KLASIFIKASI KERUSAKAN MESIN SEPEDA MOTOR MENGGUNAKAN METODE *NEURAL NETWORK BACKPROPAGATION*

SKRIPSI

Oleh :
MUHAMMAD HARIS IBRAHIM
NIM. 200605110097

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2024

KLASIFIKASI KERUSAKAN MESIN SEPEDA MOTOR MENGGUNAKAN METODE *NEURAL NETWORK BACKPROPAGATION*

SKRIPSI

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

Oleh :
MUHAMMAD HARIS IBRAHIM
NIM. 200605110097

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2024

HALAMAN PERSETUJUAN

KLASIFIKASI KERUSAKAN MESIN SEPEDA MOTOR MENGGUNAKAN METODE *NEURAL NETWORK BACKPROPAGATION*

SKRIPSI

Oleh :
MUHAMMAD HARIS IBRAHIM
NIM. 200605110097

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 20 November 2024

Pembimbing I,	Pembimbing II,
Dr. Agung Teguh Wibowo Almais, M.T	Dr. M. Ainul Yaqin, M.Kom
NIP. 1986031 202321 1 016	NIP. 19761013 200604 1 004

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

Dr. Ir. Fachrul Kurniawan, M.MT, IPU
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

KLASIFIKASI KERUSAKAN MESIN SEPEDA MOTOR MENGGUNAKAN METODE *NEURAL NETWORK BACKPROPAGATION*

SKRIPSI

Oleh :
MUHAMMAD HARIS IBRAHIM
NIM. 200605110097

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 03 Desember 2024

Susunan Dewan Penguji

Ketua Penguji	: Syahiduz Zaman, M.Kom NIP. 19700502 200501 1 005	( )
Anggota Penguji I	: Nurizal Dwi Priandani, M.Kom NIP. 19920830 202203 1 001	( )
Anggota Penguji II	: Dr. Agung Teguh Wibowo Almais, M.T NIP. 1986031 202321 1 016	( )
Anggota Penguji III	: Dr. M. Ainul Yaqin, M.Kom NIP. 19761013 200604 1 004	( )

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

Dr. Ir. Fachrul Kurniawan, M.MT, IPU
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama	: Muhammad Haris Ibrahim
NIM	: 200605110097
Fakultas / Program Studi	: Sains dan Teknologi / Teknik Informatika
Judul Skripsi	: Klasifikasi Kerusakan Mesin Sepeda Motor Menggunakan *Metode Neural Network Backpropagation*

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, 03 Desember 2024
Yang membuat pernyataan,

Muhammad Haris Ibrahim
NIM. 200605110097

MOTTO

Fortis Fortuna Adiuvat

"Keberuntungan berpihak pada yang berani"

(John Wick)

HALAMAN PERSEMBAHAN

Segala puji bagi Allah SWT yang telah memberikan limpahan rahmat dan hidayah kepada hamba-Nya, sehingga saya bisa menyelesaikan karya ini. Serta shalawat dan salam semoga selalu tercurahkan kepada nabi Muhammad SAW.

Saya persembahkan karya ini kepada:

Ayah dan ibu saya,

Beny Ibrahim dan Arifah Ulyani

Terima kasih atas segala ilmu, dukungan dan semangatnya serta yang tidak kalah penting adalah doa di setiap pagi dan malamnya, sehingga saya bisa menyelesaikan ujian demi ujian khusunya skripsi ini. semoga karya ini menjadi karya yang bermanfaat dunia dan akhirat serta menjadi keberkahan untuk semuanya.

Teruntuk keluarga besar saya,

Terima kasih atas semangat, dukungan, kesabaran, dan doa serta selalu ada baik saat senang maupun duka

Tim Penelitian serta pihak yang membantu penelitian

Terima kasih atas ilmu, pengalaman, dan kesempatannya serta dukungan yang selalu diberikan. Sehingga saya bisa meneyelesaikan karya ini.

Teman-teman terdekat saya,

Grup belajar, Grup BPH depan rektorat, dan teman-teman seperjuangan,

Seruluh Mahasiswa Teknik Informatika khususnya untuk Angkatan 2020 yang tidak bisa saya sebutkan satu-persatu

Yang telah berbagi ilmu, support, serta pengalaman yang berharga selama

perkuliahan ini. Semoga kita semua selalu diberi kesehatan dan kemudahan untuk

menuntut ilmu serta dimudahkan dalam urusan lainnya oleh Allah SWT

KATA PENGANTAR

Alhamdulillah, segala puji hanya milik Allah Subhanahu Wa Ta'ala atas segala nikmat dan kasih sayang-Nya yang telah memudahkan penulis untuk menyelesaikan skripsi yang berjudul "Klasifikasi Kerusakan Mesin Sepeda Motor Menggunakan Metode *Neural Network Backpropagation*". Semoga shalawat dan salam senantiasa terlimpah kepada Nabi Muhammad Sallalahu 'Alaihi wa Sallam. Dan semoga kita semua mendapat syafaatnya di hari kiamat nanti, Aamiin.

Penulis mengucapkan rasa syukur dan terima kasih yang tak terhingga kepada semua pihak-pihak yang selalu memberikan bantuan dan motivasi kepada penulis untuk dapat menyelesaikan skripsi ini. Ucapan ini penulis sampaikan kepada:

1. Prof. Dr. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Hj. Sri Harini, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Ir. Fachrul Kurniawan, M.MT, IPU., selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Dr. Agung Teguh Wibowo Almais, M.T, selaku pembimbing utama yang dengan penuh kesabaran dan ketulusan hati memberikan bimbingan, arahan, serta dorongan dalam setiap tahap penyusunan skripsi ini.
5. Dr. M. Ainul Yaqin, M.Kom., selaku dosen pembimbing kedua penulis yang memberikan masukan terkait integrasi ayat pada skripsi ini.

6. Syahiduz Zaman, M.Kom., selaku penguji utama dan Nurizal Dwi Priandani, M.Kom., selaku penguji kedua yang telah berkenan menguji serta memberikan masukan yang membangun sehingga skripsi ini dapat terselesaikan dengan baik.
7. Nia Faricha S, Si., selaku admin Program Studi Teknik Informatika yang selalu sabar memberikan informasi, membantu, dan memberikan arahan selama perkuliahan dan proses penulisan skripsi ini.
8. Segenap dosen, laboran, dan jajaran staff Program Studi Teknik Informatika yang telah memberikan ilmu, pengetahuan, dan dukungan selama penulis menjalani studi hingga selesainya skripsi ini.
9. Kedua orang tua tercinta, Ibu Arifah Ulyani dan Ayah Beny Ibrahim yang selalu menjadi sumber kekuatan bagi penulis. Terima kasih karena selalu mengusahakan yang terbaik. Semoga Allah senantiasa memberikan kesehatan dan lindungan, sehingga dapat selalu berada disetiap perjalanan dan pencapaian penulis.
10. Sahabat seperjuangan "Grup Belajar" yang sudah setia menemani, memberikan semangat serta segala bantuan dari awal perkuliahan hingga saat ini.
11. Teman dekat penulis, Abdurrozzaaq, Fahreza, Galan, Onny, Revaldi, Rivaldi, Izzul, dan Yoga, yang sudah membersamai dan membantu penulis hingga saat ini. Terimakasih sudah menjadi keluarga sehingga penulis tidak pernah merasa kespian selama jauh dari rumah. Doa terbaik untuk

kalian, jangan sampai putus kekeluargaan dan pertemanan kita. Masih banyak hal-hal "*random*" yang belum kita lakukan bersama.

12. Teman-teman penulis, Rahma, Salma Ainur, Salma Nabila, Alif, dan Fahrendra yang sudah memberikan banyak bantuan serta masukan dalam proses penelitian ini.
13. Pemilik NIM 200401110209 sebagai *support system* yang selalu menemani, menjadi tempat berkeluh kesah, memberikan *supply* berupa cemilan, dan memberikan dorongan agar segera menyelesaikan skripsi ini.
14. Seluruh warga Teknik Informatika khususnya angkatan 2020 "Integer" yang telah memberikan kehangatan, motivasi, dan dukungan kepada penulis.
15. Seluruh pihak yang telah terlibat, baik secara langsung maupun tidak langsung dari awal perkuliahan hingga akhir penulisan skripsi ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih jauh dari kata sempurna. Maka dari itu penulis menerima saran, kritik dan masukan yang bersifat membangun sehingga skripsi ini dapat lebih dikembangkan. Penulis berharap semoga skripsi ini dapat memberikan manfaat untuk kedepannya.

Malang, 15 Desember 2024

Penulis

DAFTAR ISI

HALAMAN JUDUL ..... **ii**
HALAMAN PERSETUJUAN ..... **iii**
HALAMAN PENGESAHAN ..... **v**
PERNYATAAN KEASLIAN TULISAN ..... **v**
MOTTO ..... **vi**
HALAMAN PERSEMBAHAN ..... **vii**
KATA PENGANTAR ..... **viii**
DAFTAR ISI ..... **xi**
DAFTAR GAMBAR ..... **xiii**
DAFTAR TABEL ..... **xiv**
ABSTRAK ..... **xv**
ABSTRACT ..... **xvi**
البحث مستخلص ..... **xvii**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Rumusan Masalah 4
1.3 Batasan Masalah 4
1.4 Tujuan Penelitian 4
1.5 Manfaat Penelitian 4
BAB II STUDI PUSTAKA **6**
2.1 Penelitian Terkait 6
2.2 Sistem Prediksi 8
2.3 Mesin Sepeda Motor 9
2.4 *Neural Network Backpropagation* 10
2.4.1 Pengertian *Neural Network* 10
2.4.2 Pengertian *Backpropagation* 11
2.4.3 Langkah-Langkah Algoritma *Backpropagation* 12
BAB III METODE PENELITIAN **16**
3.1 Desain Penelitian 16
3.1.1 *Input* Data 16
3.1.2 *Preprocessing* 18
3.1.3 Implementasi *Neural Network* Backpropagation 23
3.2 Arsitektur Jaringan 24
3.3 Desain Eksperimen 28
3.4 Evaluasi 29
BAB IV HASIL DAN PEMBAHASAN **31**
4.1 Pengujian Model 31
4.1.1 Pengujian Model A 31
4.1.2 Pengujian Model B 34
4.1.3 Pengujian Model C 37
4.1.4 Pengujian Model D 40
4.1.5 Pengujian Model E 43
4.2 Visualisasi hasil uji coba 46
4.2.1 Pengaruh *Hidden layer* 47
4.2.2 Pengaruh *Epoch* 49
4.2.3 Pengaruh *Learning Rate* 50

BAB V KESIMPULAN DAN SARAN 54
5.1 Kesimpulan 54
5.2 Saran..... 55
DAFTAR PUSTAKA

DAFTAR GAMBAR

Gambar 2. 1 Arsitektur Neural Network Backpropagation 12
Gambar 3. 1 Desain Penelitian..... 16
Gambar 3. 2 Label Encoding 20
Gambar 3. 3 Distribusi kelas sebelum balancing..... 21
Gambar 3. 4 Source code untuk balancing data..... 21
Gambar 3. 5 Distribusi kelas setelah balancing 22
Gambar 3. 6 Source code untuk memisahkan kolom diagnosa 23
Gambar 3. 7 Source Code Data Split dengan K-Fold Cross Validation 23
Gambar 3. 8 Model A 25
Gambar 3. 9 Model B..... 26
Gambar 3. 10 Model C..... 26
Gambar 3. 11 Model D 27
Gambar 3. 12 Model E..... 28
Gambar 4. 1 Combined Confusion Matrix Model A1 32
Gambar 4. 2 Combined Confusion Matrix Model A2 33
Gambar 4. 3 Combined Confusion Matrix Model A3 34
Gambar 4. 4 Combined Confusion Matrix Model B1..... 35
Gambar 4. 5 Combined Confusion Matrix Model B2..... 36
Gambar 4. 6 Combined Confusion Matrix Model B3..... 37
Gambar 4. 7 Combined Confusion Matrix Model C1..... 38
Gambar 4. 8 Combined Confusion Matrix Model C2..... 39
Gambar 4. 9 Combined Confusion Matrix Model C3..... 40
Gambar 4. 10 Combined Confusion Matrix Model D1 41
Gambar 4. 11 Combined Confusion Matrix Model D2 42
Gambar 4. 12 Combined Confusion Matrix Model D3 43
Gambar 4. 13 Combined Confusion Matrix Model E1..... 44
Gambar 4. 14 Combined Confusion Matrix Model E2..... 45
Gambar 4. 15 Combined Confusion Matrix Model E3..... 46
Gambar 4. 16 Visualisasi Perbandingan Hidden layer..... 48
Gambar 4. 17 Perbandingan Pengaruh Epoch 49
Gambar 4. 18 Grafik Perbandingan Learning Rate 0.001-0.009 51

DAFTAR TABEL

Tabel 2. 1 Penelitian Terkait 6
Tabel 3. 1 Data Servis di AHASS Putra Jaya Tunggulwulung 2023 17
Tabel 3. 2 Data Kriteria Keluhan 17
Tabel 3. 3 Data Alternatif Kerusakan 18
Tabel 3. 4 Labeling Variabel 18
Tabel 3. 5 Data sebelum dilakukan balancing 20
Tabel 3. 6 Nilai Hyperparameter 29
Tabel 4. 1 Skenario Pengujian Model A 31
Tabel 4. 2 Skenario Pengujian Model B 35
Tabel 4. 3 Skenario Pengujian Model C 38
Tabel 4. 4 Skenario Pengujian Model D 41
Tabel 4. 5 Skenario Pengujian Model E 44
Tabel 4. 6 Visualisasi Hasil Uji Coba 47
Tabel 4. 7 Visualisasi Pengaruh Hidden layer Terhadap Akurasi 48
Tabel 4. 8 Perbandingan Pengaruh Epoch 49
Tabel 4. 9 Pengujian Learning Rate pada Model E dengan Epoch 2000 50

ABSTRAK

Ibrahim, Muhammad Haris. 2024. **Klasifikasi Kerusakan Mesin Sepeda Motor Menggunakan Metode *Neural Network Backpropagation***. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Agung Teguh Wibowo Almais, M.T. (II) Dr. M. Ainul Yaqin, M.Kom.

Kata kunci: Kerusakan Mesin, *Neural Network, Backpropagation,* Klasifikasi

Sepeda motor merupakan salah satu jenis kendaraan yang banyak digunakan oleh masyarakat sebagai alat transportasi.Mesin motor yang bekerja dalam kondisi berat seringkali rentan terhadap berbagai masalah dan kerusakan sehingga perlu dilakukan servis secara rutin. Bengkel AHASS Putra Jaya Tunggulwulung mengumpulkan data untuk menentukan diagnosa kerusakan. Beberapa data yang dikumpulkan adalah tipe sepeda motor, tahun pembelian sepeda motor, jumlah kilometer, dan keluhan yang dialami oleh pengguna sepeda motor, digunakan dalam penelitian ini untuk menentukan diagnosa kerusakan. Penelitian ini bertujuan untuk memprediksi kerusakan mesin sepeda motor. Metode yang digunakan adalah *Neural Network* Backpropagation. Penelitian ini menggunakan 333 data yang kemudian dilakukan balancing data menggunakan metode RandomOverSampler. Kemudian data split menggunakan metode 10-Fold Cross Validation. Epoch yang digunakan dalam penelitian ini yaitu sebesar 500, 1000, dan 2000, dengan learning rate sebesar 0.001. Dalam penelitian ini, juga terdapat 5 jenis model dengan jumlah *hidden layer* yang berbeda. Untuk mengevaluasi performa model, pengukuran dilakukan menggunakan confusion matrix. Hasil analisis menunjukkan bahwa pengaruh *hidden layer*, epoch, dan learning rate terhadap akurasi model *Neural Network* saling berhubungan. Penambahan *hidden layer* meningkatkan kemampuan model menangkap pola data kompleks. Secara keseluruhan, *hidden layer* berjumlah 10 menghasilkan akurasi yang lebih optimal. namun *hidden layer* sebanyak 12 memberikan hasil akurasi terbaik di 81.61%. Jumlah 1000 dan 2000 memungkinkan model mencapai akurasi optimal dengan peningkatan performa signifikan dari epoch 500. Selain itu, *learning rate* sebesar 0.001 menghasilkan *accuracy* yang paling tinggi dibandingkan nilai *learning rate* lain yang lebih besar. Kombinasi optimal dari parameter ini menunjukkan bahwa *tuning* hyperparameter yang tepat sangat penting dalam meningkatkan performa model.

ABSTRACT

Ibrahim, Muhammad Haris. 2024. **Classification of Motorcycle Engine Damage Using Neural Network Backpropagation Method**. Thesis. Informatics Engineering Study Program, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University Malang. Advisor: (I) Dr. Agung Teguh Wibowo Almais, M.T. (II) Dr. M. Ainul Yaqin, M.Kom.

Motorcycles are one type of vehicle that is widely used by the public as a means of transportation. Motorcycle engines that work in heavy conditions are often prone to various problems and damage so they need to be serviced regularly. AHASS Putra Jaya Tunggulwulung workshop collects data to determine the diagnosis of damage. Some of the data collected are the type of motorcycle, year of purchase of the motorcycle, number of kilometers, and complaints experienced by motorcycle users, used in this study to determine the diagnosis of damage. This research aims to predict motorcycle engine damage. The method used is Neural Network Backpropagation. This research uses 333 data which is then carried out data balancing using the RandomOverSampler method. Then the data is split using the 10-Fold Cross Validation method. The epochs used in this study are 500, 1000, and 2000, with a learning rate of 0.001. In this study, there are also 5 types of models with different numbers of hidden layers. To evaluate the performance of the model, measurements are made using the confusion matrix. The analysis shows that the influence of hidden layer, epoch, and learning rate on the accuracy of neural network models is interconnected. The addition of hidden layers improves the model's ability to capture complex data patterns. Overall, the model with 12 hidden layers gave the best accuracy result at 81.61%. The number of 1000 and 2000 enabled the model to achieve optimal accuracy with significant performance improvement from epoch 500. In addition, the learning rate of 0.001 resulted in the highest accuracy compared to other larger learning rate values. The optimal combination of these parameters shows that proper hyperparameter *tuning* is crucial in improving the model's performance.

Keywords: Motorcycle Engine Damage, Neural Network, Backpropagation, Classification

البحث مستخلص

إبراهيم، محمد حارس. 2024. **تصنيف تلف محرك الدراجة النارية باستخدام طريقة الانتشار الخلفي للشبكة العصبية**. الأطروحة. برنامج دراسة هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المستشار: (الأول) د. أغونغ تيغوه ويبوو أليس، ماجستير في الهندسة المعلوماتية (الثاني) د. م. عين اليقين، ماجستير في العلوم والتكنولوجيا.

الكلمات المفتاحية : تلف محرك الدراجة النارية، الشبكة العصبية، التكاثر العكسي، التصنيف بيرسون

تُعد الدراجات النارية أحد أنواع المركبات التي يستخدمها الجمهور على نطاق واسع كوسيلة نقل. وغالباً ما تكون محركات الدراجات النارية التي تعمل في ظروف قاسية عرضة لمشاكل وأضرار مختلفة، لذا يجب صيانتها بانتظام. تجمع ورشة عمل AHASS بوترا جايا تونجولولولونج البيانات لتحديد تشخيص الأضرار. بعض البيانات التي تم جمعها هي نوع الدراجة النارية، وسنة شراء الدراجة النارية، وعدد الكيلومترات التي قطعتها الدراجة النارية، والشكاوى التي يعاني منها مستخدمو الدراجات النارية، والتي تم استخدامها في هذه الدراسة لتحديد تشخيص التلف. يهدف هذا البحث إلى التنبؤ بتلف محرك الدراجة النارية. الطريقة المستخدمة هي الشبكة العصبية ذات الانتشار الخلفي للشبكة العصبية. يستخدم هذا البحث 333 بيانات ثم يتم إجراء موازنة البيانات باستخدام طريقة RandomOverSampler. ثم يتم تقسيم البيانات باستخدام طريقة التحقق المتقاطع 10 أضعاف. الحقب الزمنية المستخدمة في هذه الدراسة هي 500 و1000 و2000، بمعدل تعلم 0.001. في هذه الدراسة، هناك أيضًا 5 أنواع من النماذج بأعداد مختلفة من الطبقات المخفية. لتقييم أداء النموذج، يتم إجراء القياسات باستخدام مصفوفة الارتباك. يُظهر التحليل أن تأثير الطبقة المخفية والحقبة الزمنية ومعدل التعلم على دقة نماذج الشبكات العصبية مترابط. تعمل إضافة الطبقات المخفية على تحسين قدرة النموذج على التقاط أنماط البيانات المعقدة. بشكل عام، أعطى النموذج الذي يحتوي على 12 طبقة مخفية أفضل نتيجة دقة بنسبة 81.61%. مكّن العددان 1000 و2000 النموذج من تحقيق الدقة المثلى مع تحسن كبير في الأداء من الحلقة 500. بالإضافة إلى ذلك، أدى معدل التعلم 0.001 إلى تحقيق أعلى دقة مقارنةً بقيم معدلات التعلم الأخرى الأكبر. يُظهر المزيج الأمثل لهذه المعلمات أن الضبط المناسب للمعامل الفائق أمر بالغ الأهمية في تحسين أداء النموذج.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sepeda motor merupakan salah satu jenis kendaraan yang banyak digunakan oleh masyarakat sebagai alat transportasi. Harga yang terjangkau dan lebih hemat bahan bakar dibandingkan dengan kendaraan bermotor lainnya menjadi alasan utama penggunaan sepeda motor oleh masyarakat. Selain itu, sepeda motor dipilih karena efisiensi dalam pemakaiannya dan lebih cepat karena dapat lebih mudah menghindari kemacetan. Oleh karena itu, dalam penggunaan sepeda motor kemungkinan besar membutuhkan perawatan yang rutin (Epriliani et al., 2022).

Mesin motor yang bekerja dalam kondisi berat seringkali rentan terhadap berbagai masalah dan kerusakan. Masalah pada mesin motor tidak hanya dapat mengurangi kinerja kendaraan, tetapi juga dapat menyebabkan kerugian finansial dan risiko kecelakaan. Permasalahan-permasalahan ini dapat terkait dengan performa mesin, konsumsi bahan bakar, sistem pembakaran, serta kebisingan sehingga berdampak pada kenyamanan berkendara. Banyak pemilik kendaraan bermotor yang tidak mengetahui atau kurang memahami permasalahan-permasalahan yang terjadi pada mesin kendaraannya serta bagaimana cara mengatasinya (Purnama et al., 2007).

Masalah yang dialami saat ini yaitu keterbatasan informasi mengenai kerusakan sepeda motor masih membingungkan, terlebih lagi bagi pengguna sepeda motor yang tidak memiliki pengetahuan mengenai jenis-jenis kerusakan

pada mesin sepeda motor (Epriliani et al., 2022). Keterbatasan pengetahuan tersebutmembuat kurangnya kesadaran pengendara motor terhadap gejala-gejala kerusakan dan akan berdampak pada semakin parahnya kerusakan yang dialami oleh mesin motor apabila tidak segera diidentifikasi dan ditangani secara tepat.

Dalam Islam sendiri melarang manusia untuk melakukan kerusakan seperti dalam *Q.S. Al-Baqarah (2)* ayat 11-12:

وَاِذَا قِيْلَ لَهُمْ لَا تُفْسِدُوْا فِى الْاَرْضِۙ قَالُوْٓا اِنَّمَا نَحْنُ مُصْلِحُوْنَ ١١ اَلَآ ۗاِنَّهُمْ هُمُ الْمُفْسِدُوْنَ وَلٰكِنْ لَّا يَشْعُرُوْنَ ١٢

“Dan apabila dikatakan kepada mereka, "Janganlah berbuat kerusakan di bumi!" Mereka menjawab, "Sesungguhnya kami justru orang-orang yang melakukan perbaikan." Ingatlah, sesungguhnya merekalah yang berbuat kerusakan, tetapi mereka tidak menyadari.” (Q.S. Al-Baqarah: 11-12)

Kandungan makna ayat ini dapat diarahkan pada segala bentuk perbuatan yang menimbulkan kerusakan, sedangkan orang-orang yang melakukan kerusakan tersebut tidak menyadarinya dan enggan untuk mengakui perbuatan mereka. Dalam tafsir Al-Kabir secara tegas menyatakan bahwa yang dimaksud keruskan adalah segala sesuatu yang dalam hal itu hilangnya kebermanfaatan atau kebaikan. Dalam hal ini kurangnya kesadaran pengguna motor terhadap gejala-gejala kerusakan yang terjadi pada motornya merupakan suatu tindakan yang dapat menimbulkan kerusakan pada sepeda motor yang dimilikinya sehingga dapat mengurangi atau menghilangkan kebermanfaatannya.

Berdasarkan latar belakang permasalahan yang telah dijelaskan, maka dilakukan penelitian untuk memprediksi kerusakan mesin motor menggunakan metode *Neural Network*. Penelitian ini dilakukan dengan tujuan agar pengguna bisa

memperoleh informasi mengenai kerusakan yang terjadi berdasarkan variabel *input* seperti keluhan atau gejala yang dialami, tipe motor, usia motor, dan jumlah kilometer yang sudah ditempuh. *Neural Network* merupakan metode yang memiliki cara kerja yang mirip dengan cara kerja pada otak manusia dalam mengolah informasi. Perhitungan non-linear dasar yang saling berhubungan satu sama lain, atau bisa disebut sebagai *neuron* digunakan pada metode ini sehingga memiliki kemiripan dengan jaringan syaraf manusia (Kurniawan et al., 2023).

Neural Network merupakan metode yang dirancang untuk memecahkan permasalahan yang berkaitan dengan pengenalan pola atau klasifikasi (Hizham et al., 2018). Salah satu metode *Neural Network* yang digunakan pada penelitian ini adalah algoritma *Backpropagation,* karena paling umum digunakan dan bekerja secara iteratif menggunakan dataset dan membandingkan nilai prediksi dari jaringan dengan setiap sampel data (Kurniawan et al., 2023).

Epriliani et al. (2022) melakukan penelitian dengan topik yang serupa dengan menggunakan metode *Naïve Bayes*. Pada penelitian ini, *testing* dilakukan dengan menginputkan 2 gejala berbeda yaitu kerusakan pada CVT, dan kerusakan pada kampas kopling. Kemudian dilakukan hitung nilai presentase dengan *output* prediksi kerusakan pada kampas kopling dengan persentase sebesar 66,67%. Penelitian lain yang dilakukan oleh Romahadi et al. (2022) tentang perancangan sistem diagnosis getaran motor menggunakan *Neural Network Backpropagation* menyatakan bahwa hasil dari pelatihan dan pengujian ANN terbilang sangat baik dengan nilai *Cross-Entropy* paling besar 9.94, memiliki 0 nilai % *Error*, nilai MSE terbesar 8.33e-6 dan regresi terkecil 0.998.

Berdasarkan penelitian sebelumnya yang telah dijelaskan, metode *Neural Network Backpropagation* dipilih sebagai metode untuk digunakan dalam penelitian klasifikasi kerusakan mesin motor.

1.2 Rumusan Masalah

Masalah yang dibahas dalam penelitian ini adalah bagaimana cara memprediksi jenis kerusakan pada mesin sepeda motor berdasarkan variabel *input*, yaitu gejala yang muncul, jenis motor, usia motor, dan jarak tempuh.

1.3 Batasan Masalah

Penelitian ini dibatasi pada penggunaan data servis sepeda motor yang diperoleh dari AHASS Putra Jaya Tunggulwulung. Data tersebut dikumpulkan melalui *googleform* yang dibuat oleh peneliti dan diisi oleh admin serta mekanik bengkel.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk merancang model *machine learning* menggunakan metode *Neural Network Backpropagation* guna memprediksi jenis kerusakan pada mesin sepeda motor, sesuai dengan permasalahan yang telah diidentifikasi.

1.5 Manfaat Penelitian

Manfaat yang bisa didapat melalui penelitian ini adalah sebagai berikut:

1. Penelitian ini dapat meningkatkan pemahaman dan pengetahuan pengguna sepeda motor mengenai jenis-jenis kerusakan yang mungkin terjadi pada

mesin sepeda motor. Dengan demikian, pengguna dapat lebih proaktif dalam melakukan perawatan preventif dan mengidentifikasi masalah yang mungkin timbul, sehingga mengurangi risiko kerusakan yang lebih parah.

2. Penelitian ini dapat berkontribusi pada pengembangan teknologi prediksi kerusakan mesin sepeda motor yang lebih baik di masa depan, dengan memahami pola kerusakan yang sudah ada

BAB II

STUDI PUSTAKA

2.1 Penelitian Terkait

Penelitian sebelumnya yang dilakukan oleh Epriliani et al. (2022) mencoba memprediksi kerusakan sepeda motor pada bengkel Citra Djaya Motor. Pada penelitian ini, *testing* dilakukan dengan menginputkan 2 gejala berbeda yaitu kerusakan pada CVT, dan kerusakan pada kampas kopling. Kemudian dilakukan hitung nilai persentase dengan *output* prediksi kerusakan pada kampas kopling dengan persentase sebesar 66,67%.

Tabel 2. 1 Penelitian Terkait

Judul	Hasil	Persamaan	Perbedaan
Aplikasi Diagnosa Kerusakan Sepeda Motor Menggunakan Metode *Naïve Bayes* dan Rekomendasi Kesesuaian Bengkel (Muriyatmoko et al., 2021)	Penelitian ini menghasilkan sebuah aplikasi berbasis android yang memberikan informasi kerusakan yang terjadi pada sepeda motor dan informasi bengkel di kecamatan Ponorogo yang sesuai. Berdasarkan	Memprediksi kerusakan pada sepeda motor	Menggunakan metode *Naïve Bayes*
Perancangan sistem diagnosis getaran motor menggunakan jaringan saraf tiruan propagasi mundur (Romahadi et al., 2022)	Pelatihan dan pengujian ANN menghasilkan nilai Cross-Entropy paling besar 9.94, memiliki 0 nilai % Error, nilai MSE terbesar 8.33e-6 dan regresi terkecil 0.998.	Penggunaan metode *Neural Network Backpropagation* untuk diagnosa	Objek penelitian adalah getaran pada motor DC 12 kW
Prediction of Civil Servant Performance Allowances Using the Neural Network Backpropagation Method (Kurniawan et al., 2023)	Model terbaik menggunakan 8 *hidden layer*, rasio data *split* 50:50, menghasilkan akurasi sebesar 70,90%	Penggunaan metode *Neural Network Backpropagation*	Objek penelitian adalah kerusakan mesin motor Honda

Judul	Hasil	Persamaan	Perbedaan
Analisis algoritma *back propagation* dalam prediksi angka kemiskinan di Indonesia (Syaharuddin et al., 2019)	Menghasilkan prediksi tahun 2020 jumlah penduduk miskin di 33 Provinsi di Indonesia yaitu sebesar 332.005 jiwa. Adapun parameter akurasi dari arsitektur Back Propagation yakni dengan memperoleh MSE sebesar 0,119 dan MAPE sebesar 2,298. Abstract:	Penggunaan metode *Neural Network Backpropagation*	Memprediksi angka kemiskinan di Indonesia
Prediksi Data *Time-series* menggunakan Jaringan Syaraf Tiruan Algoritma *Backpropagation* Pada Kasus Prediksi Permintaan Beras (Marthasari et al., 2021)	Jaringan syaraf tiruan *backpropagation* menunjukkan prediksi yang baik yang pada penelitian ini hasil terbaik terdapat pada arsitektur 7-50-200-300-250-300-1 dengan hasil MSE = 0.001278, RMSE = 0,301950 di proses pelatihan and hasil MSE = 0.002391, RMSE = 0,204972 di proses pengujian.	Penggunaan metode *Neural Network Backpropagation*	Memprediksi permintaan beras berdasarkan data *time-series*
Sistem Pakar Diagnosa Kerusakan Motor Honda *Matic* Menggunakan Metode *Forward Chaining* (Akhsa, 2024)	Sistem dapat menganalisis kerusakan berdasarkan gejala-gejala yang dimasukkan oleh mekanik berdasarkan 23 data gejala dan 21 jenis kerusakan yang biasanya timbul pada kendaraan bermotor, yang dirangkai menjadi pohon keputusan yang dimasukan kedalam sistem.	Memprediksi kerusakan pada sepeda motor Honda berdasarkan gejala dan kelas target yang banyak (*multi-class*)	Menggunakan metode *forward chaining*
Sistem Pakar Diagnosa Kerusakan Mesin Sepeda Motor *Matic* Berbasis *Web* Menggunakan *Certainty Factor* (Dedi Saputra, Deasy Purwaningtias, 2018)	Penerapan metode *Certainty Factor* pada basis aturan menghasilkan derajat keyakinan terhadap masing-masing diagnosa kerusakan.	Memprediksi kerusakan pada mesin sepeda motor	Menggunakan metode *Certainty Factor*

Judul	Hasil	Persamaan	Perbedaan
Sistem Pakar Diagnosa Kerusakan Pada Vespa *Matic* Menggunakan Metode Naïve Bayes Berbasis Web (Studi kasus : PT. Dwi Pratama Mandiri) (Faisal et al., 2024)	Sistem pakar diagnosa kerusakan pada vespa Matic menggunakan metode *naïve bayes* berbasis web dapat menjadi solusi yang efektif bagi konsumen dan mekanik dalam meningkatkan pengetahuan mengenai kerusakan yang terjadi pada kendaraannya. Dengan	Memprediksi kerusakan pada mesin sepeda motor	Menggunakan metode *Naïve Bayes*

2.2 Sistem Prediksi

Menurut Syofian dan Nugraha (2021) sistem adalah kata yang berasal dari bahasa latin yaitu *systema* dan Bahasa Yunani *sistema*. Sistem adalah kumpulan elemen-elemen yang saling terkait dan bekerja sama dengan tujuan memperoleh masukan (*input*) mengolah masukan tersebut hingga menghasilkan *output* yang diinginkan. Sedangkan prediksi adalah proses memperkirakan tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi yang sudah dimiliki di masa lalu dan sekarang secara sistematis, agar dapat memperkecil kesalahan yang terjadi dengan hasil perkiraan. Dalam hal ini, prediksi tidak harus memberikan hasil secara akurat, melainkan berusaha untuk mencari kemungkinan hal yang akan terjadi sedekat mungkin (Rohmawati et al., 2017). Dalam surah *al-An'am ayat* 142 yang berbunyi:

وَمِنَ الْاَنْعَامِ حَمُوْلَةً وَّفَرْشًاۗ كُلُوْا مِمَّا رَزَقَكُمُ اللّٰهُ وَلَا تَتَّبِعُوْا خُطُوٰتِ الشَّيْطٰنِۗ اِنَّهٗ لَكُمْ عَدُوٌّ مُّبِيْنٌۙ

"Di antara hewan-hewan ternak itu ada yang dijadikan pengangkut beban dan ada (pula) yang untuk disembelih. Makanlah rezeki yang diberikan Allah kepadamu.

Janganlah kamu mengikuti langkah-langkah setan. Sesungguhnya dia adalah musuh yang nyata bagimu." (Q.S. Al-An'am: 142)

Allah menganugerahkan manusia dengan akal untuk memanfaatkan ciptaan-Nya dengan baik. Sistem prediksi, yang didasarkan pada analisis data, machine learning, dan teknologi lainnya, merupakan aplikasi langsung dari kemampuan berpikir manusia. Sistem ini memerlukan ketelitian dan pemahaman yang mendalam terhadap data dan pola kerusakan agar hasil prediksinya dapat diandalkan. Jadi, sistem prediksi merupakan sebuah sistem yang mengolah masukan untuk memperkirakan tentang sesuatu berdasarkan informasi yang sudah dimiliki agar menghasilkan *output* sesuai yang diinginkan.

2.3 Mesin Sepeda Motor

Dalam surah Al-Baqarah ayat 164 yang berbunyi:

اِنَّ فِيْ خَلْقِ السَّمٰوٰتِ وَالْاَرْضِ وَاخْتِلَافِ الَّيْلِ وَالنَّهَارِ وَالْفُلْكِ الَّتِيْ تَجْرِيْ فِى الْبَحْرِ بِمَا يَنْفَعُ النَّاسَ وَمَآ اَنْزَلَ اللّٰهُ مِنَ السَّمَآءِ مِنْ مَّآءٍ فَاَحْيَا بِهِ الْاَرْضَ بَعْدَ مَوْتِهَا وَبَثَّ فِيْهَا مِنْ كُلِّ دَآبَّةٍ ۖ وَّتَصْرِيْفِ الرِّيٰحِ وَالسَّحَابِ الْمُسَخَّرِ بَيْنَ السَّمَآءِ وَالْاَرْضِ لَاٰيٰتٍ لِّقَوْمٍ يَّعْقِلُوْنَ

"Sesungguhnya pada penciptaan langit dan bumi, pergantian malam dan siang bahtera yang berlayar di laut dengan (muatan) yang bermanfaat bagi manusia, apa yang Allah turunkan dari langit berupa air, lalu dengannya Dia menghidupkan bumi setelah mati (kering), dan Dia menebarkan di dalamnya semua jenis hewan, dan pengisaran angin dan awan yang dikendalikan antara langit dan bumi, (semua itu) sungguh merupakan tanda-tanda (kebesaran Allah) bagi kaum yang mengerti." (Q.S. Al-Baqarah: 164)

Ayat ini menekankan bahwa ciptaan Allah, baik di langit maupun di bumi, adalah tanda kebesaran-Nya. Dalam konteks mesin sepeda motor, setiap bagian dari mesin sepeda motor yang meliputu silinder, piston, hingga sistem pembakaran adalah hasil dari pemahaman mendalam tentang ilmu pengetahuan dan teknik yang

Allah anugerahkan kepada manusia. Mesin sepeda motor mencerminkan kemampuan manusia untuk memanfaatkan sumber daya alam, seperti bahan bakar, logam, dan energi, untuk menciptakan alat yang berguna.

Mesin sepeda motor adalah bagian dari kendaraan sepeda motor yang berperan menghasilkan tenaga untuk menggerakkan sepeda motor dengan mengubah energi panas untuk melakukan kinerja mekanik (Ghaly & Winoko, 2019). Mesin motor menggunakan satu atau lebih silinder yang di dalamnya terdapat piston yang bergerak secara bolak-balik. Pada umumnya, mesin sepeda motor menggunakan prinsip pembakaran dalam, dimana campuran bahan bakar dan udara terbakar di dalam ruang bakar sehingga menghasilkan gerakan piston untuk disalurkan ke roda melalui perantara transmisi.

2.4 *Neural Network Backpropagation*

2.4.1 Pengertian *Neural Network*

Neural Network adalah salah satu arsitektur pemrosesan informasi yang beroperasi secara paralel dan terdistribusi, terdiri dari elemen-elemen pemroses dengan memori lokal. Setiap elemen dapat melakukan operasi berdasarkan informasi lokal yang saling terhubung (Hecht-Nielsen, 1989). *Neural Network* atau Jaringan Syaraf Tiruan (JST) adalah sebuah sistem yang dirancang untuk memproses informasi dengan karakteristik yang menyerupai cara kerja otak manusia (Almais et al., 2022).

Jaringan Syaraf Tiruan berfungsi dengan mempelajari pola-pola data yang telah tersedia sebelumnya, sehingga ketika menerima sinyal input, sistem dapat

memprosesnya untuk menghasilkan output yang tepat (Suhartanto et al., 2017). Dalam surah *Al-Baqarah* ayat 31 yang berbunyi:

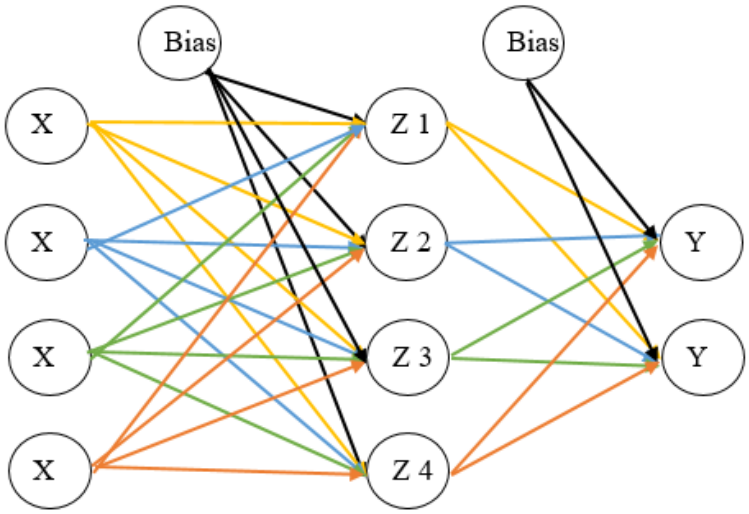
وَعَلَّمَ اٰدَمَ الْاَسْمَآءَ كُلَّهَا ثُمَّ عَرَضَهُمْ عَلَى الْمَلٰۤىِٕكَةِ فَقَالَ اَنْۢبِـُٔوْنِيْ بِاَسْمَآءِ هٰٓؤُلَآءِ اِنْ كُنْتُمْ صٰدِقِيْنَ

"Dia mengajarkan kepada Adam nama-nama (benda) seluruhnya, kemudian Dia memperlihatkannya kepada para malaikat, seraya berfirman, "Sebutkan kepada-Ku nama-nama (benda) ini jika kamu benar!"" (Q.S. Al-Baqarah: 31)

Dalam ayat ini, Allah mengajarkan Adam nama-nama segala sesuatu, yang mencerminkan pentingnya pengetahuan. Sistem jaringan syaraf tiruan berfungsi berdasarkan prinsip pembelajaran dari data. Jaringan ini belajar mengenali pola dan hubungan dalam data, mirip dengan proses belajar yang dilakukan manusia. Proses pembelajaran ini adalah inti dari kemampuan jaringan syaraf untuk memprediksi dan mengambil keputusan berdasarkan informasi yang tersedia. Jaringan syaraf tiruan adalah teknologi yang dikembangkan dengan mencontoh mekanisme otak manusia, bertujuan untuk meniru cara otak memproses dan mengelola informasi.

2.4.2 Pengertian *Backpropagation*

Neural Network Backpropagation adalah metode pembelajaran terkontrol (*supervised learning*) yang mengatur bobot pada neuron di lapisan tersembunyi (*hidden layer*) menggunakan arsitektur *multilayer*. Teknik ini bertujuan untuk mengurangi nilai *error* antara *output* prediksi dan *output* aktual (Kurniawan et al., 2023). *Backpropagation* mempunyai beberapa elemen yang ada dalam satu atau lebih *hidden layer* yang digambarkan pada gambar 2.1



Gambar 2. 1 Arsitektur *Neural Network Backpropagation*

Gambar 2.1 merupakan jaringan yang mempunyai *input* layer yang ditandai dengan huruf x, *hidden layer* ditandai huruf z, dan *output layer* ditandai huruf y, serta nilai bias yang berupa masukkan tambahan pada layer berikutnya yang akan selalu bernilai 1. Pada gambar 2.1 memiliki *input* layer sebanyak 4 *Neuron input*, sedangkan *hidden layer* tidak memiliki aturan jumlah khusus karena akan dilakukan pengujian dengan mengubah nilai dari *hidden layer* tersebut secara bertahap seperti penelitian yang dilakukan oleh Almais et al. (2022), serta pada *output layer* mempunyai sebanyak dua *neuron*.

2.4.3 Langkah-Langkah Algoritma *Backpropagation*

Terdapat 3 fase pada proses metode *Backpropagation* yang dibagi menjadi fase pertama, yaitu memasukkan data melalui *input* jaringan (*feedforward*). Setelah data dimasukkan, selanjutnya memasuki fase kedua yaitu melakukan perhitungan dan propagasi mundur *error* yang bersangkutan. Sealnjutnya masuk fase ketiga yaitu melakukan pembaharuan pada bobot dan bias. Berikut adalah langkah-

langkah algoritma *Backpropagation* berdasarkan buku dari Siang (2009) secara terperinci:

a. Fase 1: *Feedforward*

Langkah 0: Menginisialisasi bobot.

Langkah 1: Apabila kondisi penghentian (*stop*) belum terpenuhi (bernilai salah), maka lakukan langkah 2-9.

Langkah 2: Untuk setiap pasang data *training*, lakukan langkah 3-8.

Langkah 3: Setiap unit *input* $(X_i, i = 1, \dots, n)$ menerima sinyal dan meneruskan ke seluruh unit tersembunyi di atasnya.

Langkah 4: Hitung semua *output* di *hidden layer* $(Z_j, j = 1, \dots, n)$.

$$Z_net_j = v_{j0} + \sum_{i=0}^{n} x_i v_{ji} \tag{2.1}$$

Lalu hitung fungsi aktivasi *sigmoid biner*.

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

Keluaran dari fungsi aktivasi tersebut kemudian dikirim ke semua unit *output* yang tersembunyi.

Langkah 5: Lakukan perhitungan semua *output* jaringan di unit $(y_k k = 1, 2, \dots, m)$.

$$y_net_k = w_{k0} + \sum_{j=1}^{p} z_j w_{kj} \tag{2.3}$$

Kemudian hitung sinyal *output* menggunakan fungsi aktivasi *softmax*:

$$Softmax(Yk) = \frac{\exp(y_i)}{\sum_j \exp(y_j)} \quad (2.4)$$

Langkah 6: Setelah nilai *output* dari hasil fungsi *softmax* didapatkan, selanjutnya adalah menghitung *loss* atau *error* menggunakan *Categorical Cross-Entropy Loss*.

$$Loss = -\sum_{j=1}^{C} y \log(yk) \quad (2.5)$$

b. Fase 2: *Backpropagation*

Langkah 6: Hitung faktor δ unit *output* berdasarkan kesalahan di setiap unit *output* y_k $(k = 1, 2, \dots, m)$.

$$\delta_k = (t_k - y_k) f'(y_{net_k}) = (t_k - y_k) y_k (1 - y_k) \quad (2.6)$$

Perubahan bobot w_{kj} dihitung dengan memanfaatkan unit error δ_k dan laju percepatan α. Rumus perubahannya adalah sebagai berikut:

$$\Delta w_{kj} = \alpha \delta_k z_j; k = 1, 2, \dots, m; j = 0, 1, \dots, p \quad (2.7)$$

Langkah 7: Hitung faktor δ pada unit tersembunyi z_j $(j = 0, 1, \dots, p.)$ dengan rumus berikut:

$$\delta_net_j = \sum_{k=1}^{m} \delta_k w_{kj} \dots \quad (2.8)$$

Faktor δ unit tersembunyi:

$$\delta_j = \delta_net_j f'(z_net_j) = \delta_net_j z_j (1 - z_j) \quad (2.9)$$

Hitung suku perubahan bobot v_{ji} (yang digunakan untuk mengubah bobot v_{ji}).

$$v_{ji} = \alpha\delta j x_i; j = 1, 2, \dots, p; i = 0, 1, \dots n \quad (2.10)$$

c. Fase 3: Perubahan Bobot

Langkah 8: Hitung semua perubahan bobot yang menuju unit *output*:

$$w_{kj}(baru) = w_{kj}(lama) + \Delta w_{kj} \quad (k = 1,2, \dots, m; j = 0,2, \dots p) \quad (2.11)$$

Perubahan bobot bias yang menuju ke unit tersembunyi:

$$v_{ji}(baru) = v_{ji}(lama) + \Delta v_{ji} \quad (j = 1,2, \dots, p; i = 0,2, \dots n) \quad (2.12)$$

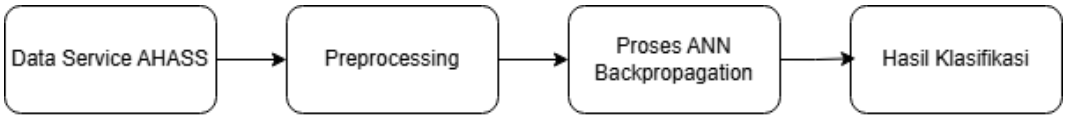
Langkah 9: Lakukan evaluasi untuk memastikan apakah kondisi penghentian telah tercapai, yaitu ketika nilai *error* yang dihasilkan lebih kecil dibandingkan dengan nilai *error* referensi.

BAB III

METODE PENELITIAN

3.1 Desain Penelitian

Desain penelitian menggambarkan rencana implementasi penelitian yang akan dilakukan. Rancangan ini terdiri dari empat komponen utama, yaitu *input* data, *preprocessing*, proses jaringan syaraf tiruan (*ANN*) *backpropagation*, dan *output* hasil, seperti yang ditunjukkan pada gambar 3.1 berikut.



Gambar 3. 1 Desain Penelitian

Berdasarkan Gambar 3.1, *input* yang digunakan dalam penelitian ini berasal dari data servis pelanggan di AHASS Tunggulwulung. Data yang dimasukkan meliputi tipe motor, usia motor, jumlah kilometer yang telah ditempuh, keluhan, dan diagnosa kerusakan. Selanjutnya, data tersebut diproses melalui beberapa tahapan berikut untuk menghasilkan output.

3.1.1 *Input* Data

Pada penelitian ini menggunakan data primer yang diperoleh secara langsung dari AHASS Putra Jaya Tunggulwulung. Data diperoleh dari *google form* yang disediakan oleh peneliti untuk diisi oleh admin dan mekanik selama 1 bulan. Data terdiri dari beberapa atribut berikut.

Tabel 3. 1 Data Servis di AHASS Putra Jaya Tunggulwulung 2023

Index ke-	Tipe motor	Model motor	Tahun pembuatan	Kilometer yang telah ditempuh	Keluhan yang dialami customer	Diagnosa oleh mekanik
1	Matic	Vario 125	2020	1748	Stater susah nyala	Aki
2	Matic	Beat	2014	61084	Keluar asap putih	Piston
3	Matic	Beat	2015	18596	Tenaga tiba tiba hilang di kecepatan atas	Rantai kamprat
4	Matic	Vario 160	2022	9333	Getar awal jalan	Kampas ganda
…	…	…	…		…	…
331	Non-Matic	CB 150	2017	24430	Bunyi dek-dek saat di gas	Roller
332	Non-Matic	Mega Pro	2012	106404	Sering mogok	Aki
333	Non-Matic	CB 150	2014	58665	Brebet di kecepatan menengah	Filter udara
334	Non-Matic	CBR 150	2019	27660	Bunyi kletek-kletek di mesin	Piston

Pada data tersebut, sebanyak 23 keluhan dan 17 jenis kerusakan berbeda dialami oleh berbagai *customer*. Keluhan-keluhan yang dialami oleh *customer* tersebut merupakan kriteria yang akan dijadikan sebagai salah satu acuan dalam penentuan kerusakan apa yang terjadi pada mesin motor. Kriteria-kriteria tersebut akan mendukung sistem yang dikembangkan untuk memprediksi kerusakan pada mesin sepeda motor. Kriteria yang diterapkan dapat dilihat pada tabel 3.2 dan 3.3 berikut ini:

Tabel 3. 2 Data Kriteria Keluhan

Kode	Keterangan
G1	Sering mogok
G2	Kompresi hilang
G3	Stater susah nyala
G4	Double starter gak bisa
G5	Akselerasi berat
G6	Nyendat-nyendat saat jalan
G7	Bunyi kletek-kletek di mesin
G8	Suara mesin kasar
G9	Keluar asap putih
G10	Tenaga tiba-tiba hilang di kecepatan atas

Kode	Keterangan
G11	Getar awal jalan
G12	Brebet saat di gas
G13	Boros
G14	Brebet saat di tanjakan
G15	Brebet di kecepatan atas
G16	Tarikan awal delay
G17	Bunyi dek-dek saat di gas
G18	Tarikan kurang responsif
G19	Ada suara denging saat gas diturunkan
G20	Sering mati saat tidak di gas
G21	Indikator suhu nyala
G22	Bunyi berdecit
G23	Brebet di kecepatan menengah

Tabel 3. 3 Data Alternatif Kerusakan

Kode	Keterangan
K1	Aki
K2	Sekring
K3	Dinamo
K4	Komponen pengapian
K5	Piston
K6	Filter udara
K7	Throttle body
K8	Kampas kopling
K9	Oli
K10	Cvt
K11	Fuel pump
K12	Roller
K13	Rantai kamprat
K14	Kampas ganda
K15	Bearing box
K16	Radiator
K17	Coolant

3.1.2 *Preprocessing*

Dalam penelitian ini perlu dilakukan *preprocessing* data yang bertujuan untuk membersihkan data, mengubah tipe data agar mempermudah proses pengolahan data.

Tabel 3. 4 *Labeling* Variabel

No	Kriteria	Deskripsi	Value
1	Tipe motor	Matic	1
		Non-matic	2
2	Keluhan	Sering mogok	1
		Kompresi hilang	2

No	Kriteria	Deskripsi	Value
		Stater susah nyala	3
		Double starter gak bisa	4
		Akselerasi berat	5
		Nyendat-nyendat saat jalan	6
		Bunyi kletek-kletek di mesin	7
		Suara mesin kasar	8
		Keluar asap putih	9
		Tenaga tiba-tiba hilang di kecepatan atas	10
		Getar awal jalan	11
		Brebet saat di gas	12
		Boros	13
		Brebet saat di tanjakan	14
		Brebet di kecepatan atas	15
		Tarikan awal delay	16
		Bunyi dek-dek saat di gas	17
		Tarikan kurang responsif	18
		Ada suara denging saat gas diturunkan	19
		Sering mati saat tidak di gas	20
		Indikator suhu nyala	21
		Bunyi berdecit	22
		Brebet di kecepatan menengah	23
3	Diagnosa	Aki	0
		Sekring	1
		Dinamo	2
		Komponen pengapian	3
		Piston	4
		Filter udara	5
		Throttle body	6
		Kampas kopling	7
		Oli	8
		Cvt	9
		Fuel pump	10
		Roller	11
		Rantai kamprat	12
		Kampas ganda	13
		Bearing gearbox	14
		Radiator	15
		Coolant	16

Pada tabel 3.4 merupakan data kriteria yang digunakan untuk menentukan kerusakan mesin motor. Data di atas akan digunakan sebagai *input* dan target sehingga sebelum digunakan, data perlu diolah terlebih dahulu. *Preprocessing* akan dilakukan melalui tiga tahap yaitu data *encoding, balancing data*, dan *split dataset* sebagai berikut:

1) *Data transformation*

Label encoding merupakan metode yang digunakan untuk mengubah data kategorikal yang berupa teks menjadi data numerik sehingga dapat mengubahnya menjadi bentuk yang dapat dibaca oleh algoritma pembelajaran (Herdian et al., 2024). Pada penelitian ini, proses *encoding* dilakukan pada data yang belum di-*balancing* menggunakan *library* Pandas.

```
tipe_mapping = {'Matic': 1,'Non-Matic': 2}
keluhan_mapping = {'STARTER SUSAH NYALA': 0,'DOUBLE STARTER GAK BISA': 1,
                   'SERING MOGOK': 2,'INDIKATOR SUHU NYALA': 3,'BOROS': 4,
                   'BUNYI BERDECIT': 5,'SERING MATI SAAT TIDAK DI GAS': 6,
                   'NYENDAT NYENDAT SAAT JALAN': 7,'KOMPRESI HILANG': 8,
                   'AKSELERASI BERAT': 9,'KELUAR ASAP PUTIH': 10,
                   'BUNYI KLETEK-KLETEK DI MESIN': 11,'TARIKAN AWAL DELAY': 12,
                   'BREBET SAAT DI TANJAKAN': 13,'BREBET DI KECEPATAN MENENGAH': 14,
                   'BREBET DI KECEPATAN ATAS': 15,'BREBET SAAT DI GAS': 16,
                   'SUARA MESIN KASAR': 17,'GETAR AWAL JALAN': 18,
                   'TARIKAN KURANG RESPONSIF': 19,'BUNYI DEK-DEK SAAT DI GAS': 21,
                   'TENAGA TIBA TIBA HILANG DI KECEPATAN ATAS': 20,
                   'ADA SUARA DENGING SAAT GAS DITURUNKAN': 21}
diagnosa_mapping = {'AKI': 0,'SEKRING': 1,'DINAMO': 2,'KOMPONEN PENGAPIAN': 3,
                    'PISTON': 4,'FILTER UDARA': 5,'THROTTLE BODY': 6,
                    'KAMPAS KOPLING': 7,'OLI': 8,'CVT': 9,
                    'FUEL PUMP': 10,'ROLLER': 11,'RANTAI KAMPRAT': 12,
                    'KAMPAS GANDA': 13,'BEARING GEARBOX': 14,
                    'RADIATOR': 15,'COOLANT': 16}

df['Tipe Motor'] = df['Tipe Motor'].map(tipe_mapping)
df['Keluhan'] = df['Keluhan'].map(keluhan_mapping)
df['Diagnosa'] = df['Diagnosa'].map(diagnosa_mapping)
```

Gambar 3. 2 Label *Encoding*

Tabel 3. 5 Data sebelum dilakukan *balancing*

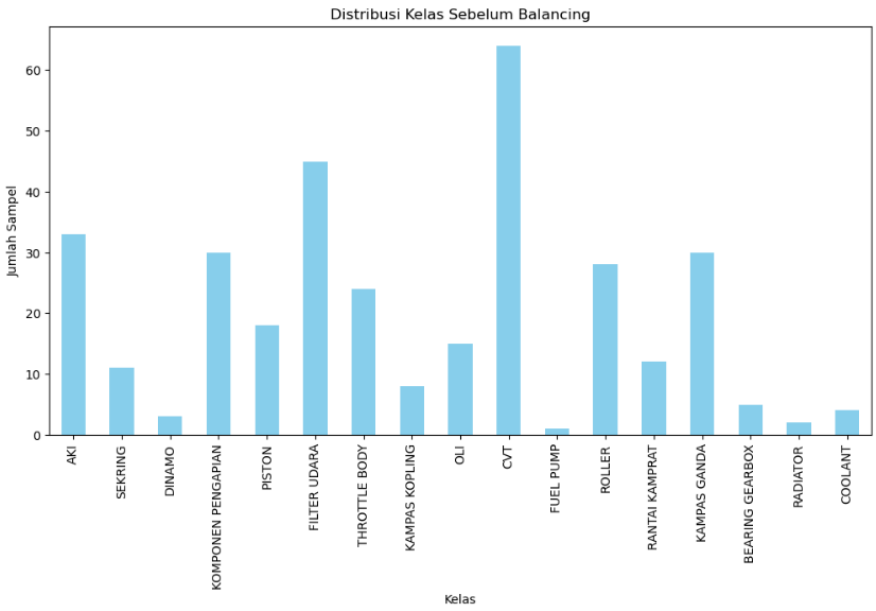
Data index ke-	Tipe motor	Usia motor	Jumlah kilometer	Keluhan	Diagnosa
0	1	4	1748	3	0
1	1	10	61084	9	4
2	1	9	18596	10	12
…	…	…	…	…	…
331	2	12	106404	1	0
332	2	10	58665	23	5
333	2	5	27660	7	4

2) *Balancing data*

Pada penelitian ini, *balancing* data menggunakan metode *random oversampling*. *Random oversampling* merupakan metode *oversampling* yang digunakan untuk mengatasi kelas dataset yang tidak seimbang. Melalui metode ini,

sampel data dari kelas minoritas akan diperbanyak secara random dengan cara menduplikat dari beberapa data yang sudah ada dalam kelas tersebut. Tujuan dari *balancing* data adalah untuk menciptakan data keseimbangan antara jumlah sampel di kelas mayoritas dan kelas minoritas, sehingga model pembelajaran mesin akan memiliki representasi yang seimbang dari semua kelas selama proses pelatihan (Sasongko, 2024).

Pada penelitian ini, terdapat ketidakseimbangan distribusi label pada dataset yang digunakan. Ketidakseimbangan distribusi label dapat dilihat pada gambar 3.3.



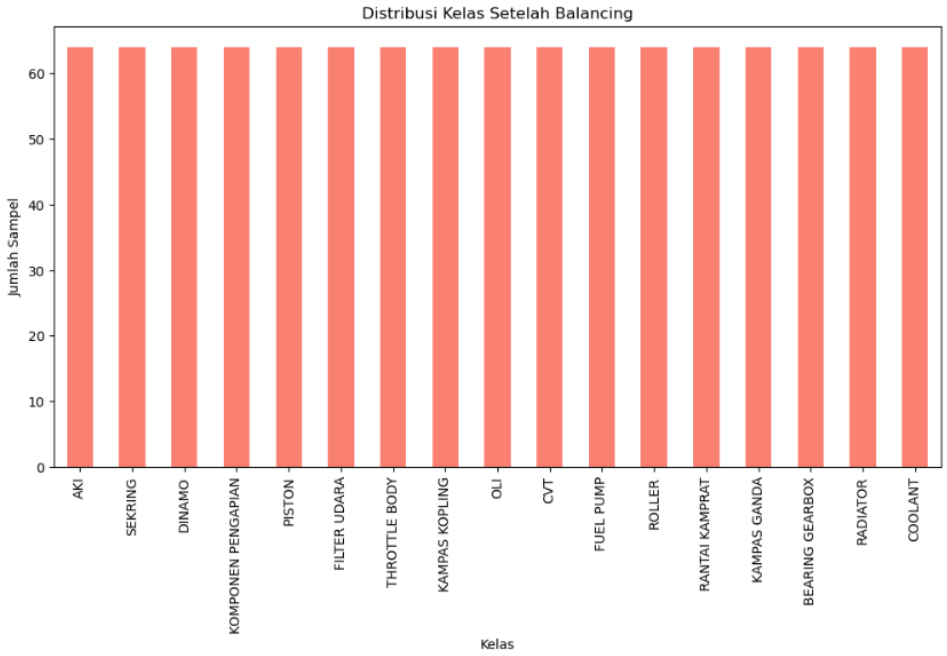
Gambar 3. 3 Distribusi kelas sebelum *balancing*

Untuk mengatasi masalah *imbalance data*, maka digunakan *library Python imbalanced-learn* (*imblearn*) dengan metode *RandomOverSampler*.

```
from imblearn.over_sampling import RandomOverSampler

X = df.drop(columns=['Diagnosa'])
y = df['Diagnosa']

ros = RandomOverSampler(random_state=None)
X_balanced, y_balanced = ros.fit_resample(X, y)

df_balanced = pd.concat([X_balanced, pd.Series(y_balanced, name='Diagnosa')], axis=1)
```

Gambar 3. 4 *Source code* untuk *balancing* data

Setelah melalui tahap *balancing*, data bertambah menjadi total 1088 data, dengan jumlah label pada masing-masing kelas sebanyak 64 data seperti pada gambar 3.4 berikut.



Gambar 3. 5 Distribusi kelas setelah *balancing*

3) *Split dataset*

Pada penelitian ini, dataset dibagi menjadi data *training* dan data *testing*. Data *training* digunakan untuk mengajarkan model pola yang ada dalam data sehingga model dapat membuat prediksi yang akurat. Data *testing* digunakan untuk mengevaluasi performa model setelah model dilatih dengan data *training* menggunakan *Backpropagation*. Model yang telah dilatih tidak melihat data *testing* selama proses pelatihan, sehingga data testing merupakan data baru bagi model. Data *testing* berfungsi untuk mengevaluasi generalisasi model, yaitu kinerja model untuk membuat prediksi yang akurat pada data baru yang tidak pernah dilihat sebelumnya.

```
X = df.drop(columns=['Diagnosa'])
y = df['Diagnosa']
```

Gambar 3. 6 *Source code* untuk memisahkan kolom diagnosa

Pada variabel X terdapat empat kolom yang mewakili variabel untuk menentukan kerusakan mesin motor. Variable y merupakan label yang merepresentasikan kerusakan yang terjadi. Pada penelitian ini, pembagian data menggunakan metode *10-Fold Cross Validation*, mengacu pada penelitian yang dilakukan oleh (Malakouti et al., 2023) . Tujuan dari penggunaan *10-Fold Cross Validation* adalah untuk mengukur kinerja model dengan lebih akurat, memastikan bahwa model memiliki kemampuan generalisasi yang baik, mengurangi *overfitting*, serta memberikan estimasi performa model yang lebih akurat.

```
# Initialize K-Fold cross-validation
k = 3
kf = KFold(n_splits=k, shuffle=True, random_state=seed_value)
fold = 1

# Split the data for the current fold
X_train_fold, X_test_fold = X.iloc[train_index], X.iloc[test_index]
y_train_fold, y_test_fold = y.iloc[train_index], y.iloc[test_index]
```

Gambar 3. 7 *Source Code* Data *Split* dengan *K-Fold Cross Validation*

3.1.3 Implementasi *Neural Network* Backpropagation

Data yang sudah diolah pada tahap sebelumnya, kemudian digunakan dalam proses berikut ini:

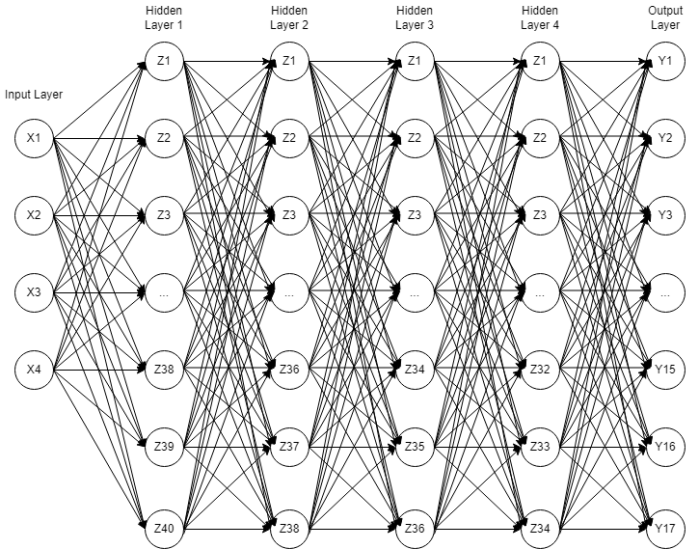
1. Menentukan jumlah *hidden layer*. Pada penelitian ini, jumlah *hidden layer* mulai dari 4 hingga 12 *hidden layer*. Jumlah ini ditetapkan berdasarkan penelitian sebelumnya yang dilakukan oleh (Kurniawan et al., 2023)
2. Membuat lapisan jaringan dan menentukan nilai bobot awal secara acak dan bias awal bernilai 1.

3. Menetapkan jumlah maksimum *epoch*. Pada penelitian ini, menggunakan *epoch* sebesar 500, 1000, dan 2000. Nilai *epoch* ditetapkan berdasarkan penelitian yang dilakukan oleh (Kaur dan Singh, 2016).
4. Menetapkan *learning rate*, yaitu kecepatan pembelajaran, di mana semakin tinggi nilai *learning rate* yang digunakan, maka semakin besar langkah yang diambil dalam proses pembelajaran. Pada penelitian ini, menggunakan *learning rate* sebesar 0.001
5. Membuat *class* dengan fungsi yang mendefinisikan fungsi aktivasi *sigmoid*, fungsi aktivasi *softmax*, perhitungan *loss categorical cross entropy*, dan *Optimizer Adam*.
6. Melakukan pelatihan model dengan data *training*, kemudian menghitung nilai *loss* atau *error*, memperbarui nilai bobot-bobot pada *Neural Network*.
7. Melakukan proses *testing* menggunakan data *testing* dengan tujuan untuk mengukur kemampuan model yang telah dilatih.
8. Mengukur hasil akurasi model menggunakan *confussion matrix multiclass*.

3.2 Arsitektur Jaringan

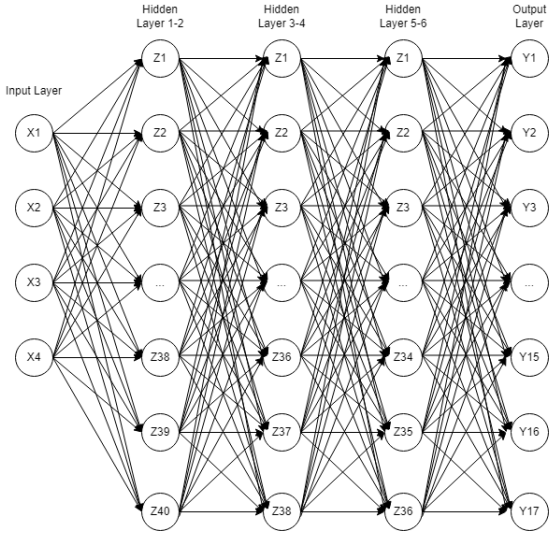
Beberapa model arsitektur jaringan telah disiapkan untuk digunakan dalam proses pelatihan dan pengujian. Dari berbagai model tersebut, model terbaik akan dipilih untuk digunakan dalam tahap pengujian. Model A memiliki 4 *neuron* pada input layer, 4 *hidden layer* dengan jumlah *neuron* yang berkurang secara bertahap: 40 *neuron* pada *hidden layer* pertama, 38 *neuron* pada *hidden layer* kedua, 36 *neuron* pada *hidden layer* ketiga, dan 34 *neuron* pada *hidden layer* keempat, serta

17 *neuron* pada *output layer*. Arsitektur jaringan Model A dapat dilihat pada gambar 3.6 berikut.



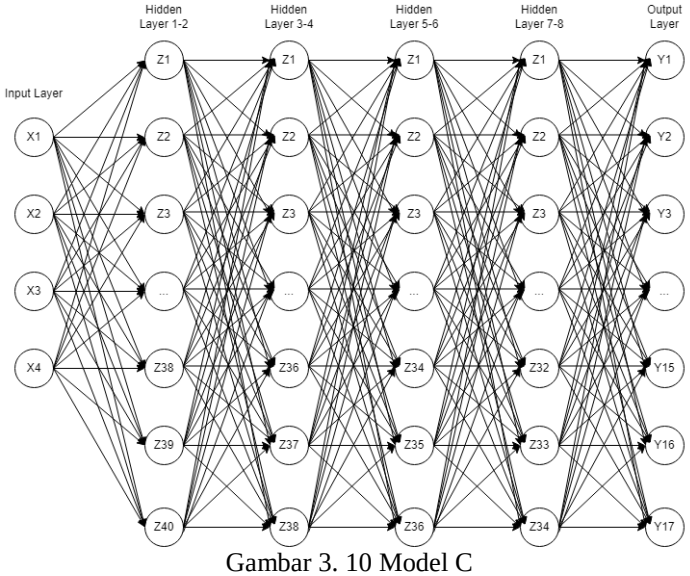
Gambar 3. 8 Model A

Model B memiliki arsitektur dengan 4 *neuron* pada lapisan *input*, 6 lapisan tersembunyi yang terdiri dari 40 *neuron* pada lapisan pertama dan kedua, 38 *neuron* pada lapisan ketiga dan keempat, serta 36 *neuron* pada lapisan kelima dan keenam. Model ini juga memiliki 17*neuron* pada lapisan *output*. Struktur jaringan Model B dapat dilihat pada gambar 3.7 berikut.



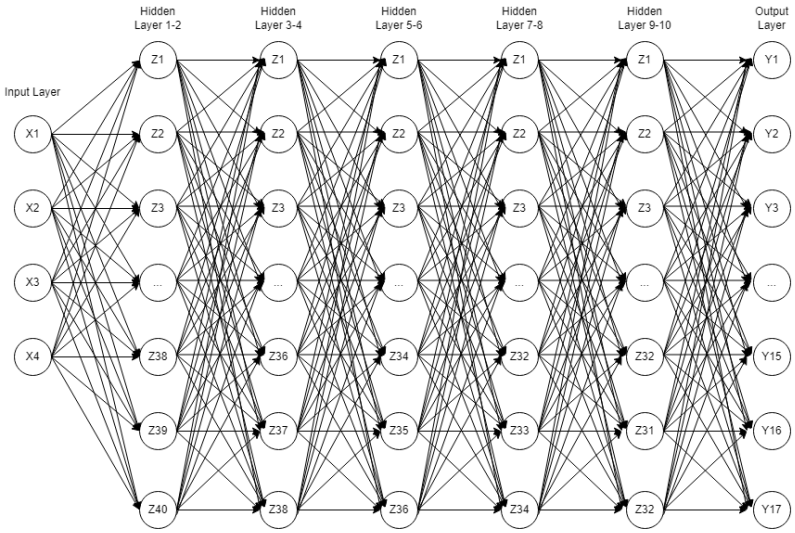
Gambar 3. 9 Model B

Model C memiliki arsitektur dengan 4 *neuron* pada lapisan *input*, 8 lapisan tersembunyi, dan 17 *neuron* pada lapisan *output*. Pada setiap dua lapisan tersembunyi pertama dan kedua, terdapat 40 *neuron*, sedangkan pada lapisan ketiga dan keempat masing-masing terdapat 38 *neuron*. Lapisan kelima dan keenam memiliki 36 *neuron*, dan lapisan ketujuh serta kedelapan masing-masing memiliki 34 *neuron*. Struktur jaringan Model C ditunjukkan pada gambar 3.8 berikut.



Gambar 3. 10 Model C

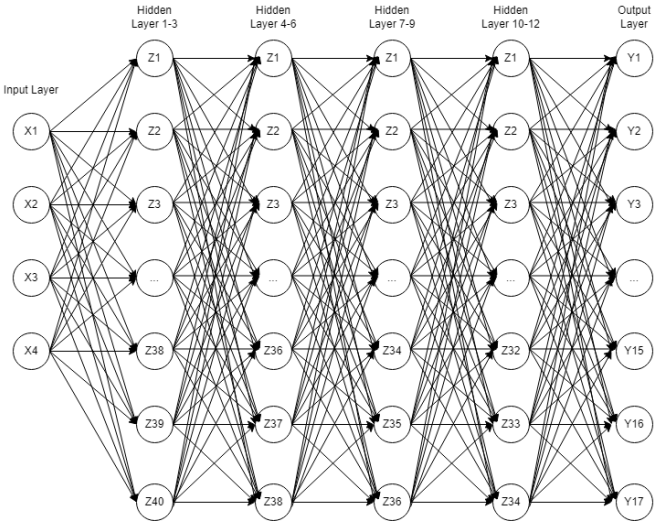
Model D memiliki arsitektur yang terdiri dari 4 *neuron* di lapisan *input*, 10 lapisan tersembunyi, dengan jumlah *neuron* yang berbeda pada setiap lapisan. Lapisan tersembunyi pertama dan kedua masing-masing memiliki 40 *neuron*, lapisan ketiga dan keempat memiliki 38 *neuron*, lapisan kelima dan keenam terdiri dari 36 *neuron*, lapisan ketujuh dan kedelapan memiliki 34 *neuron*, serta lapisan kesembilan dan kesepuluh masing-masing terdiri dari 32 *neuron*. Pada lapisan *output* terdapat 17 *neuron*. Struktur jaringan Model D dapat dilihat pada gambar 3.9 berikut.



Gambar 3. 11 Model D

Model E memiliki arsitektur jaringan yang terdiri dari 4 *neuron* pada *input* layer, 12 *hidden layer* dengan jumlah *neuron* yang bervariasi di setiap lapisan. Tiga lapisan pertama (*hidden layer* pertama, kedua, dan ketiga) masing-masing memiliki 40 *neuron*, sedangkan tiga lapisan berikutnya (*hidden layer* keempat, kelima, dan keenam) terdiri dari 38 *neuron*. Tiga lapisan selanjutnya (*hidden layer* ketujuh, kedelapan, dan kesembilan) memiliki 36 *neuron*, dan tiga lapisan terakhir (*hidden layer* kesepuluh, kesebelas, dan keduabelas) terdiri dari 34 *neuron*. Model ini

memiliki 17 *neuron* pada *output layer*. Struktur jaringan Model E dapat dilihat pada gambar 3.10 berikut.



Gambar 3. 12 Model E

3.3 Desain Eksperimen

Terdapat beberapa nilai *hyperparameter* optimal yang ditentukan dalam penelitian ini, yaitu:

1. Jumlah *hidden layer*
2. Jumlah *Neuron* dalam *hidden layer*
3. *Epoch* maksimum
4. *Learning rate* (α)

Pada *hyperparameter* jumlah *hidden layer*, dilakukan 6 variasi untuk mencari arsitektur *hidden layer* yang paling optimal dalam memprediksi data. Sementara itu, variasi jumlah *neuron* dalam *hidden layer* hanya memiliki 1 variasi, yaitu jumlah *neuron* yang berkurang seiring bertambahnya jumlah *hidden layer*. Hasil terbaik dari pengujian jumlah *hidden layer* dan jumlah *neuron* dalam *hidden*

layer ditentukan berdasarkan akurasi tertinggi yang diperoleh dari setiap arsitektur yang diuji.

Tabel 3. 6 Nilai *Hyperparameter*

NO	*Hidden layer*	*Neuron* pada *Hidden layer*	*Learning Rate (α)*
1	4	40→38→36→34	0.001
2	6	40→40→38→38→36→36	
3	8	40→40→38→38→36→36→34→34	
4	10	40→40→38→38→36→36→34→34→32→32	
5	12	40→40→40→38→38→38→36→36→36→34→34→34	

3.4 Evaluasi

Perhitungan akurasi *Neural Network* menggunakan *confusion matrix multiclass*. *Confusion matrix* sering digunakan pada masalah klasifikasi sebagai sebuah pengukuran performa di mana *output* dapat terdiri dari dua kelas atau lebih (Putra et al., 2022). Menurut Ringo et al. (2021), *confusion matrix* dibentuk dari masing-masing model untuk membandingkan kinerja setiap model. Melalui *confusion matrix*, dilakukan perhitungan pada masing-masing ukuran evaluasi model berupa *accuracy, precision, recall, dan F1-Measure.* Terdapat empat komponen dalam membentuk ukuran evaluasi model yang merupakan kombinasi dari nilai yang diprediksi dan nilai yang sebenarnya. Keempat komponen tersebut adalah TP (*True Positive*) yang merupakan jumlah data yang bernilai benar pada kategori yang diprediksi maupun yang sebenarnya, TN (*True Negative*) merupakan jumlah data yang bernilai salah pada kategori yang diprediksi maupun yang sebenarnya, FP (*False Positive*) merupakan jumlah data yang bernilai benar pada kategori yang diprediksi namun bernilai salah pada kategori yang sebenarnya, FN (*False Negative*) merupakan jumlah data yang bernilai salah pada kategori yang diprediksi namun bernilai benar pada kategori yang sebenarnya.

Untuk menghitung ukuran evaluasi model, maka digunakan rumus sebagai berikut.

$$Accuracy = \frac{TP}{Total\ Data} \times 100\% \quad (3.1)$$

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (3.2)$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (3.3)$$

$$F1 - Measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \times 100\% \quad (3.4)$$

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan hasil penelitian yang diperoleh dari penerapan metode *Neural Network Backpropagation* dalam klasifikasi kerusakan mesin sepeda motor.

4.1 Pengujian Model

Pengujian model *Neural Network Backpropagation* pada data pengujian bertujuan untuk menentukan model dengan akurasi terbaik dalam mengklasifikasikan kerusakan mesin sepeda motor berdasarkan data pelatihan. Ada lima model yang akan diuji dalam penelitian ini.

4.1.1 Pengujian Model A

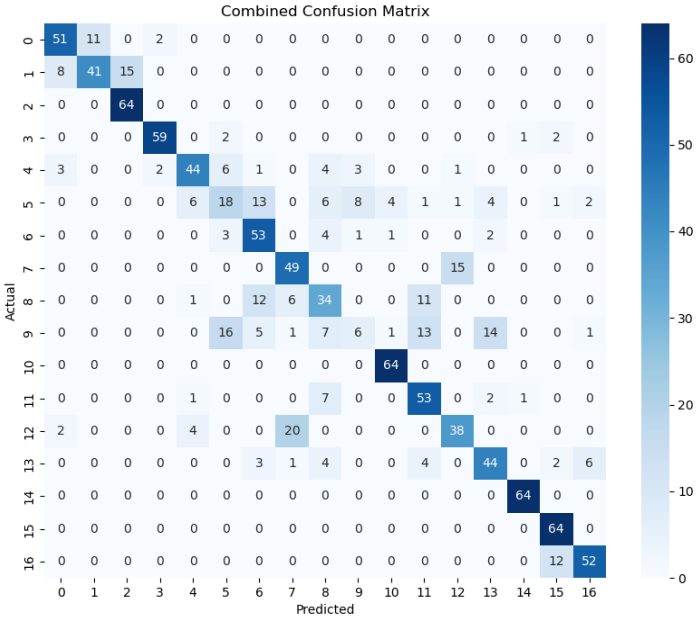
Model A memiliki 4 *hidden layer* dan menggunakan metode *10-Fold Cross-validation* untuk membagi data *training* dan *testing*. Model A juga menguji 3 kombinasi *hyperparameter* yang berbeda.

Tabel 4. 1 Skenario Pengujian Model A

Hyperparameter	Model A		
	A1	A2	A3
Epoch	500	1000	2000
Jumlah k	10		
Hidden layer	40→38→36→34		
Learning rate	0.001		

Hasil pengujian menggunakan model A sebagai berikut.

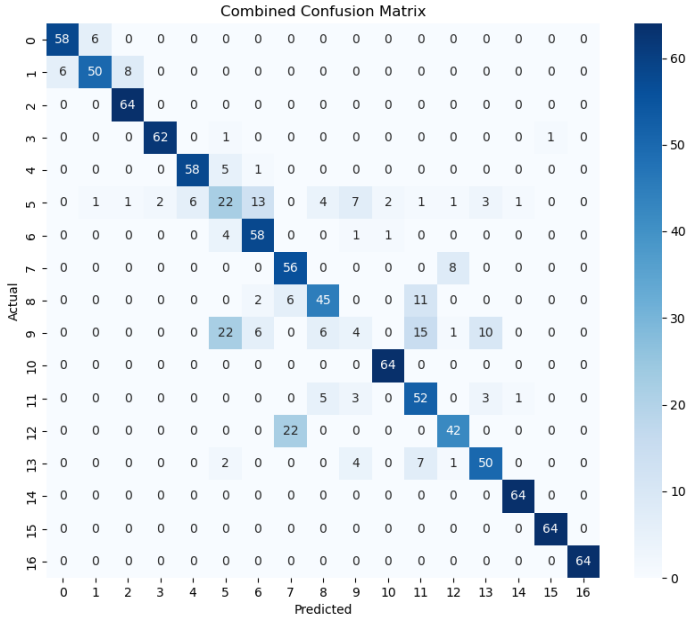
a. Model A1



Gambar 4. 1 *Combined Confusion Matrix* Model A1

Pembagian data di setiap *fold* pada model A1 yaitu sebanyak 979 data *training* dan 109 data *testing*. Pada gambar 4.1 menampilkan hasil *confusion matrix* dari semua *fold* di model A1 dengan *epoch* sebesar 500. Kemudian dilakukan perhitungan akurasi menggunakan persamaan 3.2 sehingga pada model A1 secara keseluruhan menghasilkan rata-rata akurasi sebesar 73.34% dengan rata-rata nilai *loss* sebesar 0.7757.

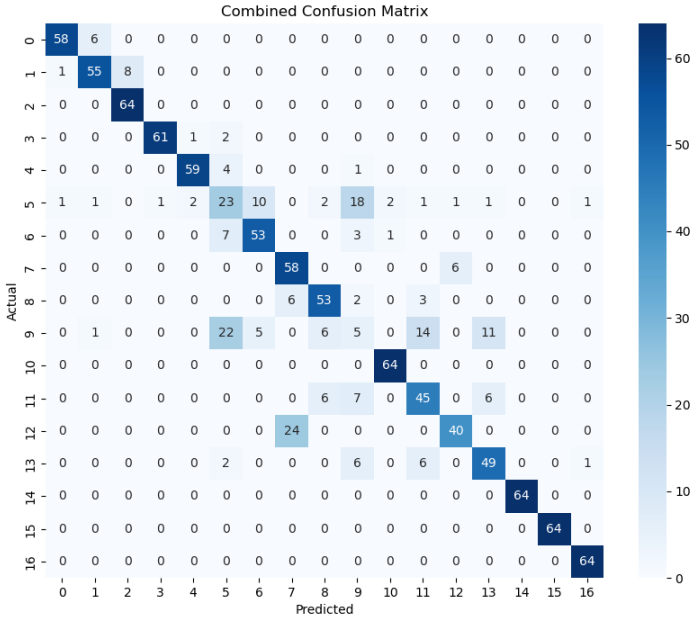
b. Model A2



Gambar 4. 2 *Combined Confusion Matrix* Model A2

Pembagian data di setiap *fold* pada model A2 yaitu sebanyak 979 data *training* dan 109 data *testing*. Pada gambar 4.2 menampilkan hasil *confusion matrix* dari semua *fold* di model A2 dengan *epoch* sebesar 1000. Kemudian dilakukan perhitungan akurasi menggunakan persamaan 3.2 sehingga pada model A2 secara keseluruhan menghasilkan rata-rata akurasi sebesar 80.60% dengan rata-rata nilai *loss* sebesar 0.6078.

c. Model A3



Gambar 4. 3 *Combined Confusion Matrix* Model A3

Pembagian data di setiap *fold* pada model A3 yaitu sebanyak 979 data *training* dan 109 data *testing*. Pada gambar 4.3 menampilkan hasil *confusion matrix* dari semua *fold* di model A3 dengan *epoch* sebesar 2000. Kemudian dilakukan perhitungan akurasi menggunakan persamaan 3.2 sehingga pada model A3 secara keseluruhan menghasilkan rata-rata akurasi sebesar 80.78% dengan rata-rata nilai *loss* sebesar 0.5690.

4.1.2 Pengujian Model B

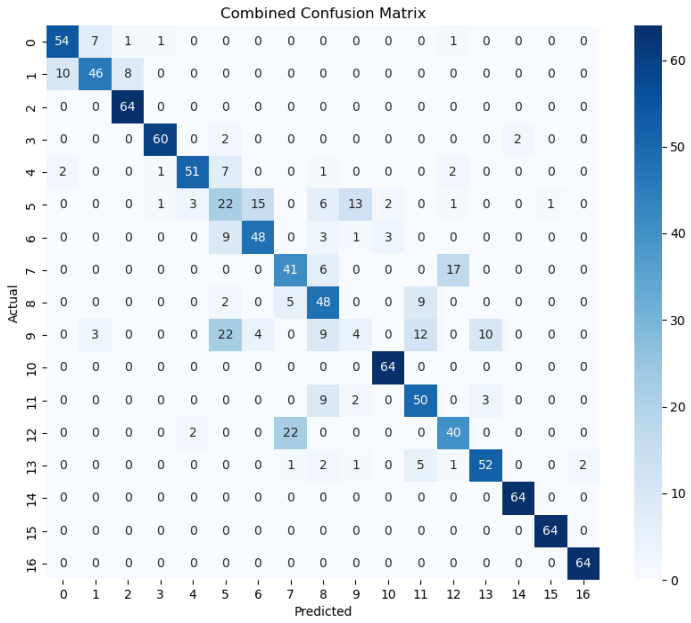
Model B memiliki 6 *hidden layer* dan menggunakan metode *10-Fold Cross-validation* untuk membagi data *training* dan *testing*. Model B juga menguji 3 kombinasi *hyperparameter* yang berbeda.

Tabel 4. 2 Skenario Pengujian Model B

Hyperparameter	Model B		
	B1	B2	B3
Epoch	500	1000	2000
Jumlah k	10		
Hidden layer	40→40→38→38→36→36		
Learning rate	0.001		

Hasil pengujian menggunakan model B sebagai berikut:

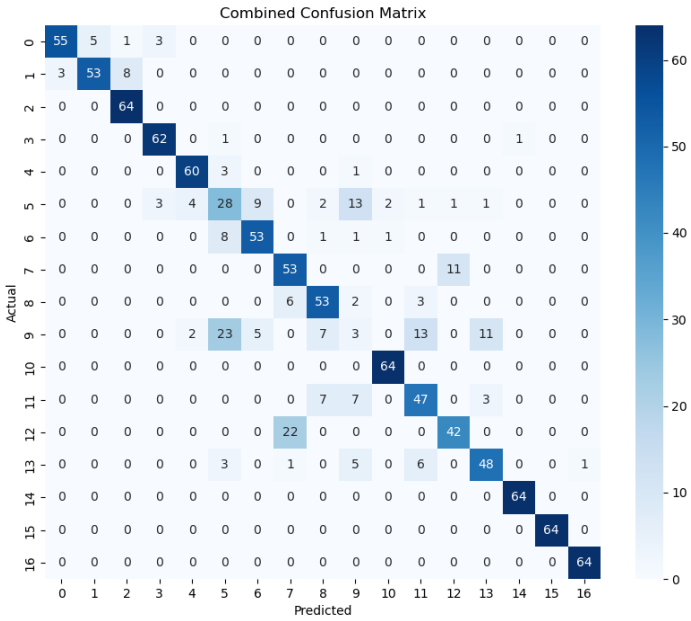
a. Model B1



Gambar 4. 4 *Combined Confusion Matrix* Model B1

Pembagian data di setiap *fold* pada model B1 yaitu sebanyak 979 data *training* dan 109 data *testing*. Pada gambar 4.4 menampilkan hasil *confusion matrix* dari semua *fold* di model B1 dengan *epoch* sebesar 500. Kemudian dilakukan perhitungan akurasi menggunakan persamaan 3.2 sehingga pada model B1 secara keseluruhan menghasilkan rata-rata akurasi sebesar 76.84% dengan rata-rata nilai *loss* sebesar 0.5695.

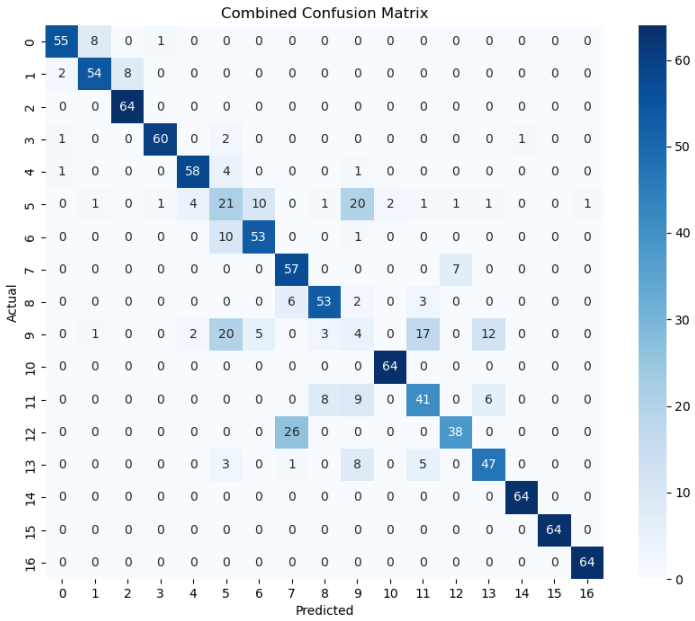
b. Model B2



Gambar 4. 5 *Combined Confusion Matrix* Model B2

Pembagian data di setiap *fold* pada model B2 yaitu sebanyak 979 data *training* dan 109 data *testing*. Pada gambar 4.5 menampilkan hasil *confusion matrix* dari semua *fold* di model B2 dengan *epoch* sebesar 1000. Kemudian dilakukan perhitungan akurasi menggunakan persamaan 3.2 sehingga pada model B2 secara keseluruhan menghasilkan rata-rata akurasi sebesar 80.60% dengan rata-rata nilai *loss* sebesar 0.5962.

c. Model B3



Gambar 4. 6 *Combined Confusion Matrix* Model B3

Pembagian data di setiap *fold* pada model B3 yaitu sebanyak 979 data *training* dan 109 data *testing*. Pada gambar 4.6 menampilkan hasil *confusion matrix* dari semua *fold* di model B3 dengan *epoch* sebesar 2000. Kemudian dilakukan perhitungan akurasi menggunakan persamaan 3.2 sehingga pada model B3 secara keseluruhan menghasilkan rata-rata akurasi sebesar 79.13% dengan rata-rata nilai *loss* sebesar 0.5947.

4.1.3 Pengujian Model C

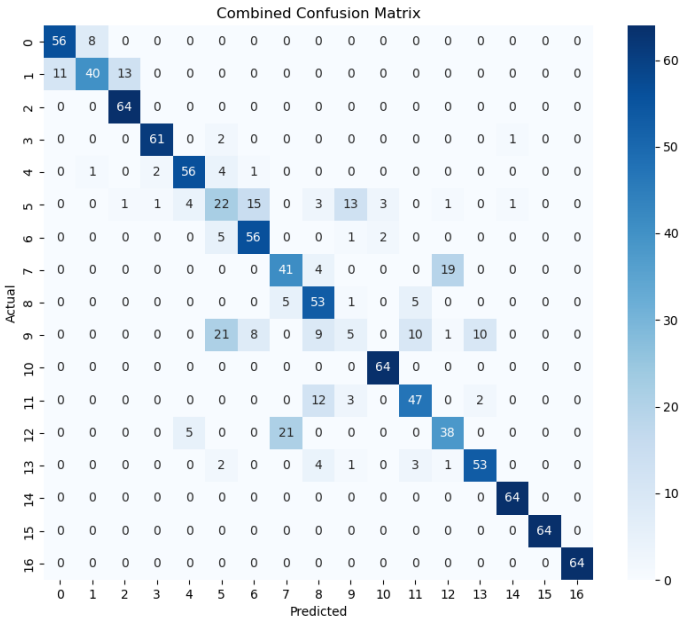
Model C memiliki 8 *hidden layer* dan menggunakan metode *10-Fold Cross-validation* untuk membagi data *training* dan *testing*. Model C juga menguji 3 kombinasi *hyperparameter* yang berbeda.

Tabel 4. 3 Skenario Pengujian Model C

Hyperparameter	Model C		
	C1	C2	C3
Epoch	500	1000	2000
Jumlah k	10		
Hidden layer	40→40→38→38→36→36→34→34		
Learning rate	0.001		

Hasil pengujian menggunakan model C sebagai berikut:

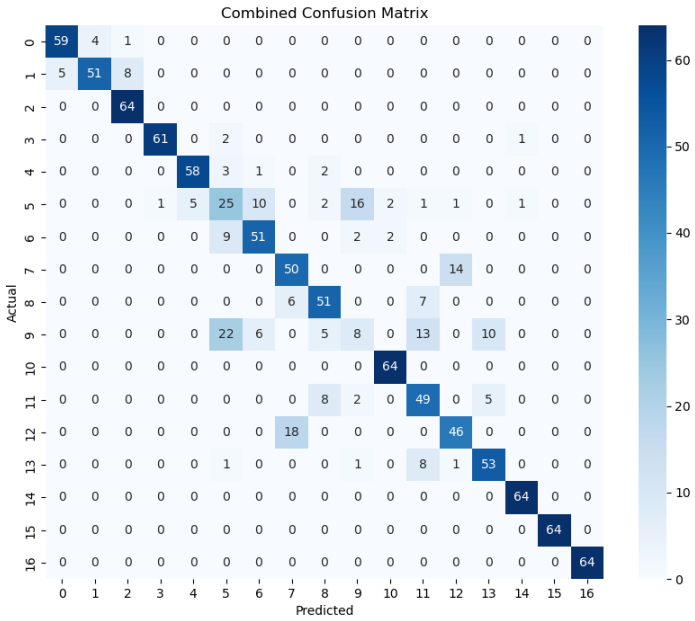
a. Model C1



Gambar 4. 7 *Combined Confusion Matrix* Model C1

Pembagian data di setiap *fold* pada model C1 yaitu sebanyak 979 data *training* dan 109 data *testing*. Pada gambar 4.7 menampilkan hasil *confusion matrix* dari semua *fold* di model C1 dengan *epoch* sebesar 500. Kemudian dilakukan perhitungan akurasi menggunakan persamaan 3.2 sehingga pada model C1 secara keseluruhan menghasilkan rata-rata akurasi sebesar 77.94% dengan rata-rata nilai *loss* sebesar 0.5906.

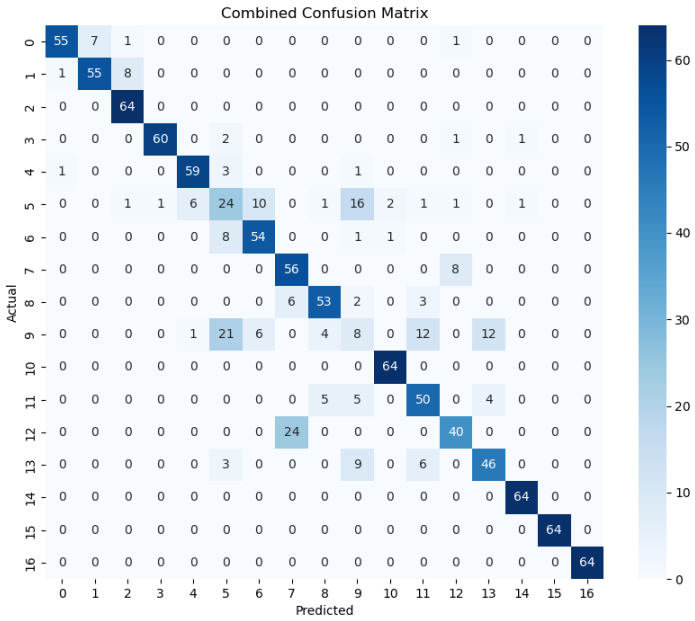
b. Model C2



Gambar 4. 8 *Combined Confusion Matrix* Model C2

Pembagian data di setiap *fold* pada model C2 yaitu sebanyak 979 data *training* dan 109 data *testing*. Pada gambar 4.8 menampilkan hasil *confusion matrix* dari semua *fold* di model C2 dengan *epoch* sebesar 1000. Kemudian dilakukan perhitungan akurasi menggunakan persamaan 3.2 sehingga pada model C2 secara keseluruhan menghasilkan rata-rata akurasi sebesar 81.06% dengan rata-rata nilai *loss* sebesar 0.5382.

c. Model C3



Gambar 4. 9 *Combined Confusion Matrix* Model C3

Pembagian data di setiap *fold* pada model C3 yaitu sebanyak 979 data *training* dan 109 data *testing*. Pada gambar 4.9 menampilkan hasil *confusion matrix* dari semua *fold* di model C3 dengan *epoch* sebesar 2000. Kemudian dilakukan perhitungan akurasi menggunakan persamaan 3.2 sehingga pada model C3 secara keseluruhan menghasilkan rata-rata akurasi sebesar 80.87% dengan rata-rata nilai *loss* sebesar 0.5704.

4.1.4 Pengujian Model D

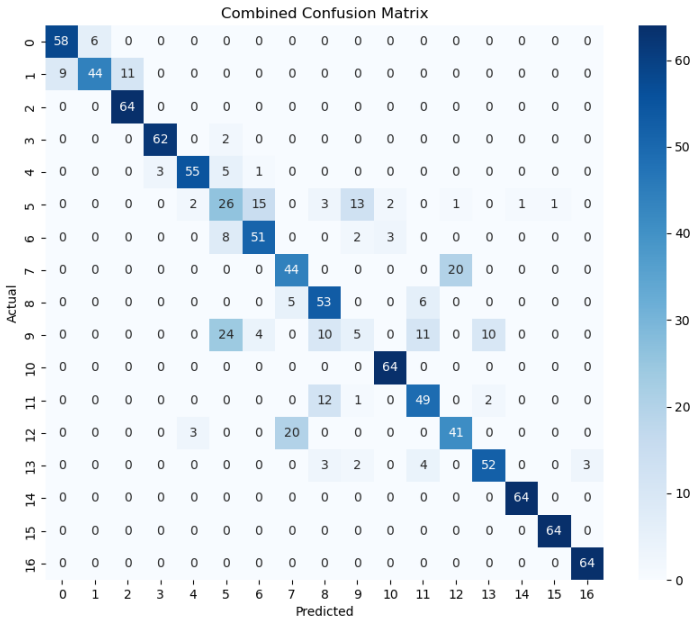
Model D memiliki 10 *hidden layer* dan menggunakan metode *10-Fold Cross-validation* untuk membagi data *training* dan *testing*. Model D juga menguji 3 kombinasi *hyperparameter* yang berbeda.

Tabel 4. 4 Skenario Pengujian Model D

Hyperparameter	Model D		
	D1	D2	D3
Epoch	500	1000	2000
Jumlah k	10		
Hidden layer	40→40→38→38→36→36→34→34→32→32		
Learning rate	0.001		

Hasil pengujian menggunakan model D sebagai berikut:

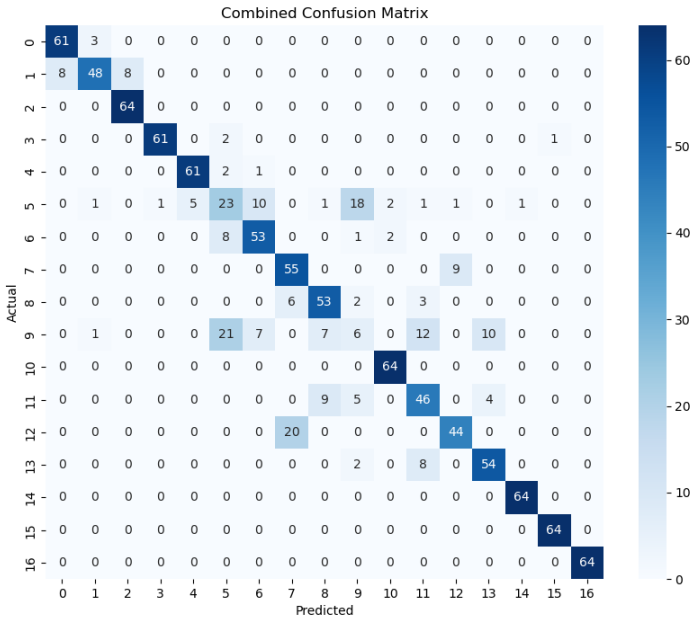
a. Model D1



Gambar 4. 10 *Combined Confusion Matrix* Model D1

Pembagian data di setiap *fold* pada model D1 yaitu sebanyak 979 data *training* dan 109 data *testing*. Pada gambar 4.10 menampilkan hasil *confusion matrix* dari semua *fold* di model D1 dengan *epoch* sebesar 500. Kemudian dilakukan perhitungan akurasi menggunakan persamaan 3.2 sehingga pada model D1 secara keseluruhan menghasilkan rata-rata akurasi sebesar 79.04% dengan rata-rata nilai *loss* sebesar 0.5777.

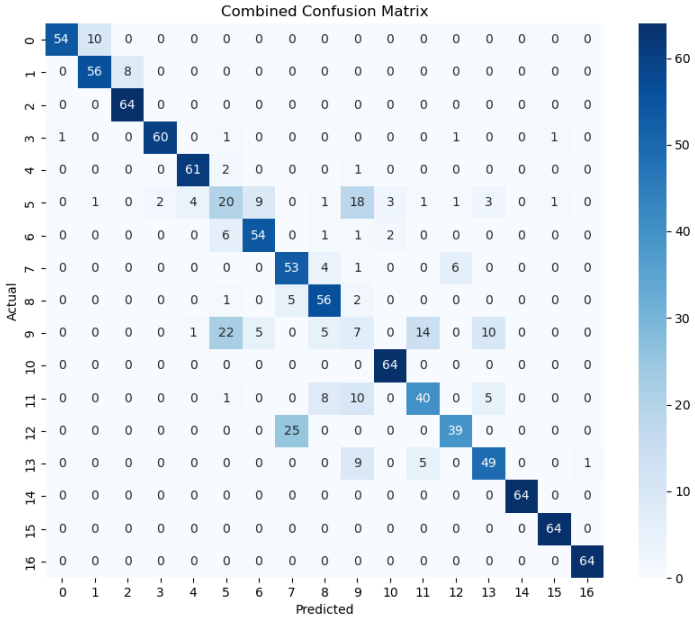
b. Model D2



Gambar 4. 11 *Combined Confusion Matrix* Model D2

Pembagian data di setiap *fold* pada model D2 yaitu sebanyak 979 data *training* dan 109 data *testing*. Pada gambar 4.11 menampilkan hasil *confusion matrix* dari semua *fold* di model D2 dengan *epoch* sebesar 1000. Kemudian dilakukan perhitungan akurasi menggunakan persamaan 3.2 sehingga pada model D2 secara keseluruhan menghasilkan rata-rata akurasi sebesar 81.34% dengan rata-rata nilai *loss* sebesar 0.6395.

c. Model D3



Gambar 4. 12 *Combined Confusion Matrix* Model D3

Pembagian data di setiap *fold* pada model D3 yaitu sebanyak 979 data *training* dan 109 data *testing*. Pada gambar 4.12 menampilkan hasil *confusion matrix* dari semua *fold* di model D3 dengan *epoch* sebesar 2000. Kemudian dilakukan perhitungan akurasi menggunakan persamaan 3.2 sehingga pada model D3 secara keseluruhan menghasilkan rata-rata akurasi sebesar 79.87% dengan rata-rata nilai *loss* sebesar 0.7377.

4.1.5 Pengujian Model E

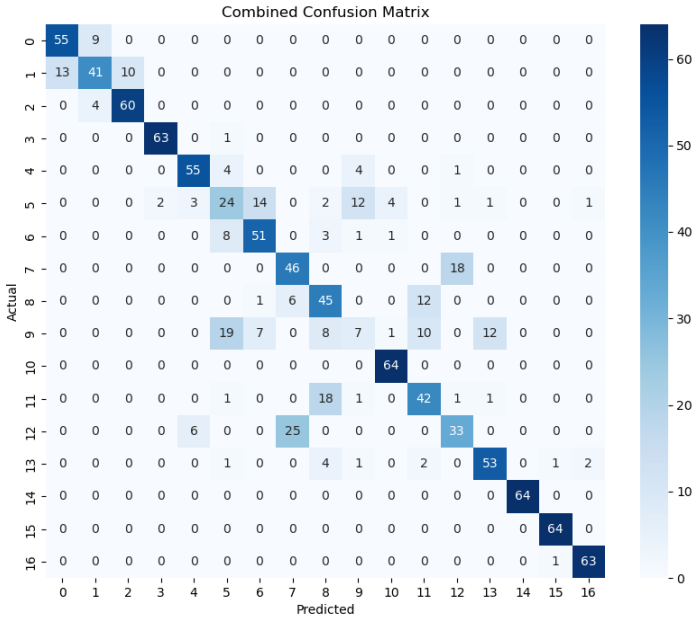
Model E memiliki 12 *hidden layer* dan menggunakan metode *10-Fold Cross-validation* untuk membagi data *training* dan *testing*. Model E juga menguji 3 kombinasi *hyperparameter* yang berbeda.

Tabel 4. 5 Skenario Pengujian Model E

Hyperparameter	Model E		
	E1	E2	E3
Epoch	500	1000	2000
Jumlah k	10		
Hidden layer	40→40→40→38→38→38→36→36→36→34→34→34		
Learning rate	0.001		

Hasil pengujian menggunakan model E sebagai berikut:

a. Model E1

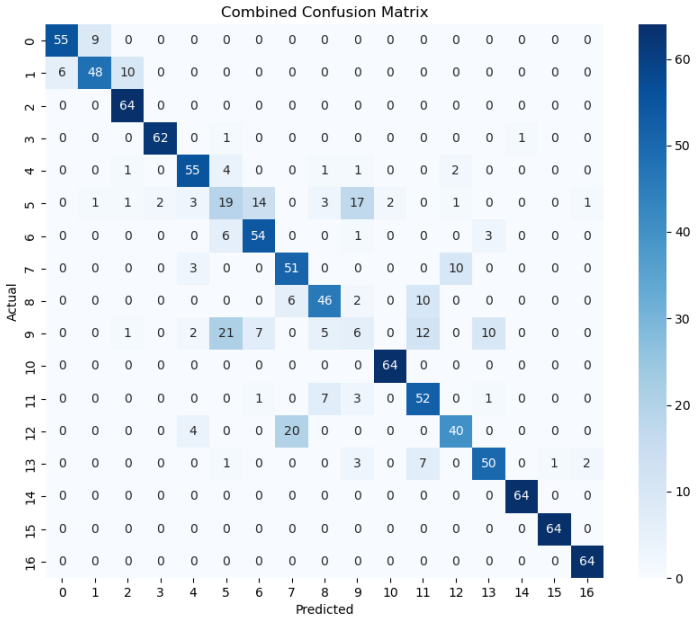


Gambar 4. 13 *Combined Confusion Matrix* Model E1

Pembagian data di setiap *fold* pada model E1 yaitu sebanyak 979 data *training* dan 109 data *testing*. Pada gambar 4.13 menampilkan hasil *confusion matrix* dari semua *fold* di model E1 dengan *epoch* sebesar 500. Kemudian dilakukan perhitungan akurasi menggunakan persamaan 3.2 sehingga pada model E1 secara

keseluruhan menghasilkan rata-rata akurasi sebesar 76.29% dengan rata-rata nilai *loss* sebesar 0.66.

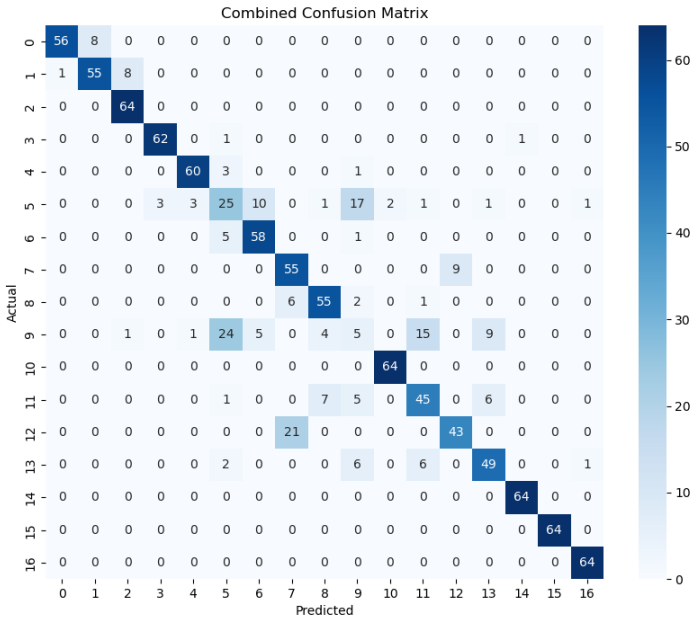
b. Model E2



Gambar 4. 14 *Combined Confusion Matrix* Model E2

Pembagian data di setiap *fold* pada model E2 yaitu sebanyak 979 data *training* dan 109 data *testing*. Pada gambar 4.14 menampilkan hasil *confusion matrix* dari semua *fold* di model E2 dengan *epoch* sebesar 1000. Kemudian dilakukan perhitungan akurasi menggunakan persamaan 3.2 sehingga pada model E2 secara keseluruhan menghasilkan rata-rata akurasi sebesar 78.86% dengan rata-rata nilai *loss* sebesar 0.7015.

c. Model E3



Gambar 4. 15 *Combined Confusion Matrix* Model E3

Pembagian data di setiap *fold* pada model E3 yaitu sebanyak 979 data *training* dan 109 data *testing*. Pada gambar 4.15 menampilkan hasil *confusion matrix* dari semua *fold* di model E3 dengan *epoch* sebesar 2000. Kemudian dilakukan perhitungan akurasi menggunakan persamaan 3.2 sehingga pada model E3 secara keseluruhan menghasilkan rata-rata akurasi sebesar 81.61% dengan rata-rata nilai *loss* sebesar 0.5973.

4.2 Visualisasi Hasil Uji Coba

Berdasarkan uji coba yang sudah dilakukan, berikut adalah sampel yang diambil berdasarkan akurasi tertinggi pada hasil uji coba dari seluruh model.

Tabel 4. 6 Visualisasi Hasil Uji Coba

Model	*Hidden layer*	Epoch	Rata-rata akurasi	Rata-rata *loss*	Standar deviasi akurasi	Standar deviasi *loss*
A1	4	500	73.34%	0.7757	4.61%	0.0937
B1	6		76.84%	0.6101	2.58%	0.1172
C1	8		77.94%	0.5622	3.99%	0.1024
D1	10		79.04%	0.5637	4.06%	0.1513
E1	12		76.29%	0.5973	3.82%	0.1236
A2	4	1000	80.60%	0.5886	4.42%	0.1758
B2	6		80.60%	0.5870	3.52%	0.1803
C2	8		81.06%	0.5382	4.09%	0.1480
D2	10		81.34%	0.5602	4.79%	0.1788
E2	12		78.86%	0.5732	5.34%	0.1620
A3	4	2000	80.78%	0.6530	4.83%	0.2766
B3	6		79.13%	0.7320	4.41%	0.2717
C3	8		80.87%	0.6660	5.02%	0.2026
D3	10		79.87%	0.7209	4.67%	0.2248
E3	12		81.61%	0.5973	3.75%	0.2107

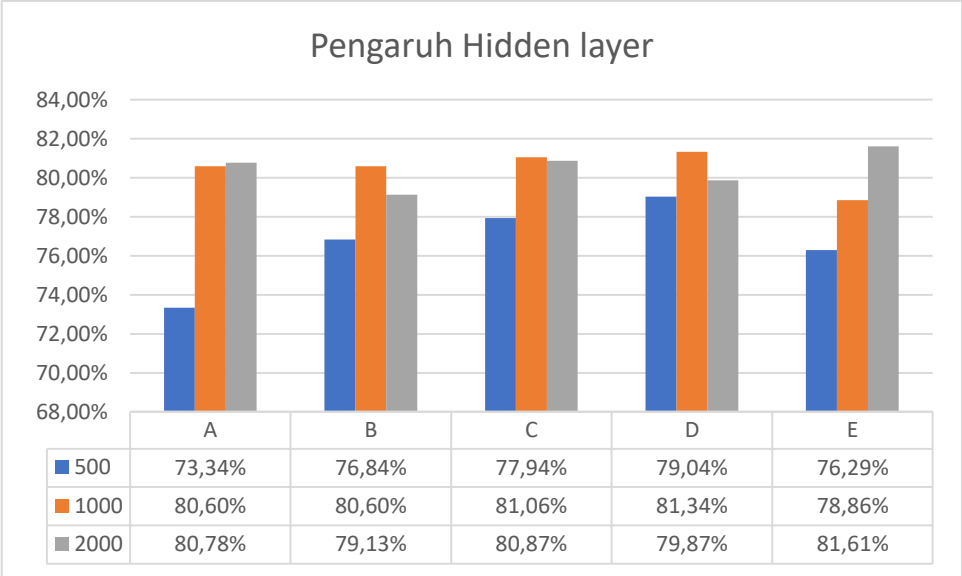
Berdasarkan tabel 4.6, terdapat 15 kombinasi yang telah diuji, dan didapatkan nilai akurasi tertinggi pada model E3 dengan kombinasi 12 *hidden layer*, *epoch* sebesar 2000 menghasilkan akurasi sebesar 81.61%. Berdasarkan hasil percobaan yang telah dilakukan pada seluruh model pada tabel 4.7 dengan setiap kombinasi *hyperparameter*-nya, berikut merupakan pengaruh setiap *hyperparameter* yang di*tuning* pada penelitian ini.

4.2.1 Pengaruh *Hidden layer*

Pada penelitian ini, uji coba dilakukan dengan 5 kombinasi *hidden layer* yang mengacu pada penelitian sebelumnya yang dilakukan oleh (Kurniawan et al., 2023). Pada model A menggunakan 4 *hidden layer*, model B menggunakan 6 *hidden layer*, model C menggunakan 8 *hidden layer*, model D menggunakan 10 *hidden layer*, dan model E menggunakan 12 *hidden layer*. *Hidden layer* terbaik ditentukan dengan cara membandingkan rata-rata akurasi di setiap model *hidden layer*.

Tabel 4. 7 Visualisasi Pengaruh *Hidden layer* Terhadap Akurasi

Perbandingan *Hidden layer*									
4		**6**		**8**		**10**		**12**	
A1	73.34%	**B1**	76.84%	C1	77.94%	**D1**	79.04%	**E1**	76.29%
A2	80.60%	**B2**	80.60%	C2	81.06%	**D2**	81.34%	**E2**	78.86%
A3	80.78%	**B3**	79.13%	C3	80.87%	**D3**	79.87%	**E3**	81.61%



Gambar 4. 16 Visualisasi Perbandingan *Hidden layer*

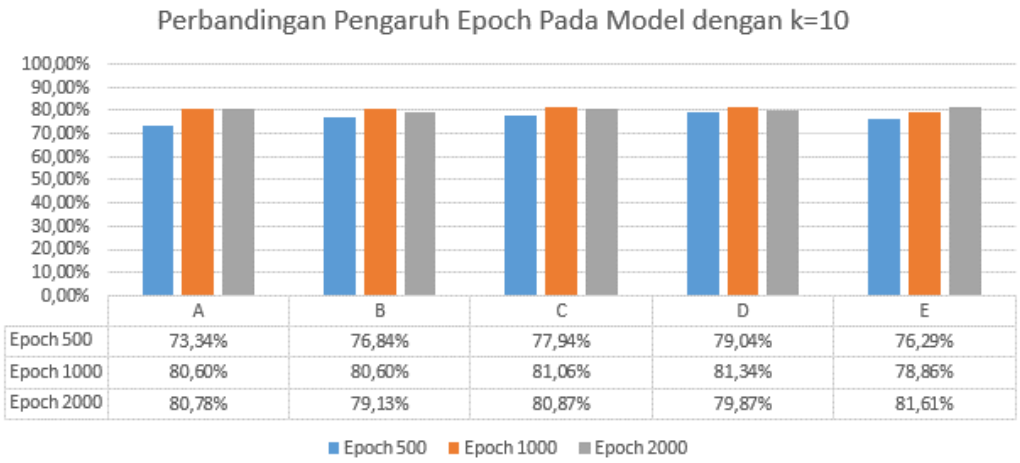
Berdasarkan visualisasi perbandingan *hidden layer* pada tabel 4.7 dan gambar 4.16, model dengan *hidden layer* sebanyak 8, 10, dan 12 yang mencapai akurasi tertinggi di atas 81% di beberapa model, misalnya pada model E3 dengan 12 *hidden layer* yang mencapai 81,61%. Di sisi lain, *hidden layer* 4 menghasilkan akurasi lebih rendah jika dibandingkan dengan jumlah *hidden layer* lainnya. Hal ini menandakan peningkatan jumlah *hidden layer* memengaruhi akurasi model. Jumlah *hidden layer* yang terlalu sedikit cenderung tidak cukup menangkap kompleksitas data, sedangkan jumlah yang lebih banyak, seperti 10 dan 12 *hidden layer*, menunjukkan hasil yang lebih optimal untuk menangani pola data yang lebih kompleks.

4.2.2 Pengaruh *Epoch*

Dalam penelitian ini, pengaruh *epoch* berpengaruh terhadap kinerja model. *Epoch* memberikan kesempatan bagi model untuk mempelajari data secara berulang kali. Setelah melalui satu *epoch*, sebuah model mungkin belum cukup belajar dari data untuk membuat prediksi yang akurat. Seiring bertambahnya *epoch*, model mempunyai kemungkinan untuk semakin baik dalam menangkap pola-pola penting dalam data. Dengan demikian, suatu model mungkin memiliki kinerja yang lebih baik pada data pelatihan apabila dilatih dengan lebih banyak *epoch*. Untuk mengetahui *epoch* yang optimal pada penelitian ini dilakukan dengan membandingkan pengaruh *epoch* terhadap akurasi sebagai berikut.

Tabel 4. 8 Perbandingan Pengaruh *Epoch*

Perbandingan pengaruh epoch terhadap akurasi					
500		**1000**		**2000**	
A1	73.34%	**A2**	80.60%	**A3**	80.78%
B1	76.84%	**B2**	80.60%	**B3**	79.13%
C1	77.94%	**C2**	81.06%	**C3**	80.87%
D1	79.04%	**D2**	81.34%	**D3**	79.87%
E1	76.29%	**E2**	78.86%	**E3**	81.61%



Gambar 4. 17 Perbandingan Pengaruh *Epoch*

Berdasarkan tabel 4.8 dan gambar 4.17, pada *epoch* 500 akurasi model juga masih relatif rendah. Ketika *epoch* bertambah menjadi 1000, akurasi meningkat cukup banyak, terutama untuk model C yang mencatatkan akurasi sebesar 81.06% dan D sebesar 81.34%. Pada *epoch* 2000, meskipun akurasi sedikit menurun di beberapa model, hasilnya tetap tinggi, dengan Model E mencatatkan akurasi tertinggi yaitu 81.61%. Secara keseluruhan, penggunaan *epoch* sebesar 1000 dan 2000 memberikan hasil yang lebih baik.

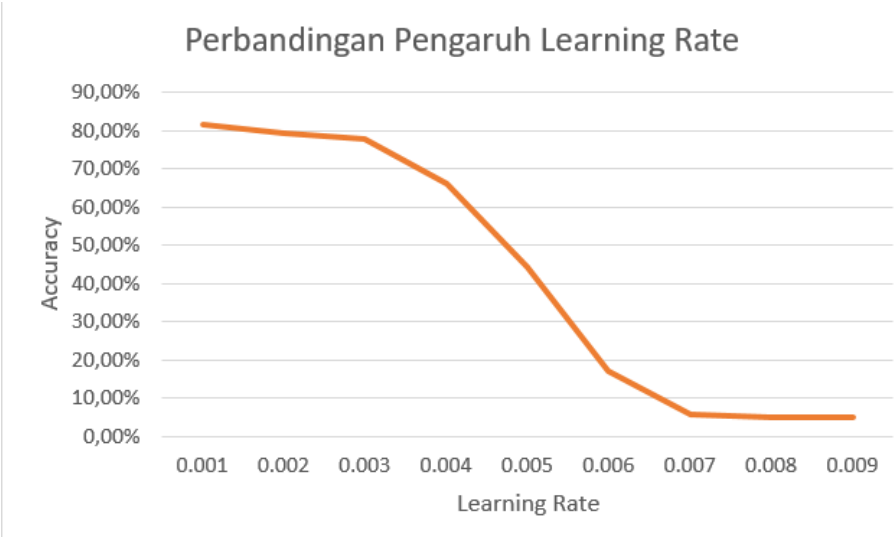
Secara umum, *epoch* sebesar 1000 dan 2000 memberikan peningkatan yang signifikan pada akurasi dibandingkan dengan *epoch* 500, dan meskipun ada sedikit peningkatan pada *epoch* 2000, hasilnya mulai stabil dengan sedikit penurunan atau tidak ada perubahan signifikan. Oleh karena itu, penggunaan *epoch* 1000 dan 2000 dapat dianggap optimal untuk mendapatkan akurasi yang tinggi dan stabil.

4.2.3 Pengaruh *Learning Rate*

Dalam penelitian ini, *learning rate* sebesar 0.001 digunakan pada setiap pengujian model. Nilai ini dipilih karena memberikan tingkat akurasi terbaik setelah dibandingkan dengan beberapa nilai *learning rate* lainnya, dengan model E yang memiliki *epoch* 2000 menghasilkan akurasi tertinggi saat diuji menggunakan *learning rate* 0.001.

Tabel 4. 9 Pengujian *Learning Rate* pada Model E dengan *Epoch* 2000

Learning Rate	***Accuracy***
0.001	81.61%
0.002	79.32%
0.003	77.75%
0.004	66.26%
0.005	44.29%
0.006	17.13%
0.007	5.61%
0.008	5.15%
0.009	5.15%



Gambar 4. 18 Grafik Perbandingan *Learning Rate* 0.001-0.009

Berdasarkan tabel 4.9 dan gambar 4.18, dapat disimpulkan bahwa *learning rate* 0.001 menghasilkan akurasi tertinggi dibandingkan dengan nilai *learning rate* lainnya.

Berdasarkan hasil ini, jumlah *epoch* yang optimal pada penelitian ini adalah sebesar 1000 dan 2000 untuk mencapai performa terbaik. Hasil ini juga sesuai dengan penelitian sebelumnya yang dilakukan oleh (Kaur & Singh, 2016) yang menyatakan bahwa menggunakan *epoch* sebesar 1000 dan 2000 menghasilkan akurasi yang lebih baik daripada *epoch* 500.

Dalam Surah *al-An'am* ayat 142 yang berbunyi:

وَمِنَ الْاَنْعَامِ حَمُوْلَةً وَّفَرْشًاۗ كُلُوْا مِمَّا رَزَقَكُمُ اللّٰهُ وَلَا تَتَّبِعُوْا خُطُوٰتِ الشَّيْطٰنِۗ اِنَّهٗ لَكُمْ عَدُوٌّ مُّبِيْنٌۙ

"Di antara hewan-hewan ternak itu ada yang dijadikan pengangkut beban dan ada (pula) yang untuk disembelih. Makanlah rezeki yang diberikan Allah kepadamu. Janganlah kamu mengikuti langkah-langkah setan. Sesungguhnya dia adalah musuh yang nyata bagimu." (Q.S. Al-An'am: 142)

Ayat tersebut mengingatkan manusia untuk memanfaatkan nikmat Allah dengan bijak, termasuk kendaraan yang digunakan sesuai kapasitas dan fungsinya. Ayat ini juga mengajarkan agar manusia tidak mengikuti langkah setan, yang dalam konteks penggunaan kendaraan bisa berarti penggunaan yang berlebihan atau tidak sesuai sehingga menimbulkan kerusakan lebih lanjut.

Selain itu, pada surah Al-Hujurat ayat 13 yang berbunyi:

يٰٓاَيُّهَا النَّاسُ اِنَّا خَلَقْنٰكُمْ مِّنْ ذَكَرٍ وَّاُنْثٰى وَجَعَلْنٰكُمْ شُعُوْبًا وَّقَبَاۤىِٕلَ لِتَعَارَفُوْاۚ اِنَّ اَكْرَمَكُمْ عِنْدَ اللّٰهِ اَتْقٰىكُمْۗ 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 Mahateliti." (Q.S. Al-Hujurat: 13)

Ayat tersebut mengajarkan pentingnya memahami keragaman dalam kehidupan manusia, yang juga dapat dianalogikan dengan keragaman jenis kerusakan pada mesin motor. Sama seperti manusia yang berbangsa dan bersuku-suku, setiap jenis kerusakan mesin memiliki ciri dan penyebab yang berbeda. Misalnya, kerusakan pada sistem kelistrikan berbeda dengan masalah pada sistem bahan bakar atau sistem pendingin, dan masing-masing memerlukan penanganan yang spesifik.

Sistem prediksi digunakan untuk mengidentifikasi kerusakan mesin berdasarkan variable *input* seperti keluhan atau gejala yang dialami, tipe motor, usia motor, dan jumlah kilometer yang sudah ditempuh, yang kemudian dijadikan acuan untuk perbaikan selanjutnya. Dengan menggunakan metode *Neural Network Backpropagation*, prediksi kerusakan dapat menjadi lebih akurat dan membantu

mekanik dalam memahami pola kerusakan tersebut dengan lebih mendalam. Teknologi ini memungkinkan sistem belajar dari data keluhan dan kerusakan yang sebelumnya terjadi, sehingga memudahkan diagnosa secara lebih tepat serta mengurangi risiko perbaikan yang tidak tepat sasaran. Hal ini memastikan bahwa setiap kerusakan mesin diperlakukan secara adil, dengan penanganan yang sesuai.

Melalui sistem klasifikasi dan prediksi ini, pemilik sepeda motor dapat mengetahui kemungkinan kerusakan yang terjadi pada mesin sepeda motornya sehingga bisa mengantisipasi kerusakan yang lebih parah. Sedangkan bagi mekanik dapat menangani setiap jenis kerusakan dengan pemahaman yang lebih baik dan metode yang tepat, sehingga perbaikan lebih efisien dan performa mesin tetap terjaga dengan optimal.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Peningkatan jumlah *hidden layer* secara umum meningkatkan akurasi model, dengan *hidden layer* yang lebih banyak memungkinkan model menangkap pola data yang lebih kompleks. Dalam penelitian ini, kombinasi model E dengan 12 *hidden layer* menggunakan *epoch* sebesar 2000 mencapai akurasi tertinggi sebesar 81.61%. *Hidden layer* yang terlalu sedikit seperti model A menghasilkan akurasi yang lebih rendah, yang menunjukkan keterbatasan dalam mempelajari data yang kompleks.

Jumlah *epoch* memengaruhi kemampuan model untuk belajar dari data. Pada *epoch* 500, akurasi masih cukup rendah, tetapi peningkatan signifikan terlihat pada *epoch* 1000. Sementara itu, perbedaan akurasi antara *epoch* 1000 dan 2000 cenderung kecil, menunjukkan bahwa model mulai mencapai stabilitas. Secara keseluruhan, penggunaan *epoch* 1000 dan 2000 memberikan hasil optimal, dengan 2000 menghasilkan akurasi tertinggi pada konfigurasi tertentu.

Pemilihan nilai *learning rate* juga sangat memengaruhi performa model. Pada nilai *learning rate* 0.001, akurasi mencapai angka tertinggi sebesar 81.61%, menunjukkan bahwa model dapat belajar secara optimal tanpa melampaui batas stabilitas pembelajaran. Namun, ketika nilai *learning rate* meningkat, akurasi mulai menurun hingga mencapai 5.15%. Penurunan ini menunjukkan bahwa model tidak

lagi mampu menyesuaikan parameter dengan baik, sehingga proses pembelajaran menjadi kurang efektif

Dari hasil penelitian, pengaruh *hidden layer*, *epoch*, dan *learning rate* terhadap akurasi model *Neural Network* saling berhubungan. Penambahan *hidden layer* meningkatkan kemampuan model menangkap pola data kompleks, dengan 12 *hidden layer* memberikan hasil terbaik. Penggunaan *epoch* sebesar 1000 dan 2000 memungkinkan model mencapai akurasi optimal dengan peningkatan performa signifikan dari *epoch* 500. Penggunaan *learning rate* 0.001 lebih direkomendasikan karena memberikan hasil akurasi yang lebih baik dan stabil. Pemilihan nilai *learning rate* yang optimal sangat penting untuk menjaga keseimbangan antara kecepatan konvergensi dan akurasi model. Kombinasi optimal dari parameter ini menunjukkan bahwa *tuning hyperparameter* yang tepat sangat penting dalam meningkatkan performa model.

Metode *Neural Network* Backpropagation terbukti efektif dalam mengklasifikasikan jenis kerusakan mesin motor dengan rata-rata akurasi yang cukup tinggi pada data yang telah diolah. Metode ini memungkinkan deteksi pola kerusakan berdasarkan gejala, tipe motor, usia motor, dan jarak tempuh. Penerapan metode ini diharapkan dapat membantu bengkel dan pengguna motor untuk lebih cepat mendeteksi dan menangani kerusakan, sehingga meningkatkan efisiensi perawatan kendaraan.

5.2 Saran

Hasil penelitian ini masih memiliki beberapa kelemahan. Oleh karena itu, penelitian selanjutnya diharapkan dapat meningkatkan tingkat akurasi pada

pengujian model. Berikut adalah beberapa saran dari peneliti untuk mendukung penelitian yang akan datang:

1. Penelitian selanjutnya disarankan untuk memperhatikan aspek *heterogenitas* dan *homogenitas* komposisi data guna memastikan distribusi data yang lebih seimbang dan representatif, sehingga model dapat menghasilkan performa yang lebih akurat dan generalisasi yang baik pada berbagai kondisi.
2. Disarankan untuk menerapkan *threshold error* sebagai batas toleransi kesalahan dalam evaluasi model. Hal ini bertujuan untuk mengidentifikasi hasil prediksi yang mendekati ambang batas keputusan, sehingga dapat digunakan untuk analisis lebih lanjut dalam pengambilan keputusan atau pengembangan sistem.

DAFTAR PUSTAKA

Akhsa, A. C. D. (2024). Sistem Pakar Diagnosa Kerusakan Motor Matic Menggunakan Metode Forward Chaining. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, *4*(2), 452–462. https://doi.org/10.57152/malcom.v4i2.1189

Almais, A. T. W., Crysdian, C., Holle, K. F. H., & Roihan, A. (2022). Smart Assessment Menggunakan Backpropagation *Neural Network* Smart Assessment using Backpropagation *Neural Network*. *MATRIK : Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, *21*(3), 525–538. https://doi.org/10.30812/matrik.v21i3.1382

Dedi Saputra, Deasy Purwaningtias, W. I. (2018). Sistem Pakar Diagnosa Kerusakan Mesin Sepeda Motor Matic Berbasis Web Menggunakan Certainty Factor. *Indonesian Journal on Networking and Security*, *8*(2), 63–70. https://ijns.org/journal/index.php/ijns/article/download/1555/1507

Epriliani, L., Mayadi, & Pamungkas, R. W. P. (2022). Implementasi Algoritma Naïve Bayes Untuk Memprediksi Kerusakan Sepeda Motor Pada Bengkel Citra Djaya Motor. *Journal of Informatic and Information Security*, *3*(1), 59–72. https://doi.org/10.31599/jiforty.v3i1.1268

Faisal, A., Yunial, A. H., Komputer, I., Informatika, T., Pamulang, U., & Selatan, T. (2024). Sistem Pakar Diagnosa Kerusakan Pada Vespa Matic Menggunakan Metode Naïve Bayes Berbasis Web (Studi kasus : PT . Dwi Pratama Mandiri). *Biner : Jurnal Ilmu Komputer, Teknik Dan Multimedia*, *1*(6), 1409–1424. https://journal.mediapublikasi.id/index.php/Biner/article/view/4099

Ghaly, M. S., & Winoko, Y. A. (2019). Analisis Perubahan Diameter Base Circle Camshaft Terhadap Daya Dan Torsi Pada Sepeda Motor. *Jurnal Flywheel*, *10*(2), 7–12. https://doi.org/https://doi.org/10.36040/flywheel.v10i2.742

Hecht-Nielsen, R. (1989). Theory of The Backpropagation *Neural Network*. *International 1989 Joint Conference on Neural Networks*, *1*, 593–605. https://doi.org/10.1109/ijcnn.1989.118638

Herdian, C., Kamila, A., & Agung Musa Budidarma, I. G. (2024). Studi Kasus Feature Engineering Untuk Data Teks: Perbandingan Label Encoding dan One-Hot Encoding Pada Metode Linear Regresi. *Technologia : Jurnal Ilmiah*, *15*(1), 93. https://doi.org/10.31602/tji.v15i1.13457

Hizham, F. A., Nurdiansyah, Y., & Firmansyah, D. M. (2018). Implementasi Metode Backpropagation *Neural Network* (BNN) dalam Sistem Klasifikasi Ketepatan Waktu Kelulusan Mahasiswa (Studi Kasus: Program Studi Sistem Informasi Universitas Jember). *Berkala Sainstek*, *6*(2), 97–105. https://doi.org/10.19184/bst.v6i2.9254

Kaur, J., & Singh, S. (2016). *Neural Network* based refactoring area identification in Software System with object oriented metrics. *Indian Journal of Science and Technology*, *9*(10), 1–8. https://doi.org/10.17485/ijst/2016/v9i10/85110

Kurniawan, P. M., Almais, A. T. W., Hariyadi, M. A., Yaqin, M. A., & Suhartono. (2023). Prediction of Civil Servant Performance Allowances Using the *Neural Network* Backpropagation Method. *International Journal on Informatics Visualization*, *7*(3), 673–680. https://doi.org/10.30630/joiv.7.3.1698

Malakouti, S. M., Menhaj, M. B., & Suratgar, A. A. (2023). The usage of 10-fold cross-validation and grid search to enhance ML methods performance in solar farm power generation prediction. *Cleaner Engineering and Technology*, *15*(February), 100664. https://doi.org/10.1016/j.clet.2023.100664

Marthasari, G. I., Astiti, S. A., & Azhar, Y. (2021). Prediksi Data Time-series menggunakan Jaringan Syaraf Tiruan Algoritma Backpropagation Pada Kasus Prediksi Permintaan Beras. *Jurnal Informatika: Jurnal Pengembangan IT*, *6*(3), 187–193. https://doi.org/10.30591/jpit.v6i3.2627

Muriyatmoko, D., Taufiqurrahman, T., & Pratama, D. B. (2021). Aplikasi Diagnosa Kerusakan Sepeda Motor Menggunakan Metode Naïve Bayes dan Rekomendasi Kesesuaian Bengkel. *Seminar Nasional Informatika Dan Aplikasinya*, *5*, 1–6. https://snia.unjani.ac.id/web/index.php/snia/article/view/233

Purnama, S., Firdausy, K., & Yudhana, A. (2007). Sistem Pakar Pendeteksi Kerusakan Mesin Motor Menggunakan Borland Delphi 7. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *5*(1), 33. https://doi.org/10.12928/telkomnika.v5i1.1333

Putra, T. I. Z. M., Suprapto, S., & Bukhori, A. F. (2022). Model Klasifikasi Berbasis Multiclass Classification dengan Kombinasi Indobert Embedding dan Long Short-Term Memory untuk Tweet Berbahasa Indonesia. In *Jurnal Ilmu Siber dan Teknologi Digital* (Vol. 1, Issue 1, pp. 1–28). https://doi.org/10.35912/jisted.v1i1.1509

Ringo, J. N. B. S., Mursalin, W. J., Nurfadilah, N. C., Ramadhan, D. R., & Madjida, W. O. Z. (2021). Perbandingan Metode Klasifikasi Multiclass untuk Pemetaan Zona Risiko COVID-19 di Pulau Jawa. *Jurnal Komputer Dan Informatika*, *9*(1), 98–107. https://doi.org/10.35508/jicon.v9i1.3602

Rohmawati, F., Rohman, M. G., & Mujilahwati, S. (2017). Sistem Prediksi Jumlah Pengunjung Wisata Wego Kec.Sugio Kab.Lamongan Menggunakan Metode Fuzzy Time Series. *Jouticla*, *2*(2), 67–74. https://doi.org/10.30736/jti.v2i2.66

Romahadi, D., Feriyanto, D., Suprihatiningsih, W., & Setiawan, W. N. (2022). Perancangan Sistem Diagnosis Getaran Motor Menggunakan Jaringan Saraf Tiruan Propagasi Mundur. *Jurnal Rekayasa Mesin*, *13*(1), 37–46. https://doi.org/10.21776/ub.jrm.2022.013.01.5

Sasongko, T. B. (2024). *Optimasi Performa Random Forest dengan Random Oversampling dan SMOTE pada Dataset Diabetes*. *8*, 1756–1767. https://doi.org/10.30865/mib.v8i3.7855

Siang, J. J. (2009). Jaringan Syaraf Tiruan dan Pemrogramannya menggunakan MATLAB. In *Book* (2nd ed., Vol. 2). CV Andi Offset.

Suhartanto, R. S., Dewi, C., & Muflikhah, L. (2017). Implementasi Jaringan Syaraf Tiruan Backpropagation untuk Mendiagnosis Penyakit Kulit pada Anak. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *1*(7), 555–562. http://j-ptiik.ub.ac.id

Syaharuddin, Pujiana, E., Sari, I. P., Mardika, V. M., & Putri, M. (2019). Analisis Algoritma Back Propagation Dalam Prediksi Angka Kemiskinan Di Indonesia. *CESS (Journal of Computer Engineering System and Science)*, *4*(2), 198–207. https://doi.org/https://doi.org/10.31764/pendekar.v3i1.2814

Syofian, S., & Nugraha, A. (2021). Prediksi Sistem Stok Barang Toko Elektronik Abc Dengan Algoritma Apriori Dan Metode Moving Average. *Jurnal Sains Dan Teknologi*, *11*(1), 27–32. https://unsada.e-journal.id/jst/issue/view/31/12