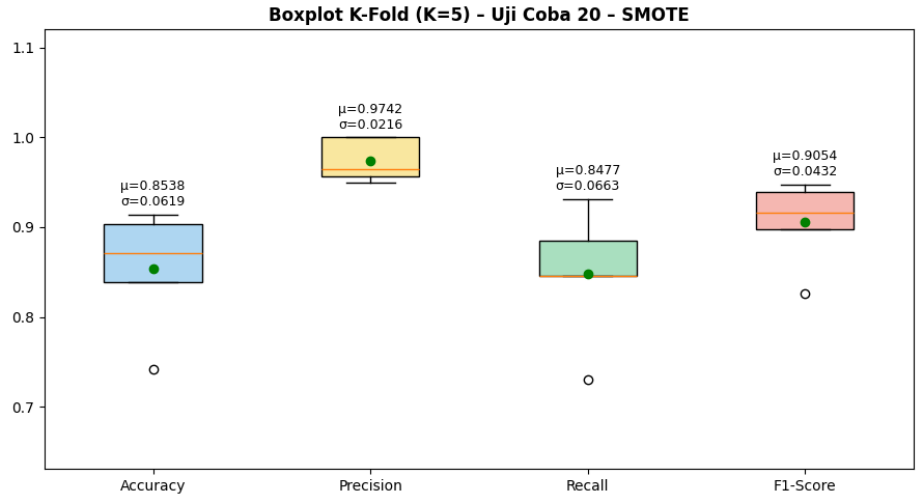
Skenario 20 merupakan pengujian lanjutan pada arsitektur 5-8-6-4-1 dengan penerapan SMOTE dan peningkatan nilai *threshold* menjadi 0,6. Pengujian ini bertujuan untuk mengamati pengaruh pengetatan *threshold* terhadap performa model pada arsitektur yang lebih kompleks setelah pada Skenario 19 diperoleh hasil yang stabil pada *threshold* standar. Ringkasan hasil evaluasi performa model pada masing-masing fold disajikan pada Tabel 4.22.

Tabel 4.22 *Evaluation Matrix* K-Fold Skenario 20 dengan SMOTE

Fold	***Accuracy***	***Precision***	***Recall***	***F1-Score***
1	83.87%	95.65%	84.62%	89.80%
2	74.19%	95.00%	73.08%	82.61%
3	90.32%	100.00%	88.46%	93.88%
4	87.10%	100.00%	84.62%	91.67%
5	80.00%	95.83%	79.31%	86.79%
Rata-rata	**83.10%**	**97.30%**	**82.02%**	**88.95%**
Standard Deviasi	0.0626	0.0238	0.0596	0.0436

Berdasarkan Tabel 4.22, model menghasilkan *Accuracy* rata-rata sebesar 83,10%, *Precision* sebesar 97,30%, *Recall* sebesar 82,02%, dan *F1-Score* sebesar 88,95%. Dibandingkan dengan Skenario 19, peningkatan *threshold* menyebabkan penurunan nilai *Recall* dan *F1-Score*, meskipun nilai *Precision* meningkat dan tetap

berada pada tingkat yang sangat tinggi. Hal ini menunjukkan bahwa model menjadi lebih selektif dalam memprediksi kelas positif, namun dengan konsekuensi berkurangnya sensitivitas deteksi.



Gambar 4.20 Boxplot Rata-rata Evaluation Matrix Skenario 20 dengan SMOTE (5-8-6-4-1)

Berdasarkan visualisasi pada Gambar 4.20, terlihat bahwa sebaran metrik *Precision* relatif rapat dan stabil, yang tercermin dari nilai standar deviasi sebesar 0,0238. Sebaliknya, variasi yang lebih besar pada metrik *Recall* dan *F1-Score* mengindikasikan bahwa pengetatan *threshold* mulai memengaruhi konsistensi performa model antar fold, meskipun secara umum kinerja model masih berada pada tingkat yang baik.

4.3.9 Skenario 21 (5-8-6-4-1)

Skenario 21 merupakan pengujian terakhir pada arsitektur 5-8-6-4-1 dengan penerapan SMOTE, di mana nilai *threshold* ditingkatkan menjadi 0,7. Pengujian ini bertujuan untuk mengevaluasi dampak pengetatan *threshold* yang lebih ketat terhadap kinerja dan stabilitas model setelah pada Skenario 19 dan 20 diperoleh

hasil yang relatif baik pada *threshold* yang lebih rendah. Ringkasan hasil evaluasi performa model pada masing-masing fold disajikan pada Tabel 4.23.

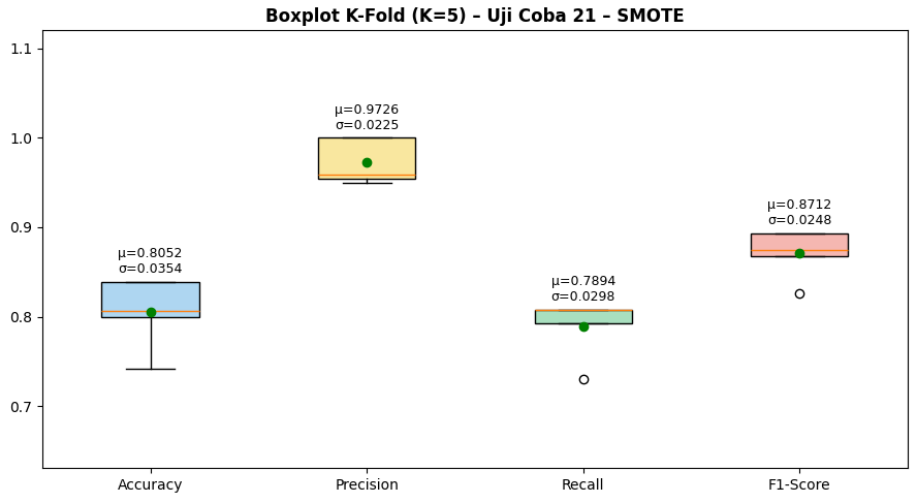
Tabel 4.23 *Evaluation Matrix* K-Fold Skenario 21 dengan SMOTE

Fold	***Accuracy***	***Precision***	***Recall***	***F1-Score***
1	80.65%	95.45%	80.77%	87.50%
2	74.19%	95.00%	73.08%	82.61%
3	83.87%	100.00%	80.77%	89.36%
4	83.87%	100.00%	80.77%	89.36%
5	80.00%	95.83%	79.31%	86.79%
Rata-rata	**80.52%**	**97.26%**	**78.94%**	**87.12%**
Standard Deviasi	0.0354	0.0225	0.0298	0.0248

Mengacu pada Tabel 4.23, model pada Skenario 21 mencapai *Accuracy* rata-rata sebesar 80,52%, dengan *Precision* 97,26%, *Recall* 78,94%, dan *F1-Score* 87,12%. Jika dibandingkan dengan Skenario 19 dan 20, penetapan *threshold* yang lebih ketat berdampak pada penurunan kemampuan deteksi kelas positif, yang tercermin dari menurunnya nilai *Recall* dan *F1-Score*. Meskipun demikian, tingginya nilai *Precision* menunjukkan bahwa model tetap konsisten dalam menghasilkan prediksi positif yang tepat, meskipun sensitivitasnya mengalami penurunan.

Berdasarkan hasil pengujian pada Skenario 19, 20, dan 21, arsitektur 5-8-6-4-1 dengan penerapan SMOTE menunjukkan kinerja yang stabil dan unggul dalam mendeteksi kelas positif, khususnya pada *threshold* standar (0,5) yang menghasilkan keseimbangan optimal antara *Precision* dan *Recall*. Peningkatan *threshold* hingga 0,6 dan 0,7 secara konsisten meningkatkan nilai *Precision*, namun diikuti oleh penurunan *Recall* dan *F1-Score* yang menunjukkan berkurangnya sensitivitas deteksi. Meskipun demikian, rendahnya variasi performa antar fold

mengindikasikan bahwa arsitektur ini memiliki stabilitas pembelajaran yang baik setelah distribusi data diseimbangkan. Secara keseluruhan, arsitektur 5-8-6-4-1 dengan SMOTE paling optimal digunakan pada *threshold* standar, serta layak dipertimbangkan sebagai arsitektur terbaik dalam menangkap pola data konsumsi energi pada kondisi data seimbang.



Gambar 4.21 Boxplot Rata-rata Evaluation Matrix Skenario 21 dengan SMOTE (5-8-6-4-1)

4.3.10 Skenario 22 (5-6-4-2-1)

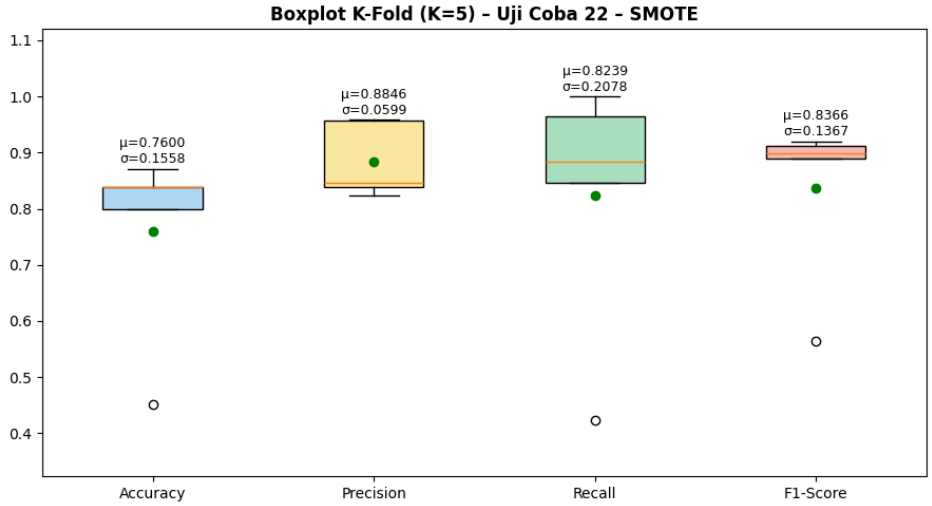
Kelompok pengujian terakhir untuk data *Balanced* beralih ke arsitektur dengan struktur *bottleneck* (5-6-4-2-1). Pada Skenario 22 ini, model diuji dengan *threshold* standar 0.5. Tujuannya adalah untuk melihat apakah arsitektur yang menyempit di layer akhir mampu menangani data sintetis SMOTE sebaik arsitektur yang lebih lebar. Ringkasan hasil evaluasi performa model pada masing-masing fold disajikan pada Tabel 4.24.

Berdasarkan Tabel 4.24, model menghasilkan *Accuracy* rata-rata sebesar 76,00%, *Precision* sebesar 88,46%, *Recall* sebesar 82,39%, dan *F1-Score* sebesar 83,66%. Hasil ini menunjukkan bahwa arsitektur 5-6-4-2-1 masih mampu

mempertahankan kemampuan deteksi kelas positif yang cukup baik setelah penerapan SMOTE, meskipun performanya belum seoptimal arsitektur yang lebih kompleks. Rendahnya nilai *Accuracy* pada salah satu fold menyebabkan penurunan nilai rata-rata dan meningkatkan variasi performa secara keseluruhan.

Tabel 4.24 *Evaluation Matrix* K-Fold Skenario 22 dengan SMOTE

Fold	***Accuracy***	***Precision***	***Recall***	***F1-Score***
1	83.87%	95.65%	84.62%	89.80%
2	87.10%	95.83%	88.46%	92.00%
3	45.16%	84.62%	42.31%	56.41%
4	83.87%	83.87%	100.00%	91.23%
5	80.00%	82.35%	96.55%	88.89%
Rata-rata	**76.00%**	**88.46%**	**82.39%**	**83.66%**
Standard Deviasi	0.1558	0.0599	0.2078	0.1367



Gambar 4.22 Boxplot Rata-rata Evaluation Matrix Skenario 22 dengan SMOTE (5-6-4-2-1)

Berdasarkan visualisasi pada Gambar 4.22, terlihat bahwa variasi performa antar fold cukup besar, khususnya pada metrik *Accuracy* dan *Recall*, yang tercermin dari nilai standar deviasi masing-masing sebesar 0,1558 dan 0,2078. Kondisi ini mengindikasikan bahwa meskipun SMOTE mampu memperbaiki distribusi data,

stabilitas performa arsitektur 5-6-4-2-1 masih dipengaruhi oleh komposisi data uji pada setiap fold.

4.3.11 Skenario 23 (5-6-4-2-1)

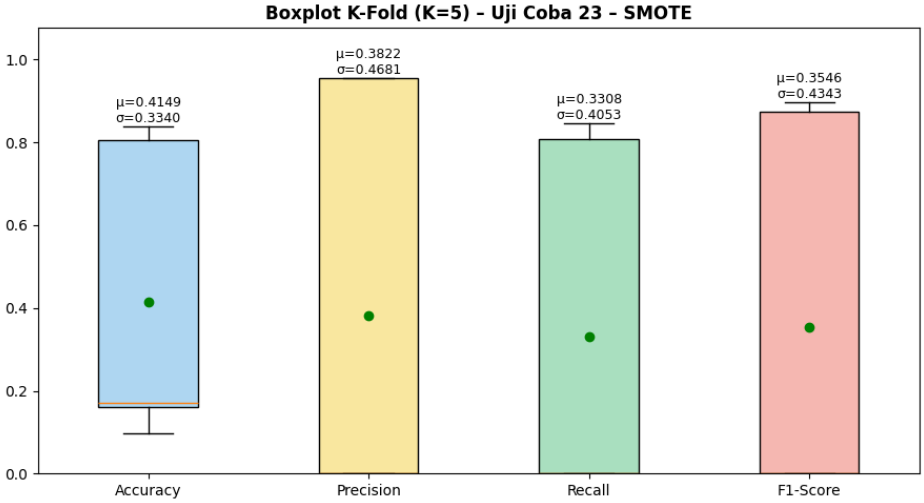
Skenario 23 melanjutkan pengujian pada arsitektur 3 *Hidden Layer* yang mengerucut yaitu 5-6-4-2-1 dengan menaikkan *threshold* menjadi 0.6. Pengujian ini bertujuan untuk mengamati dampak pengetatan *threshold* terhadap kinerja dan stabilitas model setelah pada Skenario 22 diperoleh hasil yang cukup baik pada *threshold* standar. Ringkasan hasil evaluasi performa model pada masing-masing fold disajikan pada Tabel 4.25.

Mengacu pada Tabel 4.25, kinerja model pada Skenario 23 mengalami penurunan yang sangat tajam, dengan *Accuracy* rata-rata sebesar 41,49%, *Precision* 38,22%, *Recall* 33,08%, dan *F1-Score* 35,46%. Rendahnya nilai rata-rata tersebut disebabkan oleh kegagalan klasifikasi pada beberapa fold, di mana model tidak mampu mendeteksi kelas positif sama sekali sehingga menghasilkan nilai *Precision* dan *Recall* sebesar 0%. Kondisi ini menunjukkan bahwa pengetatan *threshold* pada arsitektur 5-6-4-2-1 berdampak negatif terhadap kemampuan deteksi model meskipun distribusi data telah diseimbangkan menggunakan SMOTE.

Tabel 4.25 *Evaluation Matrix* K-Fold Skenario 23 dengan SMOTE

Fold	***Accuracy***	***Precision***	***Recall***	***F1-Score***
1	80.65%	95.45%	80.77%	87.50%
2	83.87%	95.65%	84.62%	89.80%
3	9.68%	0.00%	0.00%	0.00%
4	16.13%	0.00%	0.00%	0.00%
5	17.14%	0.00%	0.00%	0.00%
Rata-rata	**41.49%**	**38.22%**	**33.08%**	**35.46%**
Standard Deviasi	0.334	0.4681	0.4053	0.4343

Berdasarkan Gambar 4.23, sebaran metrik evaluasi menunjukkan fluktuasi yang sangat tinggi antar fold, khususnya pada metrik *Precision* dan *F1-Score*, yang tercermin dari nilai standar deviasi masing-masing sebesar 0,4681 dan 0,4343. Hal ini mengindikasikan bahwa pada konfigurasi ini, stabilitas performa model tidak dapat dipertahankan, sehingga arsitektur 5-6-4-2-1 menjadi kurang andal ketika *threshold* diperketat setelah penerapan SMOTE.



Gambar 4.23 Boxplot Rata-rata Evaluation Matrix Skenario 23 dengan SMOTE (5-6-4-2-1)

4.3.12 Skenario 24 (5-6-4-2-1)

Skenario 24 merupakan pengujian terakhir pada arsitektur 5-6-4-2-1 dengan penerapan SMOTE dan peningkatan nilai *threshold* menjadi 0,7. Pengujian ini dilakukan untuk mengevaluasi dampak pengetatan *threshold* yang paling ketat terhadap kinerja model setelah pada Skenario 22 dan 23 terjadi fluktuasi performa yang cukup signifikan. Ringkasan hasil evaluasi performa model pada masing-masing fold disajikan pada Tabel 4.26.

Pada Skenario 24, kinerja model mengalami penurunan yang cukup drastis setelah nilai *threshold* dinaikkan menjadi 0,7. Berdasarkan Tabel 4.26, model hanya

mampu mencapai *Accuracy* rata-rata sebesar 38,27%, dengan *Precision* 39,00%, *Recall* 26,92%, dan *F1-Score* 31,76%. Rendahnya nilai *Recall* menunjukkan bahwa sebagian besar sampel kelas positif tidak berhasil terdeteksi, bahkan pada beberapa fold model gagal menghasilkan prediksi positif sama sekali. Kondisi ini menegaskan bahwa pada arsitektur 5-6-4-2-1, penetapan *threshold* yang terlalu ketat setelah penerapan SMOTE justru membatasi kemampuan model dalam mengenali pola kelas minoritas.

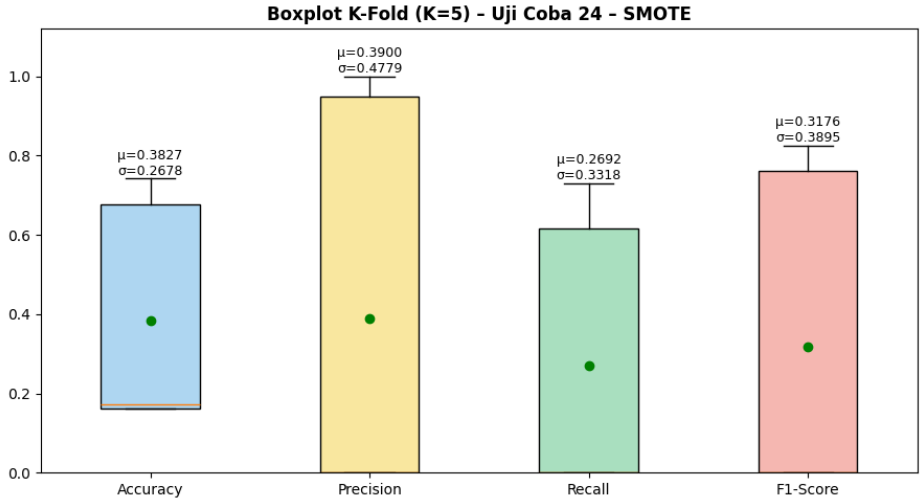
Ditinjau dari Gambar 4.24, distribusi metrik evaluasi memperlihatkan tingkat variasi yang tinggi antar fold, terutama pada metrik *Precision* dan *F1-Score*, yang masing-masing memiliki standar deviasi sebesar 0,4779 dan 0,3895. Sebaran yang lebar ini mengindikasikan bahwa performa model menjadi tidak konsisten, sehingga konfigurasi *threshold* 0,7 pada arsitektur 5-6-4-2-1 kurang sesuai digunakan meskipun data telah diseimbangkan menggunakan SMOTE.

Tabel 4.26 *Evaluation Matrix* K-Fold Skenario 24 dengan SMOTE

Fold	***Accuracy***	***Precision***	***Recall***	***F1-Score***
1	67.74%	100.00%	61.54%	76.19%
2	74.19%	95.00%	73.08%	82.61%
3	16.13%	0.00%	0.00%	0.00%
4	16.13%	0.00%	0.00%	0.00%
5	17.14%	0.00%	0.00%	0.00%
Rata-rata	**38.27%**	**39.00%**	**26.92%**	**31.76%**
Standard Deviasi	0.2678	0.4779	0.3318	0.3895

Hasil pengujian pada Skenario 22 hingga 24 menunjukkan bahwa arsitektur 5-6-4-2-1 dengan penerapan SMOTE memiliki performa yang cukup baik pada *threshold* standar, namun mengalami degradasi kinerja yang signifikan ketika nilai *threshold* diperketat. Pada *threshold* 0,5, model masih mampu mempertahankan

keseimbangan antara ketepatan dan kemampuan deteksi, sedangkan peningkatan *threshold* menjadi 0,6 dan 0,7 menyebabkan penurunan tajam pada metrik *Recall* dan *F1-Score* serta meningkatnya variasi performa antar fold. Temuan ini mengindikasikan bahwa pada arsitektur dengan kompleksitas menengah, pengetatan *threshold* setelah proses penyeimbangan data justru mengurangi efektivitas pembelajaran, sehingga konfigurasi *threshold* standar menjadi pilihan yang paling stabil dan representatif untuk arsitektur 5-6-4-2-1.



Gambar 4.24 Boxplot Rata-rata Evaluation Matrix Skenario 24 dengan SMOTE (5-6-4-2-1)

4.4 Pembahasan

Penelitian ini bertujuan untuk mengukur performa model dari Multi-Layer Percceptron dalam mendeteksi konsumsi energi pada navigasi cerdas kendaraan listrik berdasarkan metrik *Accuracy*, *Precision*, *Recall*, dan *F1-Score* serta menganalisis stabilitas performa melalui rentang nilai *Accuracy* K-Fold dan standar deviasi. Evaluasi dilakukan melalui skema K-Fold Cross Validation pada dua kondisi data, yaitu tanpa penerapan SMOTE dan dengan penerapan SMOTE, sebagaimana disajikan pada Tabel 4.27 dan Tabel 4.28.

Hasil pengujian menunjukkan bahwa model MLP mampu melakukan deteksi konsumsi energi pada navigasi cerdas kendaraan listrik dengan tingkat *Accuracy* yang bervariasi sesuai dengan konfigurasi arsitektur, jumlah neuron, serta penentuan ambang batas keputusan (*Threshold*) yang digunakan. Hasil evaluasi dari pengujian skenario dengan data tanpa SMOTE dan data SMOTE disajikan pada Tabel 4.27 dan Tabel 4.28.

Tabel 4.27 Hasil Evaluasi K-Fold Cross Validation tanpa Metode SMOTE

Skenario	Jumlah Neuron	Thr	Acc K-Fold Terendah	Acc K-Fold Tertinggi	Rata-rata Acc	Std Dev
1	5-8-4-1	0.5	87,83%	94,41%	91,12%	0.0329
2		0.6	87,74%	94,34%	91,04%	0.033
3		0.7	86,79%	94,15%	90,47%	0.0368
4	5-4-2-1	0.5	81,61%	89,59%	85,60%	0.0399
5		0.6	82,56%	93,22%	87,89%	0.0533
6		0.7	80,12%	91,78%	85,95%	0.0583
7	5-8-6-4-1	0.5	87,16%	94,92%	91,04%	0.0388
8		0.6	87,16%	94,92%	91,04%	0.0388
9		0.7	84,10%	95,40%	89,75%	0.0565
10	5-6-4-2-1	0.5	82,56%	90,64%	86,60%	0.0404
11		0.6	84,90%	94,60%	89,75%	0.0485
12		0.7	83,10%	93,82%	88,46%	0.0536

Berdasarkan hasil pengujian pada Tabel 4.27, model MLP menunjukkan performa yang relatif tinggi dan stabil pada seluruh konfigurasi arsitektur dan *threshold*. Nilai rata-rata *Accuracy* berada pada rentang 85,60% hingga 91,12%, dengan standar deviasi yang relatif kecil (≤ 0,0583), yang mengindikasikan konsistensi performa antar fold. Konfigurasi arsitektur 5-8-4-1 dan 5-8-6-4-1 menunjukkan kinerja terbaik, khususnya pada *threshold* 0,5 dan 0,6, dengan rata-rata *Accuracy* mencapai 91,12% dan 91,04%. Selain itu, rentang nilai *Accuracy* K-

Fold yang tidak terlalu lebar menunjukkan bahwa model mampu beradaptasi dengan baik terhadap variasi data uji meskipun distribusi kelas tidak seimbang.

Sebaliknya, hasil pengujian untuk data dengan penerapan SMOTE yang disajikan pada Tabel 4.28 menunjukkan pola performa yang lebih bervariasi antar konfigurasi. Pada beberapa arsitektur, khususnya 5-4-2-1 dan 5-6-4-2-1, peningkatan nilai *threshold* justru menyebabkan penurunan rata-rata *Accuracy* yang signifikan, disertai dengan standar deviasi yang tinggi. Hal ini tercermin pada Skenario 17, 18, 23, dan 24, yang memiliki rata-rata *Accuracy* di bawah 45% serta standar deviasi di atas 0,25, menandakan ketidakstabilan performa model antar fold meskipun data telah diseimbangkan.

Tabel 4.28 Hasil Evaluasi K-Fold Cross Validation menggunakan Metode SMOTE

Skenario	Jumlah Neuron	Thr	Acc K-Fold Terendah	Acc K-Fold Tertinggi	Rata-rata Acc	Std Dev
13	5-8-4-1	0.5	64,75%	94,41%	79,58%	0.1483
14		0.6	41,37%	96,15%	68,76%	0.2739
15		0.7	39,87%	90,35%	65,11%	0.2524
16	5-4-2-1	0.5	71,33%	86,53%	78,93%	0.076
17		0.6	13,16%	68,86%	41,01%	0.2785
18		0.7	8,52%	59,46%	33,99%	0.2547
19	5-8-6-4-1	0.5	82,58%	92,06%	87,32%	0.0474
20		0.6	79,19%	91,57%	85,38%	0.0619
21		0.7	76,98%	84,06%	80,52%	0.0354
22	5-6-4-2-1	0.5	60,42%	91,58%	76,00%	0.1558
23		0.6	8,09%	74,89%	41,49%	0.334
24		0.7	11,49%	65,05%	38,27%	0.2678

4.4.1 Pengaruh Arsitektur MLP (*Hidden Layer* dan *Neuron*)

Berdasarkan rangkaian uji coba yang telah dilakukan pada 24 skenario pengujian, evaluasi kinerja model selanjutnya dikelompokkan berdasarkan konfigurasi arsitektur jaringan Multilayer Perceptron (MLP) yang digunakan.

Analisis ini bertujuan untuk mengkaji pengaruh jumlah hidden layer dan jumlah neuron terhadap kemampuan model dalam mempelajari pola data. Rangkuman performa model untuk arsitektur dengan dua hidden layer dan tiga hidden layer masing-masing disajikan pada Tabel 4.29 dan Tabel 4.30, sedangkan pola sebaran performa divisualisasikan melalui Gambar 4.25.

Tabel 4.29 Rangkuman Nilai Accuracy dan *F1-Score* pada Arsitektur MLP Dua Hidden Layer

No.	**5-8-4-1**			**5-4-2-1**		
	Skenario	**Accuracy**	***F1-Score***	**Skenario**	**Accuracy**	***F1-Score***
1	1	91,12%	94.80%	4	85,60%	91.81%
2	2	91,04%	94.68%	5	87,89%	92.77%
3	3	90,47%	94.15%	6	85,95%	91.45%
4	13	79,58%	85.56%	16	78,93%	87.39%
5	14	68,76%	70.42%	17	41,01%	40.04%
6	15	65,11%	67.80%	18	33,99%	28.73%
	Rata-Rata	**81.01%**	**84.57%**	**Rata-Rata**	**68.90%**	**72.03%**

Berdasarkan Tabel 4.29, arsitektur dengan dua hidden layer menunjukkan perbedaan performa yang cukup signifikan antara konfigurasi 5-8-4-1 dan 5-4-2-1. Arsitektur 5-8-4-1 memperoleh rata-rata accuracy sebesar 81,01% dan *F1-Score* sebesar 84,57%, sedangkan arsitektur 5-4-2-1 hanya mencapai rata-rata accuracy sebesar 68,90% dan *F1-Score* sebesar 72,03%. Selisih ini mengindikasikan bahwa lebar jaringan pada hidden layer awal berperan penting dalam menjaga performa model, khususnya ketika dihadapkan pada variasi skenario pengujian.

Selain itu, terlihat bahwa arsitektur 5-4-2-1 mengalami penurunan performa yang cukup tajam pada beberapa skenario, yang tercermin dari rendahnya nilai accuracy dan *F1-Score* pada pengujian dengan data seimbang. Hal ini menunjukkan bahwa struktur jaringan yang terlalu sempit cenderung memiliki keterbatasan kapasitas representasi dalam mempelajari pola data yang lebih kompleks.

Tabel 4.30 Rangkuman Nilai Accuracy dan *F1-Score* pada Arsitektur MLP Tiga Hidden Layer

No.	5-8-6-4-1			5-6-4-2-1		
	Skenario	**Accuracy**	***F1-Score***	**Skenario**	**Accuracy**	***F1-Score***
1	7	91,04%	94.83%	10	86,60%	92.53%
2	8	91,04%	94.71%	11	89,75%	94.01%
3	9	89,75%	93.65%	12	88,46%	93.11%
4	19	87,32%	92.07%	22	76,00%	83.66%
5	20	85,38%	88.95%	23	41,49%	35.46%
6	21	80,52%	87.12%	24	38,27%	31.76%
	Rata-Rata	**87.51%**	**91.89%**	**Rata-Rata**	**70.10%**	**71.76%**

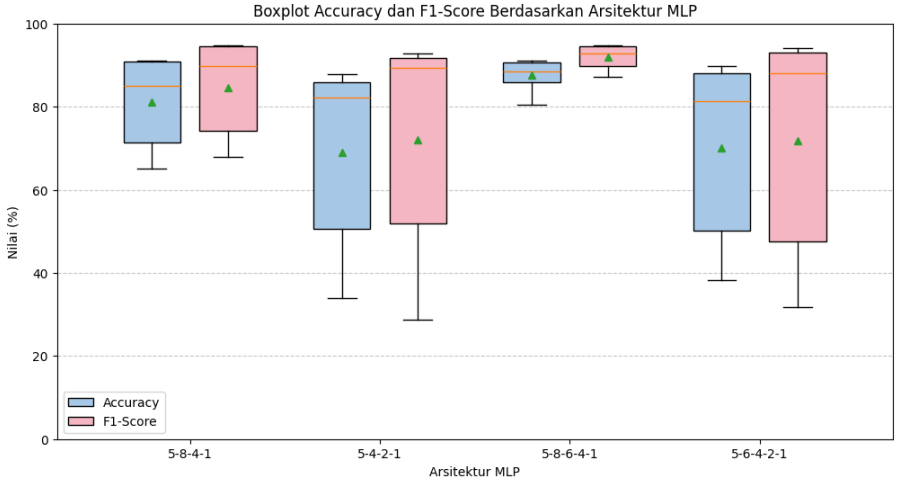
Selanjutnya, hasil evaluasi pada Tabel 4.30 menunjukkan bahwa arsitektur dengan tiga hidden layer memberikan performa yang lebih konsisten, terutama pada konfigurasi 5-8-6-4-1. Arsitektur ini mencapai rata-rata accuracy sebesar 87,51% dan *F1-Score* sebesar 91,89%, yang merupakan nilai tertinggi di antara seluruh konfigurasi yang diuji. Sebaliknya, arsitektur 5-6-4-2-1 hanya mencatatkan rata-rata accuracy sebesar 70,10% dan *F1-Score* sebesar 71,76%, meskipun memiliki jumlah hidden layer yang sama.

Perbedaan ini menunjukkan bahwa penambahan kedalaman jaringan tidak secara otomatis meningkatkan performa, terutama apabila diikuti oleh penyempitan jumlah neuron yang signifikan pada hidden layer. Dengan kata lain, kedalaman jaringan perlu diimbangi dengan kapasitas neuron yang memadai agar model mampu menangkap kompleksitas pola data.

Pola perbedaan performa antar arsitektur tersebut semakin diperjelas melalui visualisasi boxplot pada Gambar 4.25, yang menampilkan sebaran nilai Accuracy dan *F1-Score* untuk keempat arsitektur MLP. Dari visualisasi tersebut terlihat bahwa arsitektur 5-8-6-4-1 memiliki median tertinggi dengan rentang sebaran yang paling sempit, menandakan stabilitas performa yang baik pada kedua

metrik evaluasi. Arsitektur 5-8-4-1 juga menunjukkan performa yang relatif stabil dengan variasi yang lebih terkendali dibandingkan dua arsitektur lainnya.

Berdasarkan hasil pengujian yang dilakukan dalam penelitian ini, arsitektur 5-8-4-1 menunjukkan performa yang paling konsisten dan seimbang, sehingga dipilih sebagai arsitektur terbaik dalam konteks penelitian ini untuk pengembangan dan evaluasi lanjutan sistem deteksi konsumsi energi pada navigasi cerdas kendaraan listrik.



Gambar 4.25 *Boxplot* Perbandingan *Accuracy* dan *F1-Score* Arsitektur MLP

4.4.2 Pengaruh Keseimbangan Data (*Class Imbalanced*)

Selain konfigurasi arsitektur jaringan, faktor penting lain yang dianalisis dalam penelitian ini adalah kondisi keseimbangan data. Analisis ini membandingkan kinerja model pada kondisi dataset asli yang tidak seimbang (*Imbalanced*) dengan dataset yang telah diseimbangkan menggunakan teknik *Synthetic Minority Over-sampling Technique* (SMOTE). Rekapitulasi rata-rata nilai Accuracy dan *F1-Score* dari seluruh skenario pengujian pada kedua kondisi tersebut disajikan pada Tabel 4.31.

Berdasarkan Tabel 4.31, terlihat bahwa pada kondisi data Imbalanced, model MLP secara umum mampu mencapai performa yang tinggi dan relatif konsisten, dengan rata-rata Accuracy sebesar 88,68% dan *F1-Score* sebesar 90,51%. Capaian ini menunjukkan bahwa model mampu melakukan klasifikasi dengan baik pada data yang distribusi kelasnya tidak seimbang. Namun demikian, performa yang tinggi pada kondisi ini perlu diinterpretasikan secara hati-hati, karena dominasi kelas mayoritas berpotensi menyebabkan bias prediksi terhadap kelas minoritas.

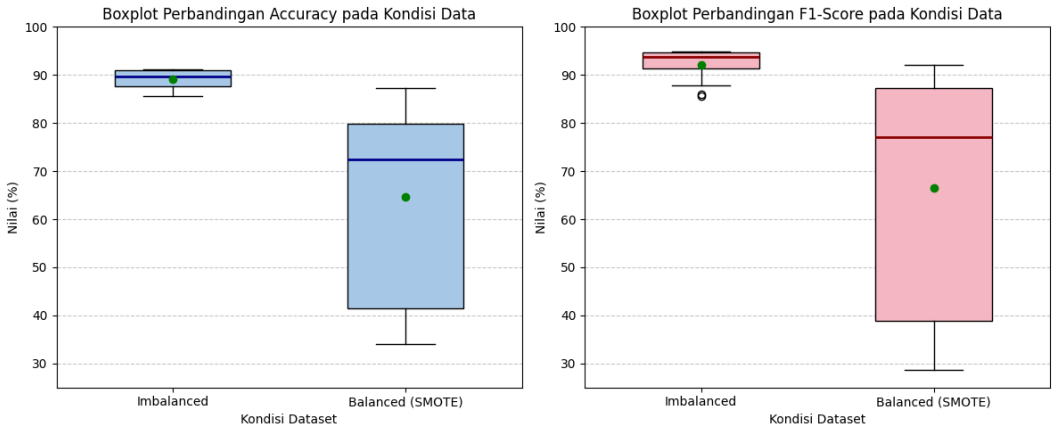
Tabel 4.31 Perbandingan Rata-rata Accuracy dan *F1-Score* pada Kondisi Data

No.	**Imbalanced**			**Balanced dengan SMOTE**		
	Skenario	**Accuracy**	***F1-Score***	**Skenario**	**Accuracy**	***F1-Score***
1	1	91.12%	94.80%	13	79.58%	85.56%
2	2	91.04%	94.68%	14	68.76%	70.42%
3	3	90.47%	94.15%	15	65.11%	67.80%
4	4	85.60%	85.60%	16	78.93%	87.39%
5	5	87.89%	87.89%	17	41.01%	40.04%
6	6	85.95%	85.95%	18	33.99%	28.73%
7	7	91.04%	94.83%	19	87.32%	92.07%
8	8	91.04%	94.71%	20	85.38%	88.95%
9	9	89.75%	93.65%	21	80.52%	87.12%
10	10	86.60%	92.53%	22	76.00%	83.66%
11	11	89.75%	94.01%	23	41.49%	35.46%
12	12	88.46%	93.11%	24	38.27%	31.76%
	Rata-Rata	**88.68%**	**90.51%**	**Rata-Rata**	**64.70%**	**66.58%**

Sebaliknya, pada kondisi data yang telah diseimbangkan menggunakan SMOTE, rata-rata performa model mengalami penurunan menjadi 64,70% untuk Accuracy dan 66,58% untuk *F1-Score*. Penurunan ini tidak terjadi secara merata pada seluruh skenario, melainkan dipengaruhi oleh perbedaan kapasitas arsitektur MLP yang digunakan. Beberapa skenario dengan arsitektur yang relatif lebar masih mampu mempertahankan performa yang cukup baik, sementara arsitektur dengan

jumlah neuron yang lebih sempit menunjukkan penurunan performa yang signifikan.

Distribusi performa model pada kedua kondisi dataset tersebut divisualisasikan melalui boxplot pada Gambar 4.26. Boxplot menunjukkan bahwa pada kondisi Imbalanced, sebaran nilai Accuracy dan *F1-Score* relatif rapat dengan median yang tinggi, menandakan stabilitas performa antar skenario. Sebaliknya, pada kondisi Balanced (SMOTE), terlihat sebaran nilai yang lebih lebar, khususnya pada metrik Accuracy, yang mengindikasikan meningkatnya variasi performa antar skenario pengujian. Temuan ini menunjukkan bahwa penerapan SMOTE meningkatkan tantangan pembelajaran model, terutama pada arsitektur yang memiliki kapasitas representasi terbatas.



Gambar 4.26 Boxplot Distribusi Accuracy dan *F1-Score* pada Kondisi Data

Berdasarkan hasil tersebut, dapat disimpulkan bahwa penerapan SMOTE memberikan dampak yang bersifat kontekstual. Teknik ini berpotensi mengurangi bias terhadap kelas mayoritas dan mendorong pembelajaran pola yang lebih seimbang, namun pada saat yang sama menuntut kapasitas model yang memadai untuk mengakomodasi kompleksitas data sintetis yang dihasilkan. Oleh karena itu,

efektivitas SMOTE dalam sistem deteksi konsumsi energi pada navigasi cerdas kendaraan listrik sangat bergantung pada kesesuaian antara metode penyeimbangan data dan arsitektur MLP yang digunakan.

4.4.3 Pengaruh *Threshold* Terhadap Trade-Off *Precision–Recall*

Parameter terakhir yang dievaluasi dalam penelitian ini adalah ambang batas keputusan (*threshold*). Secara teoritis, dalam klasifikasi biner, menaikkan nilai *threshold* akan membuat model lebih selektif dalam memprediksi kelas positif (Robles dkk., 2020). Hal ini biasanya meningkatkan *Precision* (mengurangi *False positive*) namun berisiko menurunkan *Recall* (menambah *False Negative*).

Tabel 4.32 Rekapitulasi Rata-rata Accuracy dan *F1-Score* pada Variasi *Threshold* Data Imbalanced

0.5			**0.6**			**0.7**		
Uji Coba	**Acc**	***F1-Score***	**Uji Coba**	**Acc**	***F1-Score***	**Uji Coba**	**Acc**	***F1-Score***
1	91.12%	94.80%	2	91.04%	94.68%	3	90.47%	94.15%
4	85.60%	85.60%	5	87.89%	92.77%	6	85.95%	91.45%
7	91.04%	94.83%	8	91.04%	94.71%	9	89.75%	93.65%
10	86.60%	86.60%	11	89.75%	94.01%	12	88.46%	93.11%
Rata-Rata	**88.59%**	**90.46%**	**Rata-Rata**	**89.93%**	**94.04%**	**Rata-Rata**	**88.66%**	**93.09%**

Untuk menganalisis pengaruh tersebut, dilakukan evaluasi kinerja model pada tiga nilai *threshold*, yaitu 0,50, 0,60, dan 0,70, baik pada kondisi dataset *Imbalanced* maupun dataset yang telah diseimbangkan menggunakan *Synthetic Minority Over-sampling Technique* (SMOTE). Rekapitulasi rata-rata nilai Accuracy dan *F1-Score* pada kondisi data *Imbalanced* disajikan pada Tabel 4.32, sedangkan hasil evaluasi pada kondisi data *Balanced (*SMOTE*)* disajikan pada Tabel 4.33.

Berdasarkan Tabel 4.32, terlihat bahwa pada kondisi data *Imbalanced*, performa model relatif stabil terhadap variasi nilai *threshold*. Rata-rata Accuracy berada pada rentang 88,59% hingga 89,93%, sementara *F1-Score* tetap tinggi pada seluruh nilai *threshold*. Kenaikan *threshold* dari 0,50 ke 0,60 bahkan diikuti oleh peningkatan nilai rata-rata *F1-Score*, sebelum sedikit menurun pada *threshold* 0,70. Pola ini menunjukkan bahwa pada data yang tidak seimbang, perubahan *threshold* tidak memberikan dampak yang signifikan terhadap performa agregat model, yang ditunjukkan oleh perubahan rata-rata *Accuracy* yang relatif kecil, yaitu kurang dari 1,5% (dari 88,59% pada *threshold* 0,50 menjadi 89,93% pada *threshold* 0,60 dan 88,66% pada *threshold* 0,70). Demikian pula, perubahan *F1-Score* berada di bawah 4%, sehingga secara statistik variasi *threshold* pada kondisi ini tidak mengubah kinerja model secara substansial.

Stabilitas performa tersebut mengindikasikan bahwa model cenderung menghasilkan probabilitas prediksi dengan tingkat keyakinan yang tinggi terhadap kelas mayoritas. Akibatnya, pengetatan *threshold* tidak secara langsung menurunkan sensitivitas deteksi, namun juga tidak mampu mengurangi bias prediksi terhadap kelas minoritas secara substansial.

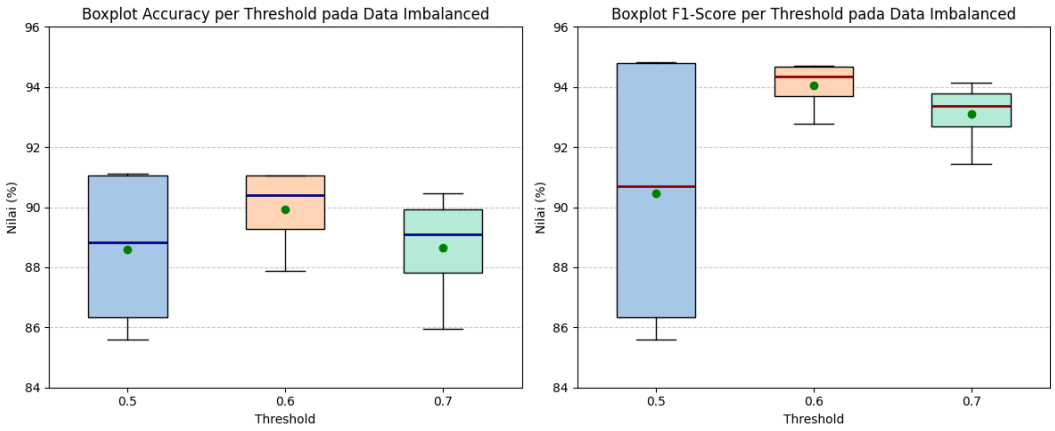
Tabel 4.33 Rekapitulasi Rata-rata Accuracy dan *F1-Score* pada Variasi *Threshold* Data Balanced (SMOTE)

0.5			**0.6**			**0.7**		
Uji Coba	**Acc**	***F1-Score***	**Uji Coba**	**Acc**	***F1-Score***	**Uji Coba**	**Acc**	***F1-Score***
13	79.58%	85.56%	14	68.76%	70.42%	15	65.11%	67.80%
16	78.93%	87.39%	17	41.01%	40.04%	18	33.99%	28.73%
19	87.32%	92.07%	20	85.38%	88.95%	21	80.52%	87.12%
22	76.00%	83.66%	23	41.49%	35.46%	24	38.27%	31.76%
Rata-Rata	**80.46%**	**87.17%**	**Rata-Rata**	**59.16%**	**58.72%**	**Rata-Rata**	**54.47%**	**53.85%**

Berbeda dengan kondisi sebelumnya, hasil pada Tabel 4.33 menunjukkan bahwa pada data yang telah diseimbangkan menggunakan SMOTE, peningkatan *threshold* berdampak signifikan terhadap penurunan performa model. Rata-rata Accuracy menurun dari 80,46% pada *threshold* 0,50 menjadi 59,16% pada *threshold* 0,60, dan kembali turun menjadi 54,47% pada *threshold* 0,70. Penurunan yang serupa juga terjadi pada *F1-Score*, yang mengindikasikan melemahnya keseimbangan antara ketepatan dan sensitivitas deteksi.

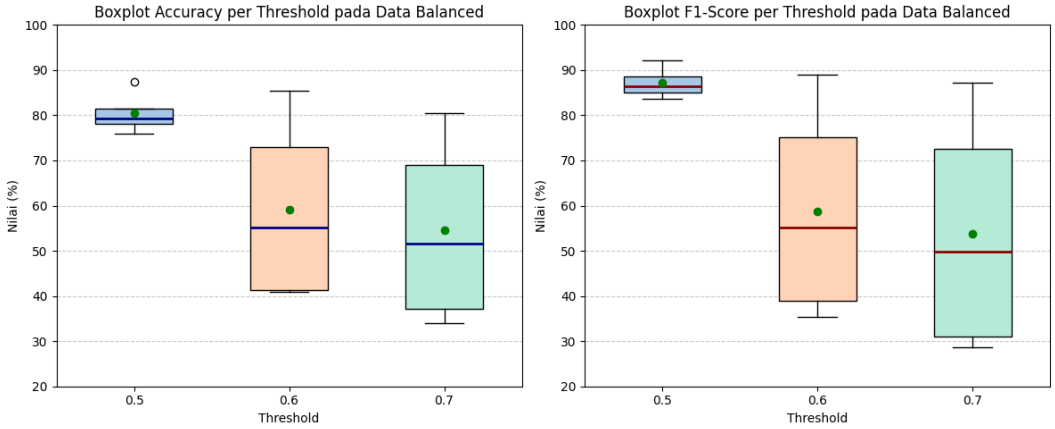
Penurunan performa ini tidak terjadi secara merata pada seluruh skenario, melainkan dipengaruhi oleh variasi kapasitas arsitektur MLP yang digunakan. Beberapa arsitektur dengan jumlah neuron yang lebih lebar masih mampu mempertahankan performa yang relatif baik, sementara arsitektur dengan kapasitas representasi terbatas mengalami penurunan performa yang drastis ketika *threshold* diperketat.

Distribusi performa model terhadap variasi *threshold* divisualisasikan lebih lanjut melalui boxplot pada Gambar 4.27 dan Gambar 4.28.



Gambar 4.27 Boxplot Distribusi Accuracy dan *F1-Score* terhadap Variasi *Threshold* pada Data Imbalanced

Berdasarkan Gambar 4.27, terlihat bahwa pada kondisi data *imbalanced*, distribusi nilai *Accuracy* dan *F1-Score* berada pada rentang yang relatif rapat dengan nilai median yang tetap tinggi di seluruh variasi *threshold*. Perubahan nilai *threshold* dari 0,50 hingga 0,70 tidak menyebabkan penurunan performa yang signifikan. Bahkan, pada beberapa skenario, terlihat adanya kecenderungan peningkatan pada nilai *F1-Score*. Fenomena ini mengindikasikan bahwa pada dataset yang tidak seimbang, model memiliki tingkat keyakinan prediksi yang sangat tinggi terhadap kelas mayoritas, sehingga variasi *threshold* navigasi tidak memberikan dampak yang besar terhadap hasil klasifikasi secara keseluruhan.



Gambar 4.28 Boxplot Distribusi Accuracy dan *F1-Score* terhadap Variasi *Threshold* pada Data Balanced (SMOTE)

Sebaliknya, pada data Balanced (SMOTE), peningkatan threshold dari 0,50 ke 0,60 dan terutama ke 0,70 menyebabkan penurunan performa dan stabilitas model yang cukup tajam, terutama pada arsitektur dengan kapasitas representasi yang terbatas. Penurunan ini ditandai dengan turunnya rata-rata Accuracy lebih dari 20% serta pelebaran sebaran nilai antar skenario, yang mengindikasikan melemahnya sensitivitas deteksi kelas positif.

Berdasarkan analisis komparatif terhadap variasi *threshold* pada kedua kondisi dataset tersebut, dapat disimpulkan bahwa pengaruh *threshold* terhadap *trade-off Precision–Recall* sangat bergantung pada keseimbangan data. Pada data *Imbalanced*, variasi *threshold* cenderung tidak memengaruhi performa model secara signifikan, namun juga tidak efektif dalam mengurangi bias terhadap kelas minoritas. Sebaliknya, pada data *Balanced* (SMOTE), peningkatan *threshold* menyebabkan penurunan performa dan stabilitas model, terutama pada arsitektur dengan kapasitas representasi yang terbatas. Oleh karena itu, dalam konteks penelitian ini, penggunaan *threshold* 0,50 dipandang sebagai konfigurasi yang paling representatif untuk menjaga keseimbangan antara stabilitas kinerja dan sensitivitas deteksi pada data yang telah diseimbangkan.

Dalam perspektif Islam, kemampuan untuk memprediksi dan membaca pola merupakan bagian dari pembacaan terhadap tanda-tanda kebesaran Allah SWT. Hal ini selaras dengan firman-Nya dalam Q.S. Yasin ayat 39, di mana Allah menjelaskan tentang keteraturan peredaran bulan.

وَالْقَمَرَ قَدَّرْنٰهُ مَنَازِلَ حَتّٰى عَادَ كَالْعُرْجُوْنِ الْقَدِيْمِ

"(Begitu juga) bulan, Kami tetapkan bagi(-nya) tempat-tempat peredaran sehingga (setelah ia sampai ke tempat peredaran yang terakhir,) kembalilah ia seperti bentuk tandan yang tua." (Q.S. Yasin [22]:39)

Berdasarkan tafsir jalalain, Allah SWT menetapkan dua puluh delapan manzilah sebagai jalur peredaran bulan yang berlangsung secara konsisten dalam durasi waktu tertentu. Proses transisi fase bulan yang berawal dari penampakan tipis hingga kembali mengecil menyerupai tandan tua yang melengkung dan berwarna kuning merupakan bukti adanya siklus sistematis yang terukur dan dapat diprediksi.

Fenomena keteraturan ini memberikan isyarat filosofis bahwa setiap pola di alam semesta memiliki karakteristik unik yang dapat diidentifikasi melalui pengamatan yang mendalam.

Prinsip keteraturan pola tersebut diadopsi dalam arsitektur Multi-Layer Perceptron (MLP) untuk mendeteksi karakteristik penggunaan energi. Melalui proses pembelajaran yang iteratif, model MLP memetakan transisi data dari satu kondisi ke kondisi lainnya sebagaimana perpindahan bulan antar manzilah. Melalui pemetaan pola variasi energi pada kondisi penggunaan yang optimal serta penggunaan yang signifikan, sistem memiliki kemampuan untuk memberikan hasil klasifikasi yang tepat mengenai estimasi kebutuhan daya pada tahap navigasi selanjutnya.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan, dapat disimpulkan bahwa model Multilayer Perceptron (MLP) mampu melakukan deteksi konsumsi energi pada navigasi cerdas kendaraan listrik dengan performa yang bervariasi, yang dipengaruhi secara signifikan oleh konfigurasi arsitektur jaringan, kondisi keseimbangan data, serta nilai ambang batas keputusan (*threshold*) yang digunakan. Evaluasi menggunakan metrik Accuracy, *Precision*, *Recall*, dan *F1-Score* menunjukkan bahwa pemilihan arsitektur MLP berperan penting dalam menentukan kapasitas model dalam mempelajari pola data, terutama ketika dataset mengalami perubahan distribusi akibat penerapan teknik penyeimbangan data.

Pada kondisi data Imbalanced, seluruh variasi arsitektur MLP cenderung menghasilkan nilai Accuracy dan *F1-Score* yang tinggi dan stabil. Namun, performa tersebut perlu diinterpretasikan secara hati-hati karena adanya kecenderungan bias terhadap kelas mayoritas, yang tercermin dari tingginya nilai *Recall* namun berpotensi mengabaikan sensitivitas terhadap kelas minoritas. Sebaliknya, pada kondisi data Balanced menggunakan SMOTE, performa model menjadi lebih bervariasi dan sangat bergantung pada kapasitas arsitektur yang digunakan. Arsitektur dengan jumlah neuron yang lebih lebar menunjukkan stabilitas dan kemampuan generalisasi yang lebih baik dibandingkan arsitektur dengan struktur yang sempit.

Selain itu, hasil analisis menunjukkan bahwa nilai *threshold* memengaruhi trade-off antara *Precision* dan *Recall*, terutama pada kondisi data Balanced. Peningkatan *threshold* cenderung menurunkan sensitivitas deteksi kelas positif, yang berdampak pada penurunan *Recall* dan *F1-Score*, meskipun *Precision* relatif meningkat. Pada data Imbalanced, perubahan *threshold* tidak memberikan dampak yang signifikan terhadap performa model secara keseluruhan, yang mengindikasikan adanya dominasi kelas mayoritas dalam proses prediksi.

Dengan demikian, tujuan penelitian untuk mengukur performa MLP berdasarkan metrik Accuracy, *Precision*, *Recall*, dan *F1-Score* telah tercapai, serta mampu memberikan pemahaman yang komprehensif mengenai pengaruh arsitektur MLP, kondisi data, dan *threshold* terhadap kinerja model dalam mendeteksi konsumsi energi. Hasil penelitian ini menunjukkan bahwa pemilihan konfigurasi model yang tepat menjadi faktor krusial untuk memperoleh keseimbangan antara *Accuracy*, stabilitas, dan sensitivitas deteksi, khususnya pada sistem navigasi cerdas kendaraan listrik.

5.2 Saran

Berdasarkan hasil penelitian dan kesimpulan yang telah diuraikan, berikut adalah beberapa saran untuk pengembangan penelitian selanjutnya:

1. **Perluasan dan Penyeimbangan Dataset** Keterbatasan utama penelitian ini terletak pada jumlah data yang relatif kecil dan distribusi kelas yang tidak seimbang (*imbalanced*). Penelitian selanjutnya sangat disarankan untuk memperluas akuisisi data, baik melalui pengujian lapangan yang lebih intensif maupun penggabungan dengan dataset eksternal yang relevan. Penambahan

jumlah data secara signifikan akan memungkinkan model untuk belajar dari variasi kasus yang lebih kaya tanpa terlalu bergantung pada data sintetis. Selain itu, upaya pengumpulan data sebaiknya difokuskan pada penambahan sampel kelas minoritas secara organik agar keseimbangan data dapat tercapai secara alami.

2. **Eksplorasi Metode yang Lebih Optimal** Meskipun Multi-Layer Perceptron (MLP) memberikan hasil yang baik, penelitian di masa depan perlu melakukan studi komparatif dengan algoritma *machine learning* lainnya. Mengingat karakteristik data yang berskala kecil hingga menengah, algoritma ensemble seperti Random Forest, XGBoost, atau Support Vector Machine (SVM) memiliki potensi untuk menghasilkan performa yang lebih efisien dan stabil. Selain itu, penerapan teknik optimasi hyperparameter otomatis (seperti Grid Search atau Bayesian Optimization) disarankan untuk menemukan kombinasi parameter pelatihan (*learning rate*, *momentum*, *batch size*) yang paling presisi.
3. **Pencarian Arsitektur Jaringan yang Lebih Unggul** Penelitian lanjutan direkomendasikan untuk melampaui batasan eksperimen saat ini yang hanya berfokus pada variasi neuron dan *layer* sederhana. Investigasi dapat diarahkan pada pengujian arsitektur tingkat lanjut seperti *Deep Neural Networks* dan LSTM, serta optimalisasi teknik *Dropout* untuk mencegah *overfitting* pada model yang lebih besar. Tujuannya adalah menemukan konfigurasi topologi yang superior dalam mendeteksi karakteristik kompleks dari data konsumsi energi.

DAFTAR PUSTAKA

Adamu-Fika, F., Ayeku, D. I., Ramalan, A. T., Mafua, H. O., Baba-Onoja, A. E., Samson, T. J., & Okpoko, O. V. (2023). Application of Synthetic Minority Over-Sampling Technique to Mitigate Class Imbalance in Stroke Prediction. *Advances in Multidisciplinary & Scientific Research Journal Publication*, *37*, 195–210. https://doi.org/10.22624/AIMS/ACCRACROSSBORDER2023V2P2P40x

Alateef, S., & Thomas, N. (2023). *Energy consumption estimation for electric vehicles usning routing API data*.

Ausri, F., & Bigazzi, A. (2024). Mesoscopic model of cycling trip energy expenditure based on operating modes. *Journal of Cycling and Micromobility Research*, *2*, 100030. https://doi.org/10.1016/j.jcmr.2024.100030

Baroughi, R. M., Attar, H., & Rastegari, F. (2023). A Comprehensive Simulation of Electric Vehicle Energy Consumption: Incorporating Route Planning and Machine Learning-Based Predictions. *2023 2nd International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEAI)*, 1–4. https://doi.org/10.1109/EICEEAI60672.2023.10590474

Călin, S. (2025). Handling Imbalanced Data: The SMOTE Technique. *2025 17th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 1–5. https://doi.org/10.1109/ECAI65401.2025.11095450

Chen, J., Li, B., Zhang, Z., Liu, M., Piao, C., & Huang, M. (2024). *Energy Consumption Estimation of Electric Vehicles for Future Driving Process Based on Driving Conditions Recognition*.

Da Silva, I. N., Hernane Spatti, D., Andrade Flauzino, R., Liboni, L. H. B., & Dos Reis Alves, S. F. (2017). Multilayer Perceptron Networks. Dalam I. N. Da Silva, D. Hernane Spatti, R. Andrade Flauzino, L. H. B. Liboni, & S. F. Dos Reis Alves, *Artificial Neural Networks* (hlm. 55–115). Springer International Publishing. https://doi.org/10.1007/978-3-319-43162-8_5

Dania, N. N. (2024). Narrative Review: Perancangan Sistem Pengisian Cepat untuk Kendaraan Listrik dengan Algoritma Kontrol Baterai. *Jupiter: Publikasi Ilmu Keteknikan Industri, Teknik Elektro dan Informatika*, *2*(6), 152–157. https://doi.org/10.61132/jupiter.v2i6.629

Diukarev, V., & Starukhin, Y. (2024). Proposed Methods for Preventing Overfitting in Machine Learning and Deep Learning. *Asian Journal of Research in Computer Science*, *17*(10), 85–94. https://doi.org/10.9734/ajrcos/2024/v17i10511

Duarte, A., & Yoshizaki, H. T. Y. (2025, Juli 2). Comparative Analysis of Machine Learning Models for Predicting Energy Consumption on Last-Mile BEV Routes. *Proceedings of the International Conference on Industrial Engineering and Operations Management*. 8th European International Conference on Industrial Engineering and Operations Management, IESEG School of Management, Paris la Defense, Prom. de l'Arche, 92800 Puteaux, France. https://doi.org/10.46254/EU08.20250198

Faye, B., Dilmi, M.-D., Azzag, H., Lebbah, M., & Bouchaffra, D. (2023). *Context Normalization Layer with Applications*. https://doi.org/10.48550/ARXIV.2303.07651

Gulo, S. H., & Lubis, A. H. (2024). Penerapan Multi-Layer Perceptron untuk Mengklasifikasi Penduduk Kurang Mampu. *Explorer*, *4*(2), 51–59. https://doi.org/10.47065/explorer.v4i2.1146

Hariyadi, M., Nur Fadila, J., & Aziz, O. Q. (2025). *Pengembangan Sistem Navigasi Dan Rute pada Smart Electric Vehicle*. https://repository.uin-malang.ac.id/26184/

Ikwen, O. B., E Eteng, I., & U Ogban, F. (2024). Predictive Crime Analysis Using Multi-Layer Perceptron Architecture. *Global Journal of Pure and Applied Sciences*, *30*(2), 203–219. https://doi.org/10.4314/gjpas.v30i2.10

Jacot, A., Golikov, E., Hongler, C., & Gabriel, F. (2022). *Feature Learning in L_{2}-regularized DNNs: Attraction/Repulsion and Sparsity* (Versi 2). arXiv. https://doi.org/10.48550/ARXIV.2205.15809

Jiang, B.-H., Hsu, C.-C., Su, N.-W., & Lin, C.-C. (2024). A Review of Modern Electric Vehicle Innovations for Energy Transition. *Energies*, *17*(12), 2906. https://doi.org/10.3390/en17122906

Lim, H., Lee, J. W., Boyack, J., & Choi, J. B. (2024). *EV-PINN: A Physics-Informed Neural Network for Predicting Electric Vehicle Dynamics*.

Markov, K. (2023). Multilayer Perceptron with Backpropagation, HDL Coder, and FPGA Technology: An Integrated Approach for Efficient Neural Network Implementation. *Problems of Engineering Cybernetics and Robotics*, *80*. https://doi.org/10.7546/PECR.80.23.02

Modi, S., & Bhattacharya, J. (2022). A system for electric vehicle's energy-aware routing in a transportation network through real-time prediction of energy

consumption. *Complex & Intelligent Systems*, *8*(6), 4727–4751. https://doi.org/10.1007/s40747-022-00727-4

Moulik, B., Kaur, S., & Ijaz, M. (2025). Optimized Energy Consumption of Electric Vehicles with Driving Pattern Recognition for Real Driving Scenarios. *Algorithms*, *18*(4), 204. https://doi.org/10.3390/a18040204

Owusu-Adjei, M., Ben Hayfron-Acquah, J., Frimpong, T., & Abdul-Salaam, G. (2023). Imbalanced class distribution and performance evaluation metrics: A systematic review of prediction accuracy for determining model performance in healthcare systems. *PLOS Digital Health*, *2*(11), e0000290. https://doi.org/10.1371/journal.pdig.0000290

Padmavathy, R., K., J. P., T., G., & K., D. (2023). A Machine Learning-Based Energy Optimization System for Electric Vehicles. *E3S Web of Conferences*, *387*, 04008. https://doi.org/10.1051/e3sconf/202338704008

Pauli, S. T. Z. D., Kleina, M., & Bonat, W. H. (2021). MULTILAYER PERCEPTRON ARTIFICIAL NEURAL NETWORKS: AN APPROACH FOR LEARNING THROUGH THE BAYESIAN FRAMEWORK. *REVISTA BRASILEIRA DE BIOMETRIA*, *39*(1), 45–59. https://doi.org/10.28951/rbb.v39i1.495

Qamar, R., & Zardari, B. A. (2023). Artificial Neural Networks: An Overview. *Mesopotamian Journal of Computer Science*, *2023*, 124–133. https://doi.org/10.58496/MJCSC/2023/015

Ranjan, A., Giribabu, D., & Kakodia, S. K. (2022). Energy Management System and Electric Motors of Electric Vehicles: A Review. *2022 IEEE International Conference on Current Development in Engineering and Technology (CCET)*, 1–6. https://doi.org/10.1109/CCET56606.2022.10080645

Riyanto, S., Sitanggang, I. S., Djatna, T., & Atikah, T. D. (2023). Comparative Analysis using Various Performance Metrics in Imbalanced Data for Multi-class Text Classification. *International Journal of Advanced Computer Science and Applications*, *14*(6). https://doi.org/10.14569/IJACSA.2023.01406116

Robles, E., Zaidouni, F., Mavromoustaki, A., & Refael, P. (2020). *Threshold Optimization in Multiple Binary Classifiers for Extreme Rare Events using Predicted Positive Data.*

Surah Tafsir Al-Hujurat:13. Quran.com. Diambil 28 Desember 2025, dari https://quran.com/id/kamar-kamar/13/tafsirs

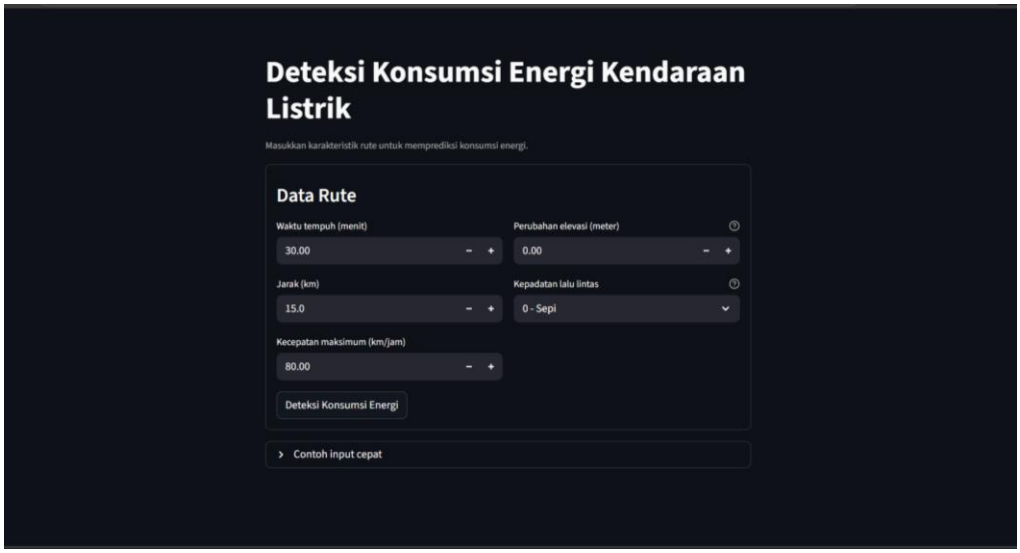
Surat Yasin Ayat 39: Arab, Latin, Terjemah dan Tafsir Lengkap | Quran NU Online. Diambil 14 Desember 2025, dari https://quran.nu.or.id/yasin/39

Zhang, J., Liu, W., Wang, Z., & Fan, R. (2024). Electric Vehicle Power Consumption Modelling Method Based on Improved Ant Colony Optimization-Support Vector Regression. *Energies*, *17*(17), 4339. https://doi.org/10.3390/en17174339

LAMPIRAN

Lampiran I Interface Deteksi Konsumsi Energi

Tampilan awal dari model Deteksi Konsumsi Energi.



Tampilan ketika *user* menginput data rute dan sistem memberikan *output* Deteksi Konsumsi Energi.