

**PENGEMBANGAN APLIKASI *MOBILE* OCR UNTUK PEMBACAAN
METERAN AIR PADA PAMDES PANDANREJO MENGGUNAKAN
SEGMENTASI DIGIT DAN MOBILENETV2**

SKRIPSI

Oleh :
MOCH MINANUR RAHMAN
NIM. 220605110112



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2026**

**PENGEMBANGAN APLIKASI *MOBILE* OCR UNTUK PEMBACAAN
METERAN AIR PADA PAMDES PANDANREJO MENGGUNAKAN
SEGMENTASI DIGIT DAN MOBILENETV2**

SKRIPSI

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

Oleh :
MOCH MINANUR RAHMAN
NIM. 220605110112

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2026**

HALAMAN PERSETUJUAN

PENGEMBANGAN APLIKASI *MOBILE* OCR UNTUK PEMBACAAN METERAN AIR PADA PAMDES PANDANREJO MENGGUNAKAN SEGMENTASI DIGIT DAN MOBILENETV2

SKRIPSI

Oleh :
MOCH MINANUR RAHMAN
NIM. 220605110112

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 12 Juni 2026

Pembimbing I,



Nur Fitriyah Ayu Tunjung Sari, M.Cs
NIP. 19911226 202012 2 001

Pembimbing II,



Nurizal Dwi Priandani, M.Kom
NIP. 19920830 202203 1 001

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



Supriyono, M. Kom
NIP. 19841010 201903 1 012

HALAMAN PENGESAHAN

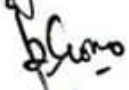
PENGEMBANGAN APLIKASI *MOBILE* OCR UNTUK PEMBACAAN METERAN AIR PADA PAMDES PANDANREJO MENGGUNAKAN SEGMENTASI DIGIT DAN MOBILENETV2

SKRIPSI

Oleh :
MOCH MINANUR RAHMAN
NIM. 220605110112

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 23 Juni 2026

Susunan Dewan Penguji

Ketua Penguji	: <u>A'la Syauqi, M.Kom</u> NIP. 19771201 200801 1 007	()
Anggota Penguji I	: <u>Roro Inda Melani, M.T, M.Sc</u> NIP. 19780925 200501 2 008	()
Anggota Penguji II	: <u>Nur Fitriyah Ayu Tunjung Sari, M.Cs</u> NIP. 19911226 202012 2 001	()
Anggota Penguji III	: <u>Nurizal Dwi Priandani, M.Kom</u> NIP. 19920830 202203 1 001	()

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



Supriyono, M.Kom
NIP. 19841010 201903 1 012

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : MOCH MINANUR RAHMAN
NIM : 220605110112
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika
Judul Skripsi : Pengembangan Aplikasi *Mobile* OCR untuk
Pembacaan Meteran Air pada PAMDes Pandanrejo
Menggunakan Segmentasi Digit dan MobileNetV2

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, 25 Juni 2026
Yang membuat pernyataan,



Moch. Minanur Rahman
NIM. 220605110112

MOTTO

*“All of the good in my life is from God’s good grace,
but all of the bad in my life is my fault.”*

HALAMAN PERSEMBAHAN

Alhamdulillah rabbil 'alamin, segala puji bagi Allah Swt. atas limpahan rahmat, nikmat, kesehatan, serta karunia-Nya yang tiada henti. Shalawat serta salam semoga senantiasa tercurah kepada Nabi Muhammad saw., beserta keluarga, sahabat, dan seluruh umatnya. Atas izin, pertolongan, serta ridha Allah Swt., penulis diberikan kekuatan, kesabaran, dan kemudahan sehingga skripsi ini dapat diselesaikan dengan baik.

Dengan penuh rasa syukur, karya sederhana ini penulis persembahkan kepada Ayah dan Ibu tercinta sebagai wujud bakti, cinta, kasih sayang, dan rasa terima kasih yang mendalam atas segala doa, pengorbanan, dukungan, serta motivasi yang tidak pernah putus. Setiap perjuangan dan ketulusan yang telah diberikan menjadi sumber semangat bagi penulis untuk terus melangkah hingga mencapai tahap ini.

Persembahan ini juga penulis tujukan kepada adik perempuan tercinta serta seluruh keluarga besar yang senantiasa memberikan doa, perhatian, dukungan, dan motivasi kepada penulis selama menempuh pendidikan. Kehadiran, kepedulian, dan semangat yang mereka berikan menjadi sumber kekuatan bagi penulis.

Kepada para sahabat, teman-teman, serta seluruh pihak yang telah memberikan doa, dukungan, bantuan, dan semangat kepada penulis selama proses penyusunan skripsi ini, penulis menyampaikan ucapan terima kasih yang sebesar-besarnya. Segala bentuk kontribusi yang diberikan menjadi bagian penting dalam perjalanan penulis hingga skripsi ini dapat diselesaikan dengan baik.

KATA PENGANTAR

Alhamdulillahillāhirabbil ‘ālamīn, segala puji bagi Allah *Subhānahu wa Ta ‘ālā*. yang telah melimpahkan rahmat, taufik, hidayah, serta karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “*Pengembangan Aplikasi Mobile OCR untuk Pembacaan Meteran Air pada PAMDes Pandanrejo Menggunakan Segmentasi Digit dan MobileNetV2*” dengan baik. Shalawat serta salam semoga senantiasa tercurah kepada junjungan kita Nabi Muhammad *Shollallohu ‘Alaihi Wasallam*, beserta keluarga, sahabat, dan seluruh pengikutnya hingga akhir zaman. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Prof. Dr. Hj. Ilfi Nurdiana, M.Si., CAHRM., CRMP., selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Agus Mulyono, M.Kes., selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Bapak Supriyono, M.Kom., selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Ibu Nur Fitriyah Ayu Tunjung Sari, M.Cs., selaku dosen pembimbing I dan Bapak Nurizal Dwi Priandani, M.Kom., selaku dosen pembimbing II, yang telah memberikan bimbingan, arahan, masukan, serta motivasi kepada

penulis selama proses penyusunan skripsi ini. Terima kasih atas waktu, kesabaran, dan perhatian yang telah diberikan sehingga penelitian dan penulisan skripsi ini dapat terselesaikan dengan baik dan tepat waktu.

5. Bapak A'la Syauqi, M.Kom., selaku dosen penguji I dan Ibu Roro Inda Melani, M.T. M.Sc., selaku dosen penguji II, yang telah memberikan evaluasi, saran, serta berbagai masukan yang berharga sehingga penelitian dan penulisan skripsi ini dapat tersusun dengan lebih baik. Terima kasih atas arahan dan perhatian yang diberikan dalam proses penyusunan skripsi.
6. Seluruh dosen, laboran, dan staff Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang, penulis menyampaikan ucapan terima kasih atas ilmu yang telah di berikan selama masa perkuliahan. Semoga segala ilmu dan kebaikan yang telah diberikan menjadi amal jariyah serta memberikan manfaat yang berkelanjutan bagi penulis di masa yang akan datang.
7. Ibu Nia Faricha, S.Si., selaku admin Program Studi Teknik Informatika, atas bantuan, informasi, dan dukungan administratif yang sangat membantu selama masa studi hingga proses penyusunan skripsi.
8. Ibu tercinta Siti Aisah, yang senantiasa memberikan kasih sayang, dukungan, motivasi, serta pengorbanan yang tiada henti kepada penulis. Keindahan dan ketulusan doamu yang tidak pernah putus menjadi penyerta dalam setiap perjuangan penulis, sehingga berbagai proses dan tantangan dapat dilalui hingga skripsi ini terselesaikan dengan baik.

9. Ayah tercinta Sugianto, yang senantiasa menjadi teladan dalam kerja keras, keteguhan, dan tanggung jawab. Terima kasih atas doa, dukungan, serta pengorbanan yang telah diberikan selama ini. Setiap nasihat dan perjuangan beliau selalu menjadi sumber motivasi bagi penulis untuk terus berusaha hingga mampu menyelesaikan skripsi ini dengan baik dan tepat waktu.
10. Adik Umi Habibah dan Adik Siti Fatimah, yang selalu hadir memberikan semangat, doa, dan dukungan kepada penulis dalam setiap proses perjalanan pendidikan.
11. Seluruh keluarga besar penulis yang tidak dapat disebutkan satu per satu, terima kasih atas doa, dukungan, perhatian, dan semangat yang telah diberikan kepada penulis selama menempuh pendidikan.
12. Sahabat sejati grup "*Clutch or ngulang*" yang senantiasa kebersamai penulis selama masa perkuliahan, yaitu Muhamad Radiyudin, Muhammad Annas Bintar Putra, Nandito Nur Permata Prasetyo. Terima kasih atas dukungan, kebersamaan, serta berbagai pengalaman yang telah dibagikan sehingga perjalanan akademik ini terasa lebih ringan dan dapat diselesaikan tepat waktu.
13. Ucapan terima kasih penulis sampaikan kepada keluarga besar Teknik Informatika angkatan 2022 Infinity, yang telah memberikan kebersamaan, dukungan, dan pengalaman berharga selama masa perkuliahan.
14. Teruntuk diri sendiri, terima kasih telah berjuang sejauh ini, tetap melangkah meski menghadapi rasa lelah dan keraguan. Terima kasih telah bertahan dan tidak menyerah, walaupun perjalanan ini tidak selalu mudah.

Semoga langkah ke depan senantiasa dipenuhi keberanian, kebaikan, dan kesempatan yang lebih luas.

Penulis menyadari bahwa skripsi ini masih jauh dari kata sempurna dan masih terdapat berbagai keterbatasan yang memerlukan penyempurnaan di masa mendatang. Namun demikian, penulis meyakini bahwa proses belajar tidak memiliki batas akhir, sehingga setiap kritik dan saran yang membangun akan menjadi bekal berharga untuk terus berkembang. Bagi penulis, skripsi yang baik bukanlah skripsi yang sempurna, melainkan skripsi yang dapat diselesaikan dengan penuh tanggung jawab dan tepat pada waktunya. Oleh karena itu, penulis berharap skripsi ini dapat memberikan manfaat, baik bagi pengembangan ilmu pengetahuan maupun bagi pihak-pihak yang membutuhkan sebagai referensi untuk penelitian selanjutnya.

Malang, 25 Juni 2026

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	xii
DAFTAR GAMBAR	xiv
DAFTAR TABEL	xvi
ABSTRAK	xvii
ABSTRACT	xviii
الملخص	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah	5
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	5
BAB II STUDI PUSTAKA	7
2.1 Penelitian Terkait	7
2.1.1 Ringkasan Arsitektur CNN pada Penelitian Terkait	12
2.2 <i>Optical Character Recognition (OCR)</i>	13
2.3 Citra Digital	14
2.3.1 Pengolahan Citra	16
2.4 Segmentasi Digit pada OCR	18
2.5 Augmentasi Data	20
2.5.1 <i>Brightness Adjustment</i>	20
2.5.2 <i>Contrast Adjustment</i>	21
2.5.3 <i>Gaussian Blur</i>	21
2.5.4 <i>Gaussian Noise</i>	22
2.5.5 <i>Rotation</i>	22
2.6 <i>Deep Learning</i>	23
2.7 <i>Convolutional Neural Network (CNN)</i>	24
2.8 MobileNetV2	34
2.8.1 <i>Depthwise Separable Convolution</i>	35
2.8.2 <i>Inverted Residual Block</i>	36
2.8.3 <i>Linear Bottleneck</i>	37
2.8.4 Arsitektur	38
2.9 <i>TensorFlow</i> dan <i>TensorFlow Lite</i>	40
2.10 Integrasi Model OCR dalam Aplikasi <i>Mobile</i>	41
2.11 <i>Waterfall</i> Model	42

2.12	Evaluasi Model	44
2.13	Akurasi OCR Secara Keseluruhan.....	46
2.14	<i>Blackbox Testing</i>	47
BAB III DESAIN DAN IMPLEMENTASI		49
3.1	Desain Penelitian	49
3.1.1	Tahapan Penelitian	49
3.1.2	Pengumpulan Data	50
3.1.3	Desain Sistem.....	50
3.1.4	<i>Input Dataset</i>	52
3.1.5	<i>Preprocessing</i>	53
3.1.6	<i>Data Splitting</i>	56
3.1.7	Pemodelan MobileNetV2.....	56
3.1.8	Skenario Eksperimen Penentuan Model Terbaik	59
3.1.9	Desain Sistem Aplikasi <i>Mobile OCR</i>	59
3.1.10	Evaluasi Sistem Aplikasi.....	75
3.2	Implementasi.....	76
3.2.1	Implementasi Pengumpulan Dataset	76
3.2.2	Implementasi Pemodelan MobileNetV2	78
3.2.3	Implementasi Skenario Eksperimen Penentuan Model Terbaik	87
3.2.4	Implementasi Aplikasi <i>Mobile OCR</i>	90
3.2.5	Implementasi Evaluasi Sistem Aplikasi.....	98
3.2.6	<i>Experimental Setup</i>	101
BAB IV HASIL DAN PEMBAHASAN		103
4.1	Hasil.....	103
4.1.1	Hasil Pengujian Skenario Penentuan Model Terbaik.....	103
4.1.2	Hasil Pengujian <i>Blackbox Testing</i>	124
4.1.3	Hasil Evaluasi Akurasi OCR dalam Pengenalan Digit	127
4.2	Pembahasan	128
4.3	Integrasi Islam	132
4.3.1	<i>Muamalah Ma'a Allah</i>	132
4.3.2	<i>Muamalah Ma'a An-Naas</i>	134
BAB V KESIMPULAN DAN SARAN		136
5.1	Kesimpulan.....	136
5.2	Saran	136
DAFTAR PUSTAKA		
LAMPIRAN		

DAFTAR GAMBAR

Gambar 2. 1 Jaringan saraf deep learning	23
Gambar 2. 2 Arsitektur CNN secara umum	24
Gambar 2. 3 Proses konvolusi.....	25
Gambar 2. 4 Proses konvolusi dengan nilai stride 3	27
Gambar 2. 5 Penambahan padding.....	27
Gambar 2. 6 Max pooling	30
Gambar 2. 7 Proses flatten	31
Gambar 2. 8 Proses Fully Connected Layer.....	32
Gambar 2. 9 Perbandingan konvolusi normal dan konvolusi depthwise separable.....	35
Gambar 2. 10 Inverted residual block	37
Gambar 2. 11 Arsitektur umum MobileNetV2	38
Gambar 2. 12 Tahapan pengembangan perangkat lunak model Waterfall	42
Gambar 3. 1 Tahapan penelitian	49
Gambar 3. 2 Desain sistem	51
Gambar 3. 3 Arsitektur MobileNetV2	57
Gambar 3. 4 Arsitektur Aplikasi	63
Gambar 3. 5 Alur Bisnis	65
Gambar 3. 6 ERD.....	67
Gambar 3. 7 DFD.....	68
Gambar 3. 8 Desain UI Login	69
Gambar 3. 9 Desain UI Halaman Beranda.....	69
Gambar 3. 10 Desain UI Halaman Profil	70
Gambar 3. 11 Desain UI Halaman Pengaturan	71
Gambar 3. 12 Desain UI Halaman Histori	71
Gambar 3. 13 Desain UI Halaman Input Baca Meter	72
Gambar 3. 14 Desain UI Halaman Daftar Pelanggan	73
Gambar 3. 15 Desain UI Halaman Ambil Gambar	73
Gambar 3. 16 Skenario Eksperimen Penentuan Model Terbaik	88
Gambar 3. 17 Halaman Login.....	91
Gambar 3. 18 Halaman Beranda	91
Gambar 3. 19 Halaman Profil	92
Gambar 3. 20 Halaman Pengaturan	93
Gambar 3. 21 Halaman Input Baca Meter	94
Gambar 3. 22 Halaman Daftar Pelanggan.....	95
Gambar 3. 23 Halaman Ambil Gambar	96
Gambar 3. 24 Halaman Histori	97
Gambar 3. 25 Halaman Detail Histori.....	98
Gambar 4. 1 Grafik Proses Pretraining Adam	107
Gambar 4. 2 Grafik Proses Fine-tuning Adam.....	108

Gambar 4. 3 Confussion Matrix Adam	109
Gambar 4. 4 Grafik Proses Pretraining SGD	110
Gambar 4. 5 Grafik Proses Fine-tuning SGD	111
Gambar 4. 6 Confussion Matrix SGD.....	112
Gambar 4. 7 Grafik Proses Pretraining RMSProp	113
Gambar 4. 8 Grafik Proses Fine-tuning RMSProp	114
Gambar 4. 9 Confussion Matrix RMSProp.....	115
Gambar 4. 10 Grafik Proses Pretraining Skenario 5	117
Gambar 4. 11 Grafik Proses Fine-tuning Skenario 5	118
Gambar 4. 12 <i>Confussion Matrix</i> Skenario 5	119
Gambar 4. 13 Grafik Proses Pretraining Skenario 6	120
Gambar 4. 14 Grafik Proses Fine-tuning Skenario 6	121
Gambar 4. 15 <i>Confussion Matrix</i> Skenario 6	123
Gambar 4. 16 Perbandingan Performa Seluruh Skenario	131

DAFTAR TABEL

Tabel 2. 1 Penelitian terdahulu.....	11
Tabel 3. 1 Kebutuhan Fungsional	60
Tabel 3. 2 Kebutuhan Non-Fungsional.....	61
Tabel 3. 3 Contoh Data pada Dataset DIGITS.....	77
Tabel 3. 4 Pemotongan Digit pada Gambar Asli	78
Tabel 3. 5 Skenario Blackbox Testing	98
Tabel 3. 6 Simulasi Evaluasi Model OCR	100
Tabel 3. 7 Tabel Spesifikasi Komputer untuk Pelatihan Model	102
Tabel 3. 8 Tabel Spesifikasi Laptop untuk Pengembangan Aplikasi.....	102
Tabel 4. 1 Hasil Training Berdasarkan Optimizer	104
Tabel 4. 2 Hasil Evaluasi Berdasarkan Optimizer	104
Tabel 4. 3 Hasil Training berdasarkan Learning Rate	105
Tabel 4. 4 Hasil Evaluasi Berdasarkan Learning Rate.....	106
Tabel 4. 5 Hasil Pengujian Aplikasi Mobile OCR dengan Blackbox Testing	125
Tabel 4. 6 Perbandingan Hasil Prediksi Sistem OCR dengan Nilai Asli pada Meteran Air	127

ABSTRAK

Rahman, Moch Minanur. 2026. **Pengembangan Aplikasi *Mobile* OCR untuk Pembacaan Meteran Air pada PAMDes Pandanrejo Menggunakan Segmentasi Digit dan MobileNetV2**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Nur Fitriyah Ayu Tunjung Sari, M.Cs (II) Nurizal Dwi Priandani, M.Kom.

Kata kunci: *Mobile* OCR, MobileNetV2, PAMDes Pandanrejo.

PAMDes Pandanrejo menerapkan proses pencatatan meteran air pelanggan secara manual dengan menggunakan formulir kertas. Metode tersebut berpotensi menimbulkan kesalahan pencatatan, ketidaksesuaian data, serta membutuhkan waktu yang relatif lama dalam proses rekapitulasi dan pengolahan informasi. Kondisi ini menjadi tantangan di era digitalisasi yang menuntut pengelolaan data secara cepat, akurat, dan terintegrasi. Oleh karena itu, diperlukan penerapan sistem informasi berbasis digital untuk meningkatkan efisiensi proses pencatatan meteran air, pengelolaan data pelanggan, dan perhitungan tagihan. Penelitian ini mengembangkan aplikasi *mobile* OCR untuk membantu proses pembacaan digit meteran air dan pencatatan hasil pembacaan pada PAMDes Pandanrejo serta menganalisis pengaruh penggunaan *optimizer* dan variasi *learning rate* pada fase *fine-tuning* terhadap performa model MobileNetV2. Pelatihan model menggunakan kombinasi *dataset* DIGITS dan data lapangan, dengan performa terbaik diperoleh menggunakan *optimizer* Adam dan *learning rate* $1e-5$ pada fase *fine-tuning*, menghasilkan nilai *accuracy* 96,51%, *precision* 96,63%, *recall* 96,51%, dan *f1-score* 96,52%. Pengujian fungsionalitas aplikasi menggunakan *blackbox testing* menunjukkan 19 skenario memperoleh status *valid* dengan tingkat keberhasilan 100%. Evaluasi akurasi OCR pada digit meteran air pelanggan PAMDes Pandanrejo menghasilkan nilai sebesar 87,74%.

ABSTRACT

Rahman, Moch Minanur. 2026. **Development of a Mobile OCR Application for Reading Water Meters at PAMDes Pandanrejo Using Digit Segmentation and MobileNetV2**. Thesis. Department of Informatics Engineering, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University Malang. Advisor: (I) Nur Fitriyah Ayu Tunjung Sari, M.Cs (II) Nurizal Dwi Priandani, M.Kom.

PAMDes Pandanrejo currently uses a manual process to record customer water meter readings using paper forms. This method is prone to recording errors and data discrepancies, and it requires a relatively long time for data compilation and processing. This situation poses a challenge in the era of digitalization, which demands fast, accurate, and integrated data management. Therefore, the implementation of a digital-based information system is necessary to improve the efficiency of the water meter reading process, customer data management, and billing calculations. This study developed an OCR mobile application to assist in reading water meter digits and recording the readings at PAMDes Pandanrejo, as well as to analyze the impact of using different optimizers and varying learning rates during the fine-tuning phase on the performance of the MobileNetV2 model. Model training utilized a combination of the DIGITS dataset and field data, with the best performance achieved using the Adam optimizer and a learning rate of $1e-5$ during the fine-tuning phase, yielding an accuracy of 96.51%, precision of 96.63%, recall of 96.51%, and an F1-score of 96.52%. Functional testing of the application using black-box testing showed that 19 scenarios were validated with a 100% success rate. The OCR accuracy evaluation of the water meter digits for PAMDes Pandanrejo customers yielded a value of 87.74%.

Key words: Mobile OCR, MobileNetV2, PAMDes Pandanrejo.

الملخص

رحمان، محمد منن. 2026. تطوير تطبيق OCR للهواتف المحمولة لقراءة عدادات المياه في PAMDes Pandanrejo باستخدام تقسيم الأرقام MobileNetV2. أطروحة. قسم هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرفان : الأول (نور فيترياه أيو تونج ساري، ماجستير في علوم الكمبيوتر) الثاني (نوريزال دوي فريانداني، ماجستير في علوم الكمبيوتر).

الكلمات المفتاحية: OCR المحمول، MobileNetV2، PAMDes Pandanrejo

تطبق مؤسسة المياه القروية (PAMDes) في باندانريجو عملية تسجيل قراءات عدادات المياه للعملاء يدوياً باستخدام نماذج ورقية. وقد تؤدي هذه الطريقة إلى حدوث أخطاء في التسجيل وتباين في البيانات، كما أنها تستغرق وقتاً طويلاً نسبياً في عملية تجميع المعلومات ومعالجتها. ويشكل هذا الوضع تحدياً في عصر الرقمنة الذي يتطلب إدارة البيانات بسرعة ودقة وبشكل متكامل. لذلك، من الضروري تطبيق نظام معلومات رقمي لتحسين كفاءة عملية قراءة عدادات المياه، وإدارة بيانات العملاء، وحساب الفواتير. يهدف هذا البحث إلى تطوير تطبيق جوال يعمل بتقنية التعرف الضوئي على الحروف (OCR) للمساعدة في عملية قراءة أرقام عدادات المياه وتسجيل نتائج القراءة في شركة PAMDes Pandanrejo، بالإضافة إلى تحليل تأثير استخدام مُحسِّن (optimizer) وتباين معدل التعلم (learning rate) في مرحلة الضبط الدقيق (fine-tuning) على أداء نموذج MobileNetV2. تم تدريب النموذج باستخدام مزيج من مجموعة بيانات DIGITS والبيانات الميدانية، حيث تم الحصول على أفضل أداء باستخدام مُحسِّن Adam ومعدل التعلم $e-51$ في مرحلة الضبط الدقيق، مما أدى إلى تحقيق دقة بنسبة 96,51%، ودقة تحديد بنسبة 96,63%، ومعدل استرجاع بنسبة 96,51%، ودرجة F1 بنسبة 96,52%. وأظهر اختبار وظائف التطبيق باستخدام اختبار الصندوق الأسود أن 19 سيناريو حصلت على حالة «صالح» بنسبة نجاح بلغت 100%. وأسفر تقييم دقة التعرف الضوئي على الحروف (OCR) لأرقام عدادات مياه عملاء PAMDes Pandanrejo عن قيمة بلغت 87,74%.

BAB I

PENDAHULUAN

1.1 Latar Belakang

PAMDes (Perusahaan Air Minum Desa) adalah unit pelayanan publik pada tingkat desa yang berfungsi menyediakan dan menyalurkan air bersih kepada masyarakat. PAMDes adalah bagian dari BUMDes (Badan Usaha Milik Desa), yang dibangun oleh aparat desa dan masyarakat untuk mengatur kapasitas desa secara mandiri dengan tujuan mengoptimalkan kesejahteraan warga (Kusmacherausanni & Ismanto, 2024). Banyak petugas layanan air bersih masih melakukan pencatatan meteran air secara manual (Paksi & Prihartono, 2019). Salah satunya adalah PAMDes Pandanrejo yang hingga saat ini masih menggunakan metode pencatatan manual dalam proses pencatatan meteran air (Rahman et al., 2025). Berdasarkan hasil wawancara dengan pihak PAMDes Pandanrejo, proses tersebut masih menghadapi beberapa tantangan dalam pelaksanaannya. Metode pencatatan manual memiliki keterbatasan dibandingkan sistem yang telah terdigitalisasi karena belum memanfaatkan otomatisasi yang bisa meningkatkan efisiensi terhadap pengelolaan data, mempermudah akses informasi, meminimalisir kesalahan manusia, dan menambah akurasi data (Saputra et al., 2024).

Proses pencatatan angka pada meter air dapat dibantu dengan memanfaatkan teknologi OCR (*Optical Character Recognition*). Teknologi ini berfungsi untuk mengidentifikasi karakter teks atau angka yang terdapat pada gambar digital, kemudian mengubah hasil identifikasi tersebut menjadi bentuk data

yang bisa diolah oleh komputer. (Mulyadi & Veronika, 2025). Dengan memanfaatkan kamera pada perangkat *mobile*, OCR dapat digunakan untuk membaca angka pemakaian air pada meteran secara otomatis sehingga berpotensi meningkatkan efisiensi proses pencatatan dan pengelolaan data meteran air.

Berdasarkan permasalahan yang terjadi di lapangan, Penelitian ini menawarkan pengembangan aplikasi *mobile* OCR dengan pendekatan segmentasi digit untuk pembacaan angka pada meteran air. Pada prosesnya, setiap digit pada meteran air dipisahkan dan diproses secara individual, kemudian hasil pengenalan dari setiap digit dikombinasikan untuk memperoleh nilai akhir pemakaian air. Proses pengenalan digit dilakukan menggunakan model MobileNetV2. Berbeda dengan arsitektur CNN konvensional yang menggunakan lapisan konvolusi standar, MobileNetV2 menerapkan kombinasi *depthwise convolution* serta *pointwise convolution* di mana mereduksi jumlah parameter beserta beban komputasi dalam jumlah besar tanpa menghilangkan kemampuan ekstraksi fitur penting (Miftahuddin & Zaelani S, 2022). Selain itu, dibandingkan dengan berbagai arsitektur CNN yang telah banyak diterapkan, seperti VGG16, DenseNet201, serta Xception, MobileNetV2 menawarkan waktu komputasi yang memiliki kecepatan pemrosesan lebih tinggi dengan tingkat akurasi yang tetap kompetitif sehingga cocok diimplementasikan pada *device* dengan keterbatasan sumber daya (Naufal & Kusuma, 2021). Efisiensi tersebut semakin ditingkatkan melalui penggunaan *linear bottleneck* beserta *shortcut connection* di mana mendukung proses ekstraksi fitur berlangsung lebih optimal dengan beban komputasi lebih rendah. Atas dasar tersebut, penelitian ini menggunakan MobileNetV2 karena mampu menghasilkan

stabilitas antara akurasi pengenalan digit dan efisiensi komputasi, sehingga sesuai untuk mendukung implementasi aplikasi *mobile* OCR dengan konsumsi sumber daya yang rendah.

Islam menekankan pentingnya ketelitian, keakuratan informasi, serta tanggung jawab dalam setiap tindakan yang dilakukan. Prinsip ini dijelaskan pada Al-Qur'an, khususnya *Surah Al-Isra'* ayat 36. Ayat tersebut mengingatkan bahwa setiap informasi yang diperoleh melalui penglihatan dan pengetahuan harus dapat dipertanggungjawabkan, sehingga menuntut kehati-hatian dalam proses pencatatan dan pengelolaan data.

وَلَا تَقْفُ مَا لَيْسَ لَكَ بِهِ عِلْمٌ إِنَّ السَّمْعَ وَالْبَصَرَ وَالْفُؤَادَ كُلُّ أُولَٰئِكَ كَانَ عَنْهُ مَسْئُولًا ۝ ٣٦

“Janganlah engkau mengikuti sesuatu yang tidak kauketahui. Sesungguhnya pendengaran, penglihatan, dan hati nurani, semua itu akan diminta pertanggungjawabannya.” (QS. Al-Isra’: 36)

Ulama Tantawi Jauhari di dalam *“Tafsir Al-Jawāhir fī Tafsīr al-Qur’ān al-Karīm”* menjelaskan *QS. Al-Isra'* ayat 36 merupakan peringatan agar manusia menggunakan pendengaran, penglihatan, serta hati secara bertanggung jawab ketika memperoleh dan mengolah informasi. Ayat ini menegaskan larangan mengikuti suatu hal tanpa dasar ilmu yang telah jelas, sehingga menuntut adanya proses verifikasi dan kehati-hatian dalam menerima informasi (Khotimah et al., 2025). Prinsip tanggung jawab indera tersebut menjadi landasan penting dalam pengelolaan data yang akurat dan dapat dipertanggungjawabkan, termasuk dalam praktik pencatatan dan pengelolaan informasi pada layanan publik di tingkat desa.

Berdasarkan teori dan hasil identifikasi permasalahan, penelitian ini berfokus pada pengembangan aplikasi *mobile* OCR supaya dapat membantu kegiatan pembacaan meteran air secara otomatis pada PAMDes Pandanrejo. Pengembangan aplikasi mengadopsi metode *Waterfall*, di mana membagi urutan proses pengembangan ke dalam beberapa langkah, yaitu analisis kebutuhan, perancangan sistem, implementasi, serta pengujian, sehingga setiap proses dapat terlaksana melalui proses yang sistematis dan didukung pencatatan yang baik. Pendekatan yang diterapkan dalam proses pengenalan angka pada citra meteran air adalah segmentasi digit yang kemudian diklasifikasikan menggunakan model MobileNetV2. Dengan judul “Pengembangan Aplikasi *Mobile* OCR untuk Pembacaan Meteran Air pada Pamdes Pandanrejo Menggunakan Segmentasi Digit dan MobileNetV2”, aplikasi ini diharapkan dapat mempercepat dan mempermudah proses pendataan pemakaian air, serta mendukung terwujudnya layanan pengelolaan air bersih dengan lebih modern dan efektif.

1.2 Rumusan Masalah

1. Bagaimana mengembangkan aplikasi *mobile* OCR untuk membantu proses pembacaan digit meteran air dan pencatatan hasil pembacaan pada PAMDes Pandanrejo?
2. Bagaimana pengaruh penggunaan *optimizer* dan variasi *learning rate* pada fase *fine-tuning* terhadap performa model MobileNetV2 dalam pembacaan digit meteran air?

1.3 Batasan Masalah

1. Penelitian hanya berfokus dalam pengembangan aplikasi *mobile* OCR guna mengenali digit pada meteran air PAMDes Pandanrejo.
2. *Dataset* yang digunakan adalah DIGITS (<https://www.kaggle.com/datasets/karnikakapoor/digits>), yang berisi citra digit angka dengan 10 kelas (digit 0 hingga 9) dan data digit yang diperoleh dari lapangan.

1.4 Tujuan Penelitian

1. Mengembangkan aplikasi *mobile* OCR untuk membantu proses pembacaan digit meteran air dan pencatatan hasil pembacaan pada PAMDes Pandanrejo.
2. Menganalisis pengaruh penggunaan *optimizer* dan variasi *learning rate* pada fase *fine-tuning* terhadap performa model MobileNetV2 dalam pembacaan digit meteran air.

1.5 Manfaat Penelitian

1. Pemerintah Desa Pandanrejo

Penelitian diharapkan bisa memberikan dampak positif langsung bagi Pemerintah Desa Pandanrejo melalui penyediaan aplikasi *mobile* OCR yang dapat membantu petugas PAMDes dalam mempercepat dan mempermudah proses pendataan jumlah pemakaian air pelanggan.

2. Peneliti dan Akademisi

Penelitian yang dilakukan diharapkan memberikan kontribusi terhadap pengembangan teknologi OCR, khususnya melalui penerapan metode segmentasi digit dan model MobileNetV2 pada studi kasus pembacaan meteran air. Hasil yang diperoleh diharapkan dapat mendukung penelitian berikutnya yang berfokus pada pengembangan sistem OCR, pemrosesan citra, serta penerapan *deep learning* dalam bidang rekayasa perangkat lunak.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terkait

Bagian ini menyajikan rangkuman berbagai studi sebelumnya yang mengandung keterkaitan topik penelitian. Kajian terhadap penelitian terdahulu dilakukan guna mendapatkan pemahaman mendalam terhadap perkembangan penelitian sejenis serta untuk mengidentifikasi pendekatan dan temuan yang relevan. Literatur yang digunakan dipilih berdasarkan kesesuaian dengan fokus penelitian dan diperoleh dari sumber-sumber akademik. Melalui peninjauan ini, diharapkan terbentuk dasar teoritis yang kuat sebagai pendukung arah penelitian.

Beberapa penelitian terdahulu telah mengembangkan sistem OCR untuk berbagai kasus penggunaan, termasuk pada proses pembacaan meteran air yang masih banyak dilakukan secara manual. Salah satu penelitian tersebut adalah karya (Putra et al., 2021), yang menerapkan metode *Run-Length Smearing Algorithm* (RLSA) untuk mengidentifikasi angka pada citra meteran air. Penelitian ini dilatarbelakangi oleh permasalahan pencatatan manual yang memakan waktu serta sering terjadi kesalahan petugas. Melalui tahapan *preprocessing*, segmentasi, dan penerapan RLSA untuk mendeteksi area karakter, sistem yang dibangun mampu mencapai tingkat keberhasilan identifikasi sebesar 83% dari total 90 sampel citra yang diuji. Temuan ini menunjukkan bahwa metode RLSA efektif dalam membantu proses OCR untuk pembacaan angka meteran air dan memberikan kontribusi penting bagi pengembangan teknologi otomasi pada bidang tersebut.

Selanjutnya, penelitian oleh (Tanjung et al., 2025), berfokus pada implementasi CNN untuk pengenalan digit angka tulisan tangan. Penelitian ini dilatarbelakangi oleh tingginya tingkat kerumitan dan variasi gaya penulisan individu yang sering menyebabkan kesalahan pada metode pengenalan tradisional. Untuk mengatasi tantangan pada citra dunia nyata, sistem ini mengintegrasikan alur pemrosesan yang komprehensif, dimulai dari tahap praproses menggunakan *adaptive thresholding* dan *morphological filtering* guna mempertegas kontur angka. Selanjutnya, dilakukan tahap segmentasi berbasis kontur yang berfungsi memisahkan setiap digit dalam citra agar dapat diproses secara individual. Karakter yang telah terpisah kemudian masuk ke tahap klasifikasi menggunakan arsitektur CNN yang terdiri dari lapisan *Conv2D* dan *MaxPooling2D*. Hasil pengujian menghasilkan performa sangat baik, yaitu nilai akurasi sebesar 99,09% pada *dataset* MNIST. Penelitian ini menegaskan bahwa pendekatan CNN melalui tahapan segmentasi dan klasifikasi yang tepat memberikan keunggulan signifikan dalam aspek akurasi.

Penelitian berikutnya oleh (Shawon et al., 2022), berfokus pada pengenalan angka tulisan tangan bahasa Bengali menggunakan algoritma CNN yang dikombinasikan dengan metode *Explainable AI* (XAI). Penelitian ini dilatarbelakangi oleh sulitnya menemukan aplikasi OCR berkualitas tinggi untuk bahasa Bengali dibandingkan dengan bahasa Inggris. Untuk memberikan pemahaman mendalam terhadap tingkat keputusan model yang sering kali dianggap sebagai "kotak hitam", studi yang dilakukan menerapkan metode Grad-CAM untuk memvisualisasikan area atau piksel tertentu dalam citra yang menjadi fokus utama

prediksi model. Eksperimen dikerjakan menggunakan dua dataset, yaitu Ekush dan NumtaDB, dengan arsitektur CNN. Hasil pada *dataset* Ekush menghasilkan performa yang sangat baik, yaitu nilai akurasi pengujian sebesar 96,71%, yang terbukti lebih unggul daripada *Random Forest* maupun NuSVC.

Penelitian selanjutnya dilakukan oleh (Sainyakit & Bisilisin, 2025), yang berfokus pada klasifikasi penggunaan meteran air menggunakan metode CNN. Penelitian ini dilatarbelakangi oleh tantangan pembacaan meteran secara manual yang rentan terhadap kesalahan manusia, keterlambatan pencatatan karena medan yang sulit, serta hambatan akses fisik seperti meteran yang berada di dalam pagar rumah pelanggan. Untuk mengatasi permasalahan tersebut, sistem ini mengintegrasikan teknologi OCR melalui alur prapemrosesan citra yang mendalam, meliputi *grayscale conversion*, *binarization*, *noise removal*, *deskewing*, hingga tahap *cropping* dan *resizing* untuk menstandarkan input gambar. Hasil penelitian menghasilkan model CNN mampu mengklasifikasikan citra meteran air dari dua wilayah berbeda, yaitu Kelurahan Manulai II dan Kelurahan TDM, dengan akurasi tertinggi sebesar 95% pada *fold* pertama dan akurasi keseluruhan sebesar 91%, serta mencapai sensitivitas 90% dan spesifisitas 92%.

Penelitian berikutnya dilakukan oleh (Goel & Ganatra, 2023) yang menguji efektivitas berbagai arsitektur *deep learning* untuk klasifikasi angka menggunakan teknik *transfer learning*. Salah satu temuan penting dalam penelitian tersebut adalah performa arsitektur MobileNetV2 yang mampu mencapai akurasi pengujian sebesar 92,08%. Penelitian tersebut menyoroti bahwa meskipun terdapat model lain dengan akurasi sedikit lebih tinggi, MobileNetV2 menawarkan keunggulan

signifikan dalam hal kecepatan inferensi (29,22 ms) dan ukuran model yang ringan (49,12 MB). Sehingga model MobileNetV2 adalah arsitektur yang sangat kompeten sekaligus efisien guna diimplementasikan pada *device* dengan keterbatasan sumber daya (perangkat seluler).

Selanjutnya penelitian oleh (Sari et al., 2023) berfokus pada pembangunan aplikasi *mobile* untuk pencatatan keuangan pribadi dengan memanfaatkan teknologi OCR pada platform *Android*. Penelitian ini dilatarbelakangi oleh kebutuhan pencatatan transaksi harian yang sering kali dilaksanakan dengan manual yang mengandung kerentanan terhadap kesalahan input dan kurang efisien. Sebagai solusi, aplikasi dikembangkan menggunakan model *Waterfall* dan *framework Xamarin*, dengan integrasi Google Vision API sebagai mesin OCR yang bertugas mengekstraksi teks dari citra setruk belanja. Sistem bekerja melalui alur pengambilan gambar setruk, pengunggahan ke server, proses ekstraksi teks dalam format JSON, hingga pengolahan nilai total transaksi secara otomatis ke dalam laporan *cash flow* pengguna. Pengujian dilakukan melalui *Blackbox Testing*, pengujian performa *framework*, serta pengujian OCR pada berbagai kondisi pencahayaan, jarak, dan kualitas kertas. Hasil penelitian menunjukkan bahwa fitur OCR memperoleh tingkat akurasi sebesar 84,61% (22 keberhasilan dari 26 pengujian), sementara pengujian *Blackbox* mencapai 100% keberhasilan fungsi, serta uji *usability* (UAT) memperoleh nilai rata-rata 81,56%, sehingga aplikasi yang dikembangkan termasuk layak untuk digunakan sebagai *prototipe* pencatatan keuangan berstandar *mobile* dengan integrasi teknologi OCR.

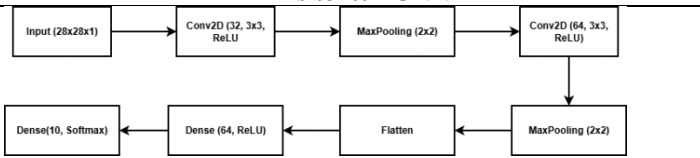
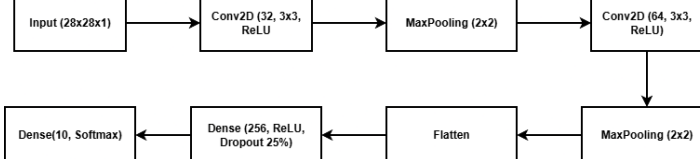
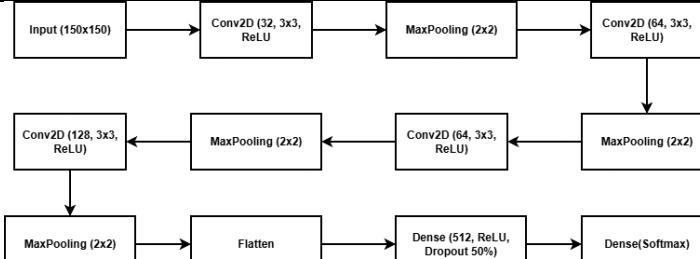
Tabel 2. 1 Penelitian terdahulu

No	Nama Peneliti	Judul Penelitian	Metode	Hasil Penelitian
1.	Putra et al., (2021)	Identifikasi Text Meteran Air Menggunakan Metode <i>Run-Length Smearing Algorithm</i> (RLSA)	RLSA (Algoritma tradisional untuk segmentasi)	Tingkat keberhasilan identifikasi mencapai 83% dari 90 sampel citra meteran air.
2.	Tanjung et al. (2025)	Implementasi Algoritma <i>Convolutional Neural Network</i> (CNN) untuk Pengenalan Angka Tulisan Tangan	CNN + <i>Data Augmentation</i> + <i>Contour-based Segmentation</i>	Akurasi pengujian mencapai 99,09% pada <i>dataset</i> MNIST.
3.	Shawon et al. (2022)	<i>Handwritten Bengali Digit Recognition using CNN and Explainable AI</i>	CNN + Grad-CAM (XAI)	Akurasi pengujian sebesar 96,71% pada <i>dataset</i> Ekush, lebih unggul dibanding Random Forest dan NuSVC.
4.	Sainyakit & Bisilisin (2025)	Klasifikasi Penggunaan Meteran Air Menggunakan <i>Convolutional Neural Network</i> (CNN) pada Perumda Air Minum Kota Kupang Berbasis Citra Digital	CNN + OCR (<i>Grayscale, Thresholding, Deskewing</i>) + 5-Fold Cross Validation.	Akurasi tertinggi 95% (<i>fold</i> pertama) dengan akurasi keseluruhan 91%, sensitivitas 90%, spesifisitas 92%.
5.	Goel & Ganatra (2023)	<i>Handwritten Gujarati Numerals Classification Based on Deep Convolution Neural Networks Using Transfer Learning Scenarios</i>	<i>Perbandingan (10 CNN Models)</i> + <i>Data Augmentation</i>	Arsitektur MobileNetV2 memberikan performa baik dengan nilai akurasi sebesar 92,08%. Selain itu, memiliki waktu inferensi yang cepat (29,22 ms) dan ukuran model yang ringan.
6.	Sari et al. (2023)	<i>Personal Track Your Cash: Prototipe Aplikasi Pembacaan Setruk 0020 Belanja Menggunakan OCR dan Google Vision</i>	SDLC <i>Waterfall</i> + <i>Xamarin</i> + Google Vision API	Implementasi OCR menggunakan Google Vision pada aplikasi pengelola keuangan (<i>cash flow</i>) mencapai akurasi pembacaan setruk sebesar 84,61%. Pengujian <i>Black Box</i> menunjukkan fungsionalitas berjalan 100% dan tingkat usabilitas (UAT) mencapai 81,56%.
7.	Rahman (2026)	Pengembangan Aplikasi <i>Mobile OCR</i> untuk Pembacaan Meteran Air pada PAMDes Pandanrejo Menggunakan Segmentasi Digit dan MobileNetV2	Segmentasi Digit (<i>Overlay Tetap</i>) + MobileNetV2 + <i>Waterfall</i>	Penelitian yang diusulkan (<i>Proposed Research</i>). Mengembangkan aplikasi <i>mobile OCR</i> berbasis segmentasi digit untuk pembacaan meteran air pada PAMDes Pandanrejo

2.1.1 Ringkasan Arsitektur CNN pada Penelitian Terkait

Berdasarkan penelitian terdahulu yang menggunakan metode CNN terdapat perbedaan konfigurasi arsitektur yang digunakan pada masing-masing penelitian. Perbedaan tersebut meliputi ukuran *input*, jumlah *convolution layer*, penggunaan *pooling layer*, jumlah neuron pada *fully connected layer*, serta penerapan teknik regularisasi seperti *dropout*. Ringkasan arsitektur CNN yang digunakan pada penelitian-penelitian tersebut disajikan pada Tabel 2. 2.

Tabel 2. 2 Ringkasan Arsitektur CNN pada Penelitian Terkait

No	Nama Peneliti	Arsitektur CNN
1.	Tanjung et al., (2021)	
2.	Shawon et al. (2025)	
3.	Sainyakit & Bisilisin. (2022)	

Berdasarkan Tabel 2. 2, penelitian Tanjung et al. (2025) dan Shawon et al. (2022) menggunakan arsitektur CNN yang relatif sederhana, yaitu dua *convolution layer* yang masing-masing memiliki 32 dan 64 filter, disertai dua *max pooling layer* untuk proses reduksi dimensi fitur. Perbedaan utama keduanya terletak pada penggunaan *fully connected layer* dan regularisasi, di mana Shawon et al. (2022) menggunakan 256 neuron pada *fully connected layer* serta menambahkan *dropout*

sebesar 25% untuk mengurangi risiko *overfitting*, sedangkan Tanjung et al. (2025) hanya menggunakan 64 neuron tanpa *dropout*.

Sementara itu, penelitian Sainyakit dan Bisilisin (2025) menggunakan arsitektur yang lebih dalam dengan empat *convolution layer* yang terdiri atas 32, 64, 64, dan 128 filter serta empat *max pooling layer*. Selain itu, penelitian tersebut menggunakan *fully connected layer* dengan 512 neuron dan *dropout* sebesar 0,5 sebelum proses klasifikasi. Penambahan jumlah *convolution layer* dan neuron pada *fully connected layer* dapat membuat model mengekstraksi fitur lebih kompleks.

2.2 Optical Character Recognition (OCR)

OCR adalah teknologi yang bertugas mengonversi teks di dalam citra baik tulisan tangan, teks cetak, maupun dokumen hasil pemindaian menjadi data digital yang dapat diolah oleh komputer. Teknologi ini dapat membuat sistem komputer mengenali karakter pada gambar melalui proses akuisisi citra, *preprocessing*, segmentasi, ekstraksi fitur citra, hingga pengenalan karakter secara otomatis (Thanki et al., 2021).

Secara lebih rinci, proses OCR diawali dengan *preprocessing* untuk meningkatkan kualitas citra melalui penghilangan *noise*, peningkatan kontras, koreksi kemiringan, *thresholding*, dan normalisasi guna memastikan karakter dapat dianalisis dengan baik. Tahap berikutnya adalah segmentasi, yaitu pemisahan citra menjadi bagian-bagian tertentu sehingga sistem fokus pada elemen bagian yang relevan. Setelah itu dilakukan *morphological processing* untuk memperjelas bentuk karakter menggunakan operasi seperti dilasi dan erosi, kemudian ekstraksi fitur untuk mengambil ciri khas karakter berupa pola garis, kurva, atau struktur tertentu.

Tahap akhir adalah *recognition* (pengenalan) yang bertugas mengidentifikasi karakter berdasarkan fitur yang telah diekstraksi, dan dapat dilanjutkan dengan *post-processing* untuk memperbaiki kesalahan pengenalan agar hasil teks menjadi lebih akurat (Karthick et al., 2019).

Menurut (Wei et al., 2024), jenis-jenis OCR dapat diklasifikasikan berdasarkan cakupan tugas dan tingkat kompleksitasnya menjadi beberapa kategori utama. Pertama, *Scene Text* OCR, yaitu OCR yang berfokus pada pengenalan teks dalam foto atau citra dunia nyata (*natural scene*) seperti papan nama, plang toko, atau teks pada media luar ruang. Kedua, *Document* OCR, yaitu OCR yang digunakan untuk memproses dokumen terstruktur seperti PDF, buku, laporan, atau halaman hasil pemindaian dengan tata letak yang rapi. Ketiga, *Fine-grained* OCR, yaitu pendekatan yang memungkinkan pengenalan teks pada wilayah tertentu saja (*region-level*) dengan bantuan *prompt* seperti koordinat, warna, atau *bounding box* sehingga cocok untuk proses OCR yang bersifat interaktif. Keempat, *More General* OCR atau sering disebut OCR-2.0, yaitu sistem OCR yang mampu mengenali tidak hanya teks umum, tetapi juga berbagai karakter optik buatan manusia lain seperti rumus matematika dan kimia, tabel kompleks, notasi musik, bentuk geometri, hingga grafik atau *chart* beserta nilai numeriknya.

2.3 Citra Digital

Citra digital didefinisikan sebagai bentuk digital dari suatu objek visual yang memungkinkan komputer melakukan penyimpanan, analisis, dan pemrosesan data. Struktur citra terdiri atas matriks piksel, yaitu elemen-elemen penyusun yang masing-masing memiliki nilai intensitas atau warna sesuai posisinya. Data citra

tersebut dapat diperoleh dari perangkat perekam, seperti kamera digital dan *scanner*, maupun dibangkitkan melalui simulasi komputer. (Dijaya, 2023). Berbeda dengan citra analog yang bersifat kontinu, citra digital gambar diskrit yang dihasilkan oleh proses digitalisasi, yaitu konversi dari gambar analog menjadi gambar diskrit yang membagi gambar menjadi baris dan kolom. Secara matematis, citra digital direpresentasikan dalam bentuk matriks dengan koordinat (x, y) pada bidang dua dimensi, di mana besar intensitas pada setiap titik disimbolkan oleh $f(x, y)$ yang tergantung pada intensitas cahaya yang dipantulkan oleh objek. Keunggulan citra digital terletak pada kemampuannya untuk diproses komputer karena datanya berbentuk numeris, sehingga memungkinkan berbagai operasi pengolahan citra seperti *image restoration* dan *image enhancement* (Mahardika et al., 2017).

Menurut (Arnita et al., 2022), dalam pengolahan citra digital, ada tiga jenis citra yang biasa digunakan berdasarkan representasi warnanya, yaitu citra RGB, *grayscale*, dan citra biner. Ketiga jenis citra ini memiliki ciri dan aplikasi yang berbeda sesuai dengan kebutuhan pemrosesan citra.

1. Citra RGB: Citra dalam bentuk tiga komponen dasar yaitu R (*red*/merah), G (*green*/hijau), dan B (*blue*/biru), dimana setiap komponen warna menggunakan 8-bit dengan nilai yang berkisar antara 0 hingga 255, sehingga kemungkinan kombinasi warna yang dapat disajikan mencapai $255 \times 255 \times 255$ atau 16.581.375 warna. Warna putih direpresentasikan sebagai $RGB(255, 255, 255)$, sedangkan warna hitam direpresentasikan sebagai $RGB(0, 0, 0)$.

2. Citra *Grayscale*: Citra yang hanya memiliki informasi intensitas tanpa informasi warna, dengan gradasi dari hitam hingga putih melalui berbagai tingkat abu-abu, di mana setiap piksel bernilai antara 0 sampai 255, yaitu warna putih dengan nilai 255, sedangkan warna hitam dengan nilai 0.
3. Citra Biner: Citra yang paling sederhana karena hanya mengandung dua kemungkinan nilai piksel yaitu 0 atau 1, nilai 0 berarti warna hitam (latar belakang) dan nilai 1 berarti warna putih (objek), dengan kebutuhan memori yang sangat minimal yaitu hanya 1 bit per piksel, sehingga sangat efisien untuk penyimpanan dan pemrosesan serta memungkinkan penggunaan operasi logika untuk mempercepat waktu komputasi.

2.3.1 Pengolahan Citra

Pengolahan citra adalah tahap awal yang penting dalam sistem pengenalan pola, termasuk OCR, karena proses ini mempersiapkan citra agar memiliki struktur dan format visual yang sesuai untuk tahap ekstraksi fitur. Tahapan ini bertujuan untuk meningkatkan keterbacaan objek, menyesuaikan representasi piksel, serta menyederhanakan informasi warna sehingga citra menjadi mudah diproses oleh model OCR.

Konversi citra RGB ke *grayscale* merupakan salah satu langkah dasar dalam proses pengolahan citra karena menyederhanakan data dari tiga *channel* warna menjadi satu *channel* intensitas tanpa menghilangkan informasi penting. Hal ini penting karena sebagian besar algoritma ekstraksi fitur bekerja berdasarkan intensitas piksel, bukan warna. penggunaan tiga kanal warna pada citra RGB meningkatkan kompleksitas komputasi dan kebutuhan memori, sehingga konversi

ke *grayscale* mempercepat proses dan membuat struktur data lebih sederhana serta tetap mempertahankan detail informasi yang relevan untuk tahapan pemrosesan selanjutnya (Saputri et al., 2025). Metode yang sering digunakan untuk konversi *grayscale* adalah metode *luminosity*, yang membobotkan lebih besar pada *channel* hijau disebabkan mata manusia lebih peka terhadap warna tersebut daripada merah dan biru. Secara matematis, konversi *grayscale* dapat dinyatakan sebagai:

$$I(x, y) = 0.299R(x, y) + 0.587G(x, y) + 0.114B(x, y) \quad (2.1)$$

Keterangan:

$R(x, y)$, $G(x, y)$, dan $B(x, y)$: Nilai intensitas piksel pada masing-masing kanal.
 $I(x, y)$: Nilai piksel *grayscale* pada posisi yang sama,

Resize adalah tahapan vital dalam pengolahan citra yang bertujuan menyesuaikan dimensi citra supaya cocok dengan kebutuhan sistem pemrosesan berikutnya. Pada aplikasi OCR, ukuran citra yang terlalu besar dapat memperlambat proses komputasi, sedangkan citra yang terlalu kecil dapat menyebabkan hilangnya detail karakter sehingga menurunkan akurasi deteksi. Oleh karena itu, *resize* berfungsi menyeimbangkan efisiensi pemrosesan dan kualitas visual dengan mempertahankan proporsi objek. Salah satu metode *resize* yang bisa digunakan adalah interpolasi *bilinear*, yaitu pendekatan estimasi nilai piksel baru berdasarkan rata-rata berbobot empat piksel terdekat pada citra asli (Abdi et al., 2011).

Normalisasi nilai piksel adalah langkah penting pengolahan citra digital, terutama untuk meningkatkan efektivitas dan stabilitas proses pelatihan model *deep learning*. Proses ini bertujuan untuk mengubah skala nilai intensitas piksel dari rentang asli (0 – 255) ke rentang yang lebih kecil dan konsisten, sehingga

memudahkan model dalam mengidentifikasi pola dan mempercepat konvergensi selama pelatihan. Normalisasi sangat penting karena nilai piksel yang besar dapat mengakibatkan gradien yang tidak stabil dan memperlambat proses pembelajaran, sedangkan nilai yang dinormalisasi membantu algoritma optimasi bekerja lebih efektif dengan mengurangi sensitivitas terhadap skala data (Arafat et al., 2025). Metode normalisasi yang dapat digunakan ialah penskalaan *min-max* ke rentang $[0,1]$, yang mempertahankan distribusi relatif nilai piksel sambil menyesuaikan magnitudenya. Secara matematis, normalisasi piksel dapat dinyatakan sebagai:

$$x_{norm} = \frac{x}{255} \quad (2.2)$$

Keterangan:

x : Nilai intensitas piksel asli dalam rentang 0 – 255.

x_{norm} : Nilai piksel yang telah dinormalisasi ke dalam rentang $[0,1]$.

255 : Nilai maksimum intensitas piksel untuk citra yang sebagai faktor pembagi.

2.4 Segmentasi Digit pada OCR

Segmentasi merupakan suatu proses krusial dalam sistem pengenalan karakter yang berfungsi untuk memisahkan setiap karakter yang dideteksi menggunakan area pengamatan (*region*) agar dapat diolah menjadi unit-unit individual (Hanif et al., 2023). Tahapan ini menjadi sangat signifikan dalam alur kerja OCR karena bertujuan untuk memisahkan setiap karakter baik huruf dan angka dari sebuah citra untuk kemudian diklasifikasikan secara akurat, sehingga mempermudah proses konversi citra karakter menjadi format teks digital (Alfajri et al., 2025).

Dalam implementasinya, pendekatan segmentasi karakter pada sistem OCR umumnya terdiri dari dua kategori utama, yaitu *explicit segmentation* dan *implicit*

segmentation. Pendekatan *implicit segmentation* (internal segmentation) merupakan metode di mana proses segmentasi tidak dilakukan secara terpisah sebelum pengenalan, melainkan berlangsung bersamaan dengan tahap klasifikasi karakter. Pada pendekatan ini, sistem menghasilkan sejumlah kandidat potongan karakter (*over-segmentation*), kemudian proses pengenalan akan menentukan hipotesis segmentasi terbaik berdasarkan hasil klasifikasi. Dengan kata lain, segmentasi menjadi bagian dari proses pengenalan itu sendiri. Sebaliknya, *explicit segmentation* (external segmentation) merupakan pendekatan yang secara eksplisit memisahkan karakter terlebih dahulu berdasarkan ciri-ciri visual seperti bentuk, celah (*ligature*), lubang, atau analisis histogram vertikal, sebelum masuk ke tahap klasifikasi. Pada pendekatan ini, proses pemotongan citra menjadi unit karakter dilakukan secara terstruktur dan terpisah, sehingga setiap segmen yang dihasilkan telah divalidasi sebagai kandidat karakter sebelum dikenali oleh model klasifikasi (Rehman et al., 2009).

Pada penelitian ini, pendekatan segmentasi yang diterapkan termasuk dalam kategori *explicit segmentation*, di mana proses pemisahan digit dilakukan sebelum tahap klasifikasi. Namun demikian, segmentasi tidak dilakukan melalui analisis morfologi kompleks atau deteksi otomatis berbasis fitur citra, melainkan melalui desain antarmuka aplikasi yang telah membatasi area pembacaan angka menggunakan *overlay* tetap pada kamera perangkat *mobile*. Area tersebut kemudian dipotong (*cropping*) menjadi beberapa bagian sesuai jumlah digit yang ditentukan, sehingga setiap potongan citra langsung merepresentasikan satu karakter untuk diproses oleh model klasifikasi berbasis CNN.

2.5 Augmentasi Data

Augmentasi data ialah sebuah teknik untuk menambah kuantitas dan variasi *dataset* latih dengan memperkenalkan variasi tambahan melalui berbagai transformasi gambar (Toyib et al., 2024). Dalam pengenalan digit, terdapat berbagai teknik augmentasi yang dapat diimplementasikan untuk meningkatkan variasi data, seperti penyesuaian kecerahan (*brightness adjustment*), perubahan kontras (*contrast adjustment*), penambahan *noise*, *blur*, serta rotasi citra. Penerapan teknik-teknik tersebut bertujuan untuk mensimulasikan berbagai kondisi pengambilan gambar di dunia nyata, seperti perbedaan pencahayaan, sudut pengambilan, maupun kualitas citra, sehingga model yang dilatih menjadi lebih *robust* serta lebih baik dalam melakukan generalisasi terhadap data yang ditemui pada kondisi nyata.

2.5.1 *Brightness Adjustment*

Brightness adjustment merupakan teknik augmentasi data yang dilakukan dengan memodifikasi tingkat kecerahan citra melalui perubahan intensitas piksel sehingga menghasilkan variasi kondisi pencahayaan pada data pelatihan yang membantu model menjadi lebih *robust* (Cossio, 2023). Penyesuaian kecerahan pada citra secara matematis dapat dinyatakan dengan persamaan berikut.

$$I'(x, y) = I(x, y) + \beta \quad (2.3)$$

Keterangan:

$I'(x, y)$: Nilai intensitas piksel setelah dilakukan penyesuaian kecerahan.

$I(x, y)$: Nilai intensitas piksel pada citra asli.

β : Nilai konstanta untuk meningkatkan atau mengurangi tingkat kecerahan.

2.5.2 Contrast Adjustment

Contrast adjustment merupakan teknik augmentasi data yang dilakukan dengan menyesuaikan kontras citra melalui penyesuaian selisih antara piksel paling terang dan paling gelap, sehingga membuat model lebih *robust* terhadap variasi kualitas citra dan derau (Cossio, 2023). Proses tersebut dapat direpresentasikan dalam persamaan berikut.

$$I'(x, y) = \alpha(I(x, y) - \mu) + \mu \quad (2.4)$$

Keterangan:

$I'(x, y)$: Nilai intensitas piksel setelah dilakukan penyesuaian kontras.

$I(x, y)$: Nilai intensitas piksel pada citra asli.

α : Faktor pengali kontras.

μ : Nilai rata-rata intensitas piksel pada citra.

2.5.3 Gaussian Blur

Gaussian blur merupakan teknik augmentasi data yang menghaluskan tekstur citra dengan menerapkan fungsi *Gaussian* melalui operasi konvolusi untuk mensimulasikan degradasi visual seperti kabut, sehingga model menjadi lebih *robust* terhadap variasi citra (Shorten & Khoshgoftaar, 2019). Secara matematis, distribusi *Gaussian* yang digunakan sebagai dasar pembentukan kernel filter dapat dinyatakan dalam persamaan berikut.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (2.5)$$

Keterangan:

$G(x, y)$: Nilai kernel *Gaussian* pada koordinat (x, y) .

x, y : Posisi piksel relatif terhadap pusat kernel.

σ : Standar deviasi distribusi *Gaussian* yang menentukan tingkat efek blur.

e : Konstanta eksponensial ($\approx 2,71828$).

π : Konstanta pi ($\approx 3,14159$).

2.5.4 *Gaussian Noise*

Gaussian noise merupakan teknik augmentasi data yang bekerja dengan menambahkan derau acak dari distribusi *Gaussian* ke citra pelatihan untuk meningkatkan variasi data dan ketahanan model (Shorten & Khoshgoftaar, 2019). Penambahan derau *Gaussian* pada citra secara matematis dapat dinyatakan melalui persamaan berikut.

$$I'(x, y) = I(x, y) + n(x, y) \quad (2.6)$$

Keterangan:

$I'(x, y)$: Nilai intensitas piksel citra setelah ditambahkan derau *Gaussian*.

$I(x, y)$: Nilai intensitas piksel pada citra asli pada koordinat (x, y) .

$n(x, y)$: Nilai derau acak yang mengikuti distribusi *Gaussian*.

2.5.5 *Rotation*

Rotation merupakan teknik augmentasi data yang dilakukan melalui pemutaran acak pada citra untuk mereplikasi variasi kondisi pengambilan gambar. Penerapan augmentasi rotasi berkontribusi dalam meningkatkan kemampuan generalisasi model dan stabilitas performa pada tahap prediksi. (Suci et al., 2026). Transformasi rotasi pada citra secara matematis dapat dinyatakan menggunakan transformasi geometri sebagai berikut.

$$x' = x \cos \theta - y \sin \theta \quad (2.7)$$

$$y' = x \sin \theta + y \cos \theta \quad (2.8)$$

Keterangan:

x, y : Koordinat piksel pada citra asli.

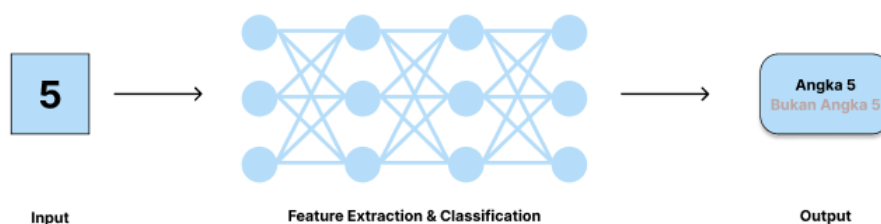
x', y' : Koordinat piksel pada citra setelah dilakukan rotasi.

θ : Sudut rotasi citra terhadap titik pusat.

$\cos \theta$ dan $\sin \theta$: Fungsi trigonometri yang menentukan perubahan posisi piksel.

2.6 Deep Learning

Deep Learning ialah teknik pembelajaran yang mengimplementasikan arsitektur jaringan saraf tiruan yang mengandung banyak lapisan (*multi-layer*). Model *deep learning* dapat mempelajari representasi fitur dengan otomatis dari data mentah tanpa membutuhkan banyak manipulasi fitur secara manual. Metode ini populer digunakan pada banyak bidang seperti klasifikasi citra, pengenalan suara, serta berbagai permasalahan lain dalam *supervised*, *unsupervised*, maupun *reinforcement learning* (Cholissodin et al., 2020). Pembelajaran fitur data dilakukan berbeda antara *Machine Learning* dan *Deep Learning*. Ekstraksi fitur dilakukan secara manual oleh manusia di dalam *Machine Learning* sebelum data diproses oleh algoritma, sedangkan ekstraksi fitur dilakukan secara otomatis di dalam *Deep Learning* oleh lapisan-lapisan jaringan saraf yang mendalam sehingga lebih efektif dalam mengatasi data yang kompleks dan berukuran besar. Jaringan saraf *deep learning* dapat diilustrasikan seperti pada Gambar 2. 1.



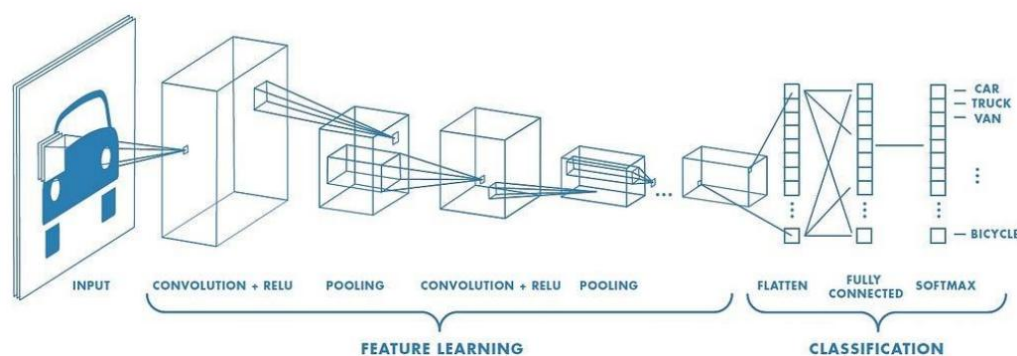
Gambar 2. 1 Jaringan saraf *deep learning*

Gambar tersebut menggambarkan proses kerja jaringan saraf pada *deep learning* yang terpisah menjadi tiga bagian, yaitu *input layer*, *hidden layers*, dan *output layer*. Pada bagian *input*, data mentah berupa angka atau citra dimasukkan ke dalam model. Selanjutnya data diproses melalui beberapa lapisan tersembunyi (*hidden layers*) yang bertugas untuk melakukan ekstraksi fitur dan proses

klasifikasi secara bertahap. Setiap lapisan akan mempelajari pola tertentu dari data sehingga menghasilkan representasi yang semakin kompleks. Setelah proses tersebut selesai, hasil akhir akan dihasilkan pada *output layer* berupa prediksi atau klasifikasi.

2.7 Convolutional Neural Network (CNN)

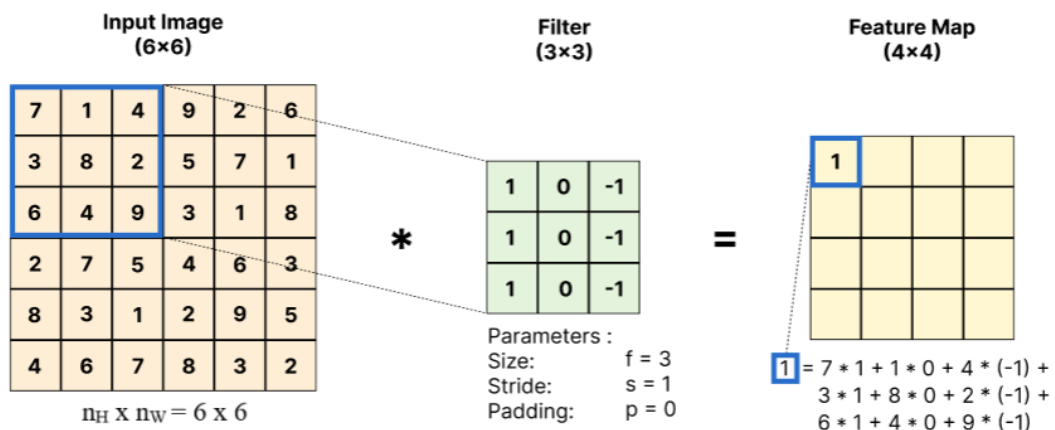
CNN adalah arsitektur *deep learning* yang sangat terkenal dan telah banyak digunakan dalam pemrosesan data visual, terutama untuk tugas pengenalan dan klasifikasi citra. CNN mengadopsi struktur visual *cortex* pada otak manusia dan hewan, di mana sel-sel kompleks membentuk sistem visual yang mampu mengenali pola secara hierarkis (Alzubaidi et al., 2021). Fitur citra dapat otomatis diekstraksi CNN melalui lapisan-lapisan konvolusi yang tersusun secara hierarkis, di mana lapisan awal mengekstraksi fitur tingkat mudah seperti tepi dan tekstur, sementara lapisan akhir mengekstraksi fitur tingkat tinggi yang lebih kompleks dan spesifik (Liu et al., 2021). Secara umum, CNN memproses citra melalui tahapan konvolusi, aktivasi, reduksi dimensi (*pooling*), dan klasifikasi, sehingga menghasilkan representasi fitur yang efektif untuk berbagai tugas pengenalan visual. Arsitektur CNN secara umum dapat dilihat pada Gambar 2. 2.



Gambar 2. 2 Arsitektur CNN secara umum
(Sumber: (Wardani & Leonardi, 2023))

Berdasarkan Gambar 2. 2, arsitektur CNN terdiri atas dua komponen utama, yaitu *feature learning* dan *classification* (Lukman & Fuad, 2026). *Feature Learning* merupakan tahap yang berguna untuk mengekstraksi dan mempelajari representasi fitur citra *input* melalui proses konvolusi, aktivasi dan *pooling*. Pada tahap ini, CNN secara bertahap membangun fitur dari sederhana (misalnya tepi dan tekstur) hingga fitur kompleks (Zeiler & Fergus, 2013). Pada dasarnya, *Feature Learning* terdiri dari dua lapisan, yaitu *convolutional layer* bertugas mengekstraksi pola atau ciri spesifik dari citra, serta *pooling layer* yang berfungsi mereduksi dimensi fitur sambil tetap menjaga informasi penting guna meningkatkan efisiensi komputasi dan mengurangi risiko *overfitting*.

Pada *convolutional layer*, proses konvolusi dijalankan dengan menggeser kernel pada citra *input*, kemudian mengalikan elemen per elemen dan menjumlahkannya supaya menghasilkan *feature map* baru sebagai representasi fitur (Zhao et al., 2024). Berikut contoh proses konvolusi pada lapisan ini.



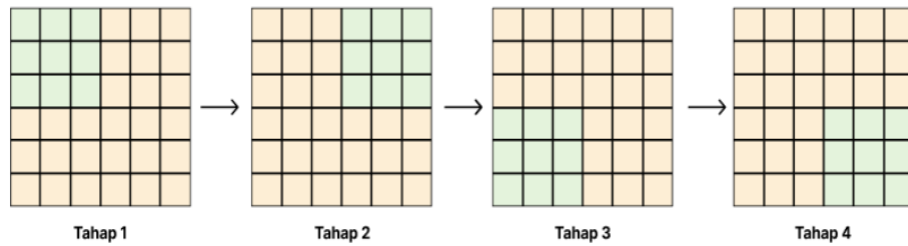
Gambar 2. 3 Proses konvolusi

Berdasarkan Gambar 2. 3 di atas, dapat diketahui bahwa proses konvolusi dimulai dengan menempatkan filter pada region 3×3 pertama di pojok kiri atas

input image (ditandai dengan kotak biru), kemudian setiap elemen dalam region tersebut dikalikan dengan elemen filter yang bersesuaian dan nilai satu *output* dihasilkan dari penjumlahan hasil perkalian yang ditempatkan di posisi pertama *feature map*. Filter kemudian bergeser ke kanan sebesar satu piksel (*stride* = 1) dan proses perhitungan diulang, pergeseran ini terus berlanjut secara horizontal hingga mencapai tepi kanan, lalu filter turun satu baris dan kembali dimulai dari sisi kiri hingga seluruh area *input* telah di-*scan* dan menghasilkan *feature map* lengkap berukuran 4×4 . Terdapat beberapa parameter penting dalam konvolusi yang perlu diperhatikan, yakni *stride* dan *padding* yang akan mempengaruhi ukuran dan karakteristik dari *feature map* yang dihasilkan.

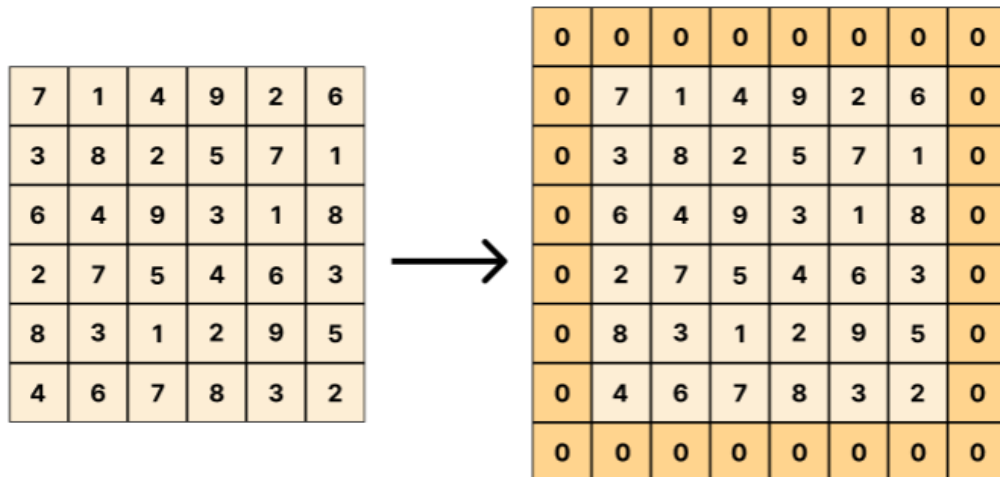
1. *Stride*

Stride adalah angka yang menentukan seberapa banyak filter bergeser saat melakukan operasi konvolusi atau *pooling*, yang berfungsi untuk mengurangi ukuran *output* secara bertahap. *Stride* merupakan parameter penting yang mengontrol tingkat pengurangan resolusi gambar (Riad et al., 2022). Nilai *stride* dapat bervariasi sesuai dengan kebutuhan arsitektur, di mana *stride* dengan nilai 1 mengindikasikan bahwa filter bergeser sebanyak satu piksel pada setiap langkah, sedangkan *stride* dengan nilai yang lebih besar seperti 2 atau 3 akan menghasilkan pergeseran yang lebih signifikan. Berikut contoh proses konvolusi dengan menggunakan nilai *stride* 3.

Gambar 2. 4 Proses konvolusi dengan nilai *stride* 3

2. *Padding*

Padding merupakan parameter di CNN yang berfungsi untuk penambahan baris dan kolom nol di sekitar batas luar *input feature map* sebelum operasi konvolusi dilakukan. Tujuannya adalah untuk mempertahankan ukuran *feature map* tetap sama sebelum dan sesudah konvolusi, serta untuk mempertahankan fitur-fitur yang berada di tepi citra agar tidak hilang dalam proses (Zhang et al., 2024). Contoh penambahan *padding* pada *matrix input* ditunjukkan pada Gambar 2. 5.

Gambar 2. 5 Penambahan *padding*

Proses konvolusi akan menghasilkan satu atau lebih *feature map*, di mana kumpulan *feature map* tersebut membentuk *convolution layer*. Untuk menentukan

ukuran dimensi *output* (*feature map*) setelah proses konvolusi, dapat digunakan persamaan berikut:

$$Output = \frac{W - N + 2P}{S} + 1 \quad (2.9)$$

Keterangan:

W : Panjang atau tinggi *input*.

N : Panjang atau tinggi filter.

P : *Zero padding*.

S : *Stride*.

Setelah proses konvolusi selesai menghasilkan *feature map*, maka berikutnya adalah penggunaan fungsi aktivasi untuk menambahkan sifat *non-linear* ke dalam jaringan. Fungsi aktivasi berperan penting dalam mentransformasikan informasi dari satu lapisan ke lapisan berikutnya sehingga model mampu belajar pola kompleks dari data citra dan tidak hanya terbatas pada pemetaan *linear*. fungsi aktivasi diperlukan agar jaringan saraf dapat menangkap fitur kompleks dan kemampuan generalisasi model bertambah (Azhar & Ismail, 2024). ReLU sering digunakan sebagai fungsi aktivasi pada arsitektur CNN. ReLU bekerja dengan mengganti nilai negatif menjadi nol sementara nilai positif dipertahankan tetap apa adanya, sehingga meningkatkan efisiensi komputasi serta mampu membantu meminimalkan permasalahan *vanishing gradient* (Mesran et al., 2024). Secara matematis, fungsi aktivasi ReLU dapat dirumuskan sebagai berikut:

$$f(x) = \max(0, x) \quad (2.10)$$

yang berarti bahwa nilai *input* yang bernilai negatif akan dikonversi menjadi nol, sementara nilai positif tetap dipertahankan tanpa perubahan.

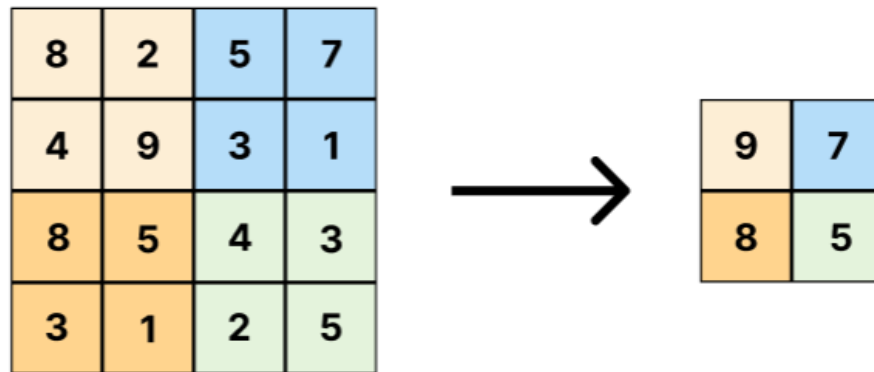
Setelah proses aktivasi dilakukan, tahap berikutnya dalam arsitektur CNN adalah *pooling layer*. *Pooling* berfungsi untuk mereduksi dimensi (*downsampling*) *feature map* yang telah melalui proses konvolusi dan aktivasi. Tujuan utama dari *pooling* ialah untuk mereduksi jumlah parameter dan kompleksitas komputasi tanpa menghilangkan informasi penting dari fitur yang telah dipelajari. Selain itu, *pooling* juga membantu meningkatkan kemampuan model dalam menghadapi variasi seperti translasi atau perubahan posisi objek pada citra *input* karena menghasilkan representasi fitur yang lebih stabil. *Pooling layer* bekerja dengan cara mengurangi ukuran (dimensi) *feature map* dengan mengambil nilai tertentu (misalnya nilai maksimum) dari setiap bagian kecil citra sehingga mempertahankan fitur penting sekaligus menurunkan kompleksitas komputasi (Azmi et al., 2023). Ukuran *output* yang dihasilkan dari proses *pooling* dapat dihitung menggunakan rumus berikut:

$$Output = \frac{(W - F)}{S} + 1 \quad (2.11)$$

Keterangan:

W : Ukuran *input*.
 F : Ukuran filter *pooling*.
 S : Nilai *stride*.

Max pooling adalah jenis *pooling* sering digunakan, di mana nilai tertinggi dalam suatu region dipilih dan dimasukkan ke *feature map* berikutnya. Contoh operasi *max pooling* dapat dilihat pada Gambar 2. 6.



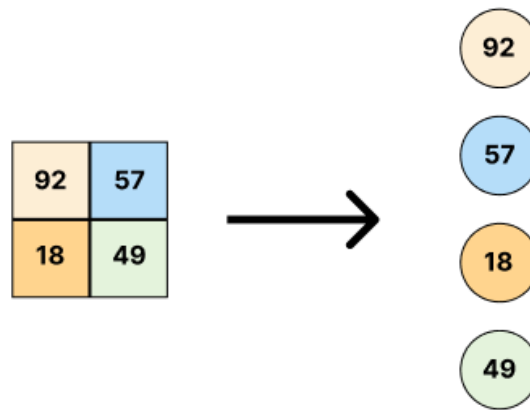
Gambar 2. 6 Max pooling

Berdasarkan Gambar 2. 6, proses *max pooling* dilakukan menggunakan ukuran jendela (*pooling window*) 2×2 dengan nilai *stride* sebesar 2, sehingga filter bergerak tanpa saling tumpang tindih antar region. Pada setiap pergeseran jendela *pooling*, dipilih nilai tertinggi dari empat elemen yang berada di dalam region tersebut untuk direpresentasikan sebagai *output*. Sebagai contoh, pada region pertama berisi angka $\{8, 2, 4, 9\}$, nilai tertingginya adalah 9 sehingga nilai tersebut dimasukkan ke posisi pertama pada *feature map* hasil *pooling*. Proses yang sama diterapkan pada region berikutnya hingga seluruh area matriks terliput. Melalui proses ini, ukuran *feature map* berkurang dari 4×4 menjadi 2×2 , namun fitur yang paling dominan tetap dipertahankan, sehingga menghasilkan representasi yang lebih ringkas dan efisien tanpa terjadi kehilangan informasi penting.

Setelah melalui tahap *feature learning*, arsitektur CNN secara umum dilanjutkan pada komponen *classification* yang berfungsi untuk melakukan proses pemetaan fitur menjadi kelas *output* tertentu. Tahap klasifikasi terdiri dari beberapa lapisan utama, yaitu *Flatten layer*, *Fully Connected layer*, serta *Softmax layer*.

1. *Flatten*

Lapisan *flatten* berfungsi mengganti keluaran *feature map* hasil proses konvolusi dan *pooling* di mana masih berbentuk matriks multidimensi ke bentuk vektor satu dimensi (Ulum et al., 2024). Hal ini bertujuan agar data dapat diproses pada tahap klasifikasi, karena *input* untuk *Fully Connected layer* hanya dapat dalam bentuk vektor satu dimensi. Berikut adalah ilustrasi proses *flatten*.

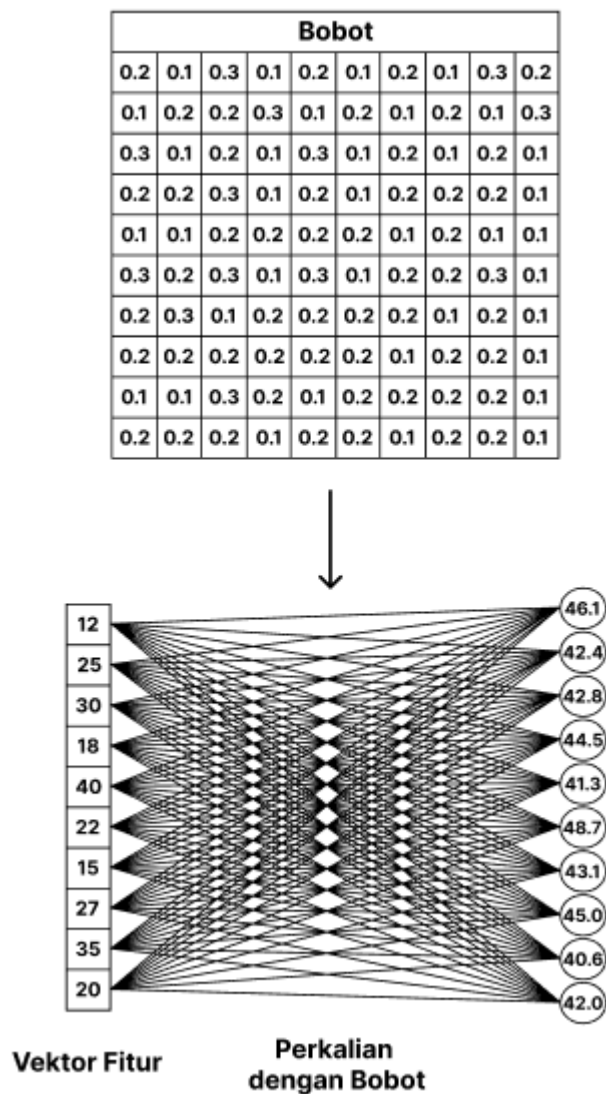


Gambar 2. 7 Proses *flatten*

2. *Fully Connected Layer*

Lapisan *fully connected* berfungsi untuk melakukan transformasi *linear* terhadap vektor fitur hasil ekstraksi untuk memetakan representasi fitur ke dalam ruang klasifikasi (Hardi & Sundari, 2022). Pada tahap ini, setiap neuron pada *fully connected layer* terhubung dengan semua elemen pada vektor fitur yang dihasilkan dari proses *flatten*. Vektor fitur tersebut merupakan hasil representasi dari berbagai fitur penting di mana sebelumnya telah diekstraksi oleh lapisan konvolusi dan diperkecil dimensinya melalui proses *pooling*. Dengan demikian, lapisan *fully connected* dapat memanfaatkan seluruh

informasi fitur yang telah diperoleh pada tahap sebelumnya untuk melakukan proses klasifikasi. Proses ini dilakukan dengan mengalikan setiap elemen pada vektor fitur dengan bobot (*weight*) yang sudah dipelajari selama proses pelatihan, lalu menjumlahkan hasil perkalian tersebut dan menambahkan nilai bias untuk menghasilkan nilai keluaran pada setiap neuron. Berikut adalah contoh ilustrasi proses *fully connected layer* dengan vektor input sebanyak 10.



Gambar 2. 8 Proses *Fully Connected Layer*

3. *Softmax*

Softmax berfungsi menghasilkan nilai probabilitas pada setiap kelas yang tersedia dengan cara mengganti nilai *output* dari proses sebelumnya (Nusantara & Akbar, 2024). Fungsi ini menghitung probabilitas dari setiap kelas sehingga total probabilitasnya bernilai 1, sehingga kelas dengan nilai probabilitas tertinggi akan ditentukan model sebagai hasil klasifikasi.

Rumus dari fungsi *softmax* digunakan untuk mengubah nilai *logit* atau skor keluaran dari jaringan menjadi probabilitas pada setiap kelas. Secara matematis, fungsi softmax dapat dinyatakan sebagai:

$$softmax(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^n \exp(x_j)} \quad (2.12)$$

Keterangan:

x_i	: Nilai output (<i>logit</i>) pada kelas ke- i .
$\exp(x_i)$	: Fungsi eksponensial dari nilai x_i .
n	: Jumlah total kelas pada proses klasifikasi.
$\sum_{j=1}^n \exp(x_j)$	: Jumlah seluruh nilai eksponensial dari setiap output kelas.

Melalui proses tersebut, nilai *output* akan dikonversi menjadi distribusi probabilitas dengan total nilai sama dengan 1, sehingga nilai probabilitas terbesar pada sebuah kelas dapat dipilih menjadi hasil prediksi model.

Berdasarkan Gambar 2. 8 sebelumnya, hasil perkalian antara vektor fitur dengan bobot pada *fully connected layer* menghasilkan sejumlah nilai *output* yang menggambarkan skor untuk setiap kelas digit yang diprediksi. Pada ilustrasi tersebut terdapat 10 nilai *output* yang merepresentasikan kelas digit 0 sampai 9, yaitu 46.1, 42.4, 42.8, 44.5, 41.3, 48.7, 43.1, 45.0, 40.6, dan 42.0. Nilai-nilai tersebut kemudian diproses dengan *softmax* supaya dikonversi

menjadi distribusi probabilitas pada setiap kelas. Dari nilai output tersebut, skor tertinggi adalah 48.7 yang berada pada kelas ke-6 (digit 5 jika indeks dimulai dari 0). Oleh karena itu, setelah melalui proses *softmax*, kelas dengan probabilitas tertinggi adalah digit 5, sehingga model memprediksi bahwa input yang diberikan merupakan digit angka 5.

2.8 MobileNetV2

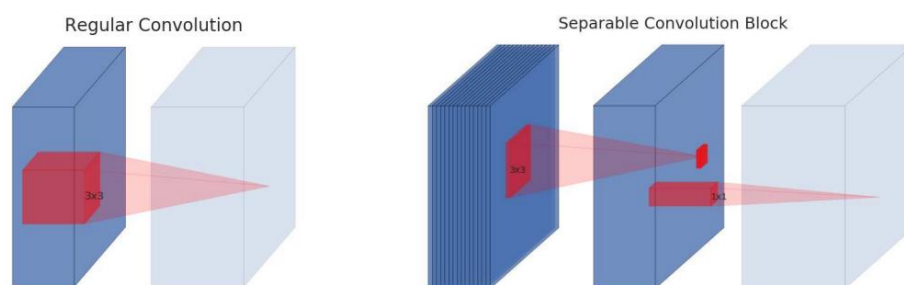
MobileNetV2 merupakan sebuah model *deep learning* yang sangat efisien dan didesain khusus untuk digunakan pada aplikasi perangkat seluler serta sistem tertanam (*embedded*). Model ini dikembangkan oleh Sandler, Howard, Zhu, Zhmoginov, dan Chen yang dipublikasikan pada tahun 2018 melalui konferensi IEEE mengenai visi komputer dan pengenalan pola. Dasar dikembangkannya arsitektur ini adalah untuk mengatasi hambatan komputasi dan kebutuhan akan unit pemrosesan grafis (GPU) yang besar, sehingga memungkinkan penerapan algoritma pembelajaran mesin sebagai alat diagnostik yang akurat dan cepat di lingkungan dengan keterbatasan sumber daya (Akay et al., 2021).

Karakteristik utama yang membedakan arsitektur MobileNetV2 dengan model pendahulunya terletak pada penggunaan struktur blok inovatif yang mengoptimalkan efisiensi tanpa mengurangi akurasi. Struktur ini mengandalkan tiga komponen teknis penting, yaitu *Depthwise Separable Convolution*, *Inverted Residual Block*, dan *Linear Bottleneck* (Sandler et al., 2018).

2.8.1 *Depthwise Separable Convolution*

Depthwise Separable Convolution (DSC) merupakan konsep konvolusi yang digunakan oleh MobileNetV2 di mana mekanisme ini bekerja dengan memecah operasi konvolusi standar menjadi dua tahap terpisah, yaitu konvolusi *depthwise* dan konvolusi *pointwise* (Sandler et al., 2018). Melalui pemisahan ini, DSC bisa mereduksi jumlah parameter dan kompleksitas komputasi dalam jumlah yang besar tanpa mengorbankan performa model.

Pada tahap *depthwise*, satu filter tunggal diterapkan ke setiap *channel* input secara mandiri untuk mengumpulkan informasi spasial, yang kemudian diikuti oleh konvolusi *pointwise* (1×1) untuk mengkombinasikan informasi antar *channel* tersebut menjadi representasi baru. Supaya mendapatkan pemahaman yang lebih mengenai *depthwise separable convolution*, berikut contoh ilustrasi perbandingannya dengan normal *convolution*.



Gambar 2. 9 Perbandingan konvolusi normal dan konvolusi *depthwise separable*
(Sumber: (Sandler et al., 2018))

Berdasarkan gambar Gambar 2. 9, dapat diketahui bahwa konvolusi standar menggunakan kernel berukuran 3×3 yang langsung bekerja pada seluruh channel input secara bersamaan untuk menghasilkan satu *feature map*. Sebaliknya, konvolusi *depthwise separable* memisahkan proses tersebut menjadi dua tahap,

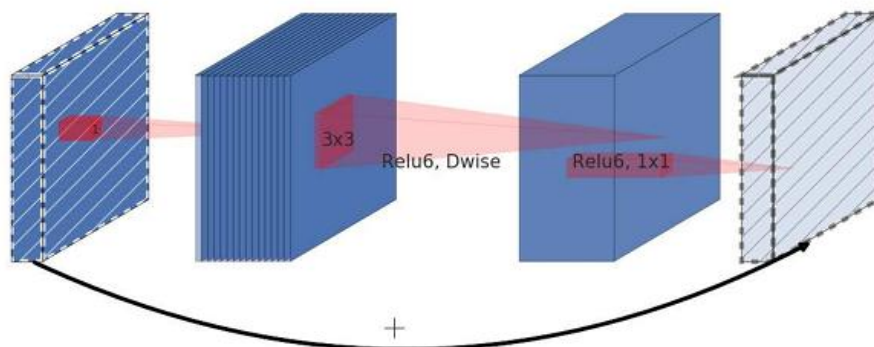
yaitu *depthwise convolution* yang melakukan ekstraksi fitur spasial pada setiap channel secara terpisah, kemudian dilanjutkan dengan *pointwise convolution* berukuran 1×1 yang berfungsi mengombinasikan informasi antar *channel*.

2.8.2 *Inverted Residual Block*

Inverted Residual Block merupakan salah satu modul utama pada arsitektur MobileNetV2 yang didesain untuk menambah efisiensi komputasi sekaligus mempertahankan kemampuan representasi fitur. Pada blok ini digunakan konsep *residual connection* atau *skip connection* yang menghubungkan bagian awal dan akhir dari suatu blok konvolusi untuk membantu aliran informasi menuju lapisan yang lebih dalam. Berbeda dengan *residual block* pada arsitektur konvensional yang biasanya memiliki jumlah *channel* lebih besar di awal dan akhir blok, *inverted residual block* justru memiliki jumlah *channel* yang lebih sedikit pada bagian yang terhubung dan lebih banyak pada lapisan di tengah. Pendekatan ini memungkinkan pengurangan kuantitas parameter dan kebutuhan memori dalam jumlah yang besar tanpa mengurangi kemampuan jaringan dalam mengekstraksi fitur krusial citra (Sandler et al., 2018).

Mekanisme kerja *inverted residual block* dimulai dengan melakukan ekspansi dimensi *channel* menggunakan konvolusi 1×1 (*pointwise convolution*) sehingga jumlah *channel* fitur meningkat. Setelah itu dijalankan *depthwise convolution* berdimensi 3×3 yang bertujuan untuk mengekstraksi informasi spasial pada setiap *channel* secara terpisah dengan proses yang lebih efisien. Selanjutnya, fitur yang telah diproses diproyeksikan kembali ke dimensi *channel* yang lebih kecil melalui konvolusi 1×1 untuk mendapatkan representasi fitur yang lebih ringkas.

Pada tahap proyeksi ini diterapkan konsep *linear bottleneck*, yaitu tanpa menerapkan fungsi aktivasi *non-linear*, sehingga informasi penting pada ruang fitur berdimensi rendah dapat tetap dipertahankan. Selain itu, koneksi residual atau *shortcut* menghubungkan input dengan *output* pada bagian *bottleneck* ketika dimensi fitur sesuai, sehingga memungkinkan aliran informasi yang lebih efisien serta membantu mengurangi permasalahan degradasi pada jaringan yang lebih dalam (Sandler et al., 2018). *Inverted residual block* dapat diilustrasikan pada Gambar 2. 10.



Gambar 2. 10 *Inverted residual block*
(Sumber: (Sandler et al., 2018))

2.8.3 *Linear Bottleneck*

Linear Bottleneck merupakan konsep pada arsitektur MobileNetV2 yang digunakan untuk mempertahankan informasi fitur penting ketika proses reduksi dimensi *channel* dilakukan. Pada lapisan ini, tidak digunakan fungsi aktivasi *non-linear* pada tahap proyeksi akhir sehingga hubungan *linear* antar fitur tetap terjaga dan informasi yang berada pada ruang berdimensi rendah tidak mengalami kehilangan signifikan. Pendekatan ini bertujuan untuk menghindari hilangnya

representasi fitur akibat transformasi *non-linear* pada ruang fitur yang sempit (Sandler et al., 2018).

2.8.4 Arsitektur

Arsitektur MobileNetV2 didesain untuk mendapatkan model jaringan saraf dengan ukuran kecil namun tetap mampu melakukan ekstraksi fitur secara efektif pada *device* dengan keterbatasan sumber daya. Secara dasar, arsitektur MobileNetV2 memiliki tiga bagian utama, yaitu *input layer*, *feature extraction layer*, dan *classification layer*. *Input layer* sebagai penerima citra masukan, *feature extraction layer* sebagai pengekstraksi fitur-fitur krusial citra melalui serangkaian *bottleneck block*, sedangkan *classification layer* digunakan untuk menghasilkan keluaran sesuai dengan kuantitas kelas yang ditetapkan. Susunan lapisan pada arsitektur umum MobileNetV2 ditunjukkan pada Gambar 2. 11.

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-

Gambar 2. 11 Arsitektur umum MobileNetV2
(Sumber: (Sandler et al., 2018))

2.8.4.1 *Input Layer*

Input layer merupakan lapisan pertama pada arsitektur MobileNetV2 sebagai penerima citra *input*. Secara umum, MobileNetV2 menggunakan ukuran *input* standar berukuran 224×224 piksel dengan tiga *channel* warna RGB sehingga dimensi *input* yang digunakan adalah $224 \times 224 \times 3$ (Lestari et al., 2024). Ukuran tersebut telah ditetapkan pada arsitektur asli MobileNetV2 dan digunakan pada proses pelatihan model menggunakan *dataset ImageNet*.

2.8.4.2 *Feature Extraction Layer*

Feature extraction layer adalah bagian utama dari MobileNetV2 yang bertugas mengekstraksi informasi dan pola penting dari citra masukan. Proses ekstraksi fitur diawali dengan lapisan *convolution* 2D yang menghasilkan *feature map* awal, kemudian dilanjutkan dengan serangkaian *bottleneck block* yang menerapkan konsep *inverted residual* dan *linear bottleneck*. Pada setiap *bottleneck block* dilakukan ekspansi *channel* menerapkan konvolusi 1×1 , dilanjutkan dengan *depthwise convolution* 3×3 untuk mengekstraksi informasi spasial, kemudian dilakukan proyeksi kembali ke jumlah *channel* yang lebih kecil. Beberapa *bottleneck block* diulang sebanyak n kali dengan konfigurasi tertentu yang meliputi faktor ekspansi (t), jumlah *channel* keluaran (c), dan nilai *stride* (s) untuk menghasilkan representasi fitur yang semakin kompleks.

2.8.4.3 *Classification Layer*

Setelah proses ekstraksi fitur selesai dilakukan, *feature map* diteruskan ke *classification layer*. Pada tahap ini digunakan konvolusi 1×1 untuk meningkatkan

kuantitas *channel* fitur menjadi 1280 *channel*, kemudian dilakukan proses *global average pooling* guna menurunkan dimensi spasial menjadi vektor fitur berukuran $1 \times 1 \times 1280$. Selanjutnya, fitur yang telah diringkas digunakan oleh lapisan klasifikasi untuk menghasilkan keluaran sesuai dengan jumlah kelas yang ditentukan.

2.9 TensorFlow dan TensorFlow Lite

TensorFlow (TF) adalah sistem *machine learning* berskala besar yang menggunakan graf aliran data (*data-flow graphs*) untuk mendefinisikan operasi, keadaan bersama (*shared state*), dan komputasi yang mengubah keadaan tersebut. Framework ini dikembangkan oleh Google dan dirancang untuk pelatihan jaringan saraf dalam (*deep neural networks*) serta inferensi dalam berbagai platform, mulai dari CPU, GPU, hingga TPU (Ramchandani et al., 2022). Di sisi lain, *TensorFlow Lite* (*tflite*) adalah varian ringan dari *framework TensorFlow* yang didesain khusus untuk *device* dengan sumber daya komputasi terbatas. *TensorFlow Lite* berfungsi sebagai platform kecerdasan komputasi untuk melakukan inferensi secara lokal (*on-device inference*) (Ario & Prasetyo, 2017).

Perbedaan utama antara *TensorFlow* dan *TensorFlow Lite* terletak pada tujuan penggunaan, arsitektur, dan lingkungan eksekusinya. *TensorFlow* didesain untuk pembangunan dan pelatihan model *deep learning* berskala penuh dengan dukungan komputasi intensif pada server, *workstation*, atau infrastruktur *cloud* yang memiliki sumber daya tinggi seperti GPU dan TPU. Sebaliknya, *TensorFlow Lite* tidak mendukung proses pelatihan (*training*) dan hanya difokuskan pada inferensi (*inference*) di perangkat dengan keterbatasan sumber daya. Untuk

mencapai efisiensi tersebut, *TFLite* menggunakan teknik seperti kuantisasi, optimasi graf, serta representasi model dalam format khusus (*.tflite*) yang lebih ringkas, sehingga memungkinkan model berjalan cepat dan hemat daya di perangkat *edge* seperti ponsel atau mikrokontroler.

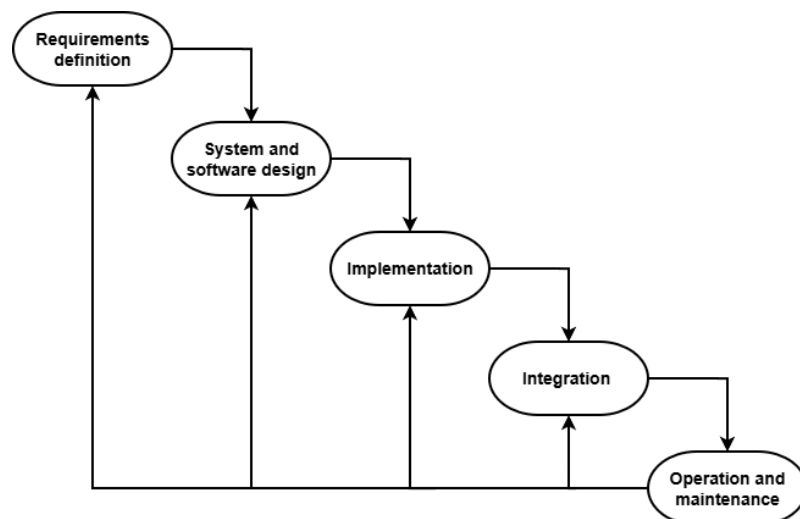
2.10 Integrasi Model OCR dalam Aplikasi *Mobile*

Integrasi model OCR yang telah dibangun diperlukan supaya model bisa diimplementasikan secara langsung oleh *user* melalui aplikasi *mobile*. Aplikasi *mobile* dalam penelitian ini dikembangkan menggunakan *framework* React Native dengan bantuan Expo sebagai *toolchain* untuk mempermudah proses pengembangan aplikasi lintas platform. Selain itu, sistem *backend* memanfaatkan *Convex* sebagai *backend-as-a-service* yang menawarkan layanan basis data reaktif dan komunikasi *realtime*, sehingga mendukung sinkronisasi data secara efisien antara aplikasi dan server.

Tahap integrasi ini menjadi penting karena menghubungkan model yang dibangun di lingkungan pengembangan dengan aplikasi yang berjalan di perangkat *mobile* yang memiliki keterbatasan memori, prosesor, dan konsumsi daya. Untuk memastikan performa yang optimal, model OCR perlu dikonversi terlebih dahulu ke dalam format (*.tflite*), yang dirancang untuk menghasilkan model dengan ukuran kecil serta proses inferensi yang cepat pada *device* yang memiliki keterbatasan sumber daya. Pendekatan ini merujuk pada (Lehtonen, 2024), yang menjelaskan bahwa integrasi model *TensorFlow Lite* ke dalam aplikasi *cross-platform* dapat dilakukan melalui pemanggilan modul *native* dari lingkungan *JavaScript*.

2.11 *Waterfall Model*

Salah satu model *System Development Life Cycle* (SDLC) adalah *Waterfall*. Model ini menerapkan alur pengembangan yang bersifat linear. Karakteristik tersebut menjadikan *Waterfall* sebagai metode yang memiliki proses pengembangan sederhana dan sistematis karena pelaksanaan setiap fase bergantung pada penyelesaian fase sebelumnya. Salah satu keunggulan metode *Waterfall* terletak pada alur pengembangannya yang jelas dan sistematis, sehingga setiap tahapan mudah dipahami (Ramadhan et al., 2023). Tahapan *Waterfall* ditunjukkan pada Gambar 2. 12.



Gambar 2. 12 Tahapan pengembangan perangkat lunak model *Waterfall*

Setiap tahap pada model pengembangan perangkat lunak *Waterfall* mempunyai peranan penting untuk memastikan proses pembangunan aplikasi berjalan secara sistematis, terstruktur, dan sesuai kebutuhan pengguna. Tahapan pengembangan yang diterapkan dalam penelitian ini adalah sebagai berikut:

1. *Requirements Definition*

Tahap ini berfokus pada pengumpulan dan penentuan kebutuhan sistem. Hasil identifikasi kebutuhan didapatkan dengan cara mengobservasi dan melakukan studi literatur untuk mengetahui kebutuhan fungsional dan non-fungsional sistem yang akan dikembangkan. Hasil tersebut digunakan sebagai acuan dalam proses tahap selanjutnya.

2. *System and Software Design*

Tahap ini bertujuan merancang solusi teknis dengan dasar kebutuhan yang telah ditetapkan. Perancangan mencakup struktur arsitektur sistem, basis data, antarmuka pengguna (UI), dan alur proses bisnis. Desain sistem harus mampu menggambarkan bagaimana sistem akan dibangun secara teknis untuk memenuhi spesifikasi kebutuhan yang sudah ditetapkan sebelumnya.

3. *Implementation*

Tahap ini merupakan proses pengembangan sistem dengan menerjemahkan hasil perancangan untuk diimplementasikan dalam bentuk kode menggunakan bahasa pemrograman serta *framework* yang dipilih. Pengembang menulis kode untuk setiap modul atau komponen sistem sesuai dengan spesifikasi desain.

4. *Integration*

Pada tahap ini, semua komponen sistem yang sudah dikembangkan secara terpisah digabungkan menjadi satu kesatuan aplikasi yang utuh. Proses ini memastikan bahwa setiap modul dapat berinteraksi satu sama lain dengan baik sesuai dengan arsitektur yang direncanakan. Hasil dari integrasi ini menjadi

versi final sistem yang siap untuk diimplementasikan atau diterapkan kepada *user*.

5. *Operation and Maintenance*

Tahap ini berlangsung ketika sistem sudah digunakan dalam lingkungan operasional nyata. Pemeliharaan dijalankan guna mengatasi kesalahan yang terjadi setelah implementasi (*corrective maintenance*), menyesuaikan sistem dengan perubahan kebutuhan atau lingkungan (*adaptive maintenance*), serta melakukan peningkatan fitur dan performa (*perfective maintenance*) guna menjaga kinerja dan keandalan sistem dalam jangka panjang.

2.12 Evaluasi Model

Proses evaluasi model diperlukan guna mengukur performa model yang telah dikembangkan. Tahap ini bertujuan untuk mengukur tingkat kemampuan model dalam menghasilkan prediksi yang tepat dan dapat diandalkan (Lathif & Novianto, 2025). Performa model dievaluasi melalui sejumlah metrik evaluasi, antara lain *Precision*, *Accuracy*, *Recall*, dan *F1-Score*. Keempat metrik tersebut digunakan guna menilai performa model dalam ketepatannya dalam melakukan klasifikasi. Berikut adalah nilai dari setiap metrik evaluasi tersebut:

1. *Precision*

Presisi (*precision*) adalah salah satu metrik evaluasi yang menunjukkan tingkat ketepatan model ketika mengidentifikasi kelas positif. Nilai ini diperoleh dari perbandingan antara jumlah prediksi positif yang benar (*True Positive/ tp*) dengan seluruh data yang diprediksi sebagai positif (*True Positive/ tp* dan *False Positive/ fp*). Nilai presisi yang tinggi menunjukkan

bahwa model memiliki tingkat kesalahan prediksi positif (*False Positive/ fp*) yang rendah (Jain et al., 2020).

$$Precision = \frac{tp}{tp + fp} \quad (2.13)$$

2. Accuracy

Akurasi (*accuracy*) adalah metrik evaluasi untuk mengukur ketepatan model dalam melakukan prediksi, yaitu perbandingan antara jumlah prediksi benar (*True Positive/ tp* dan *True Negative/ tn*) dengan total seluruh data yang dievaluasi (*True Positive/ tp*, *True Negative/ tn*, *False Positive/ fp*, dan *False Negative/ fn*) (Hicks et al., 2022).

$$Accuracy = \frac{tp + tn}{tn + tp + fp + fn} \quad (2.14)$$

3. Recall

Recall digunakan untuk menilai kecakapan model dalam mengenali semua data yang benar-benar tergolong ke dalam kelas positif, yaitu perbandingan antara jumlah prediksi positif yang benar (*True Positive/ tp*) dengan seluruh data yang sebenarnya positif (*True Positive/ tp* dan *False Negative/ fn*). Nilai *recall* yang tinggi mengindikasikan bahwa model mempunyai kemampuan yang baik dalam mendeteksi data positif (Jang et al., 2020).

$$Recall = \frac{tp}{tp + fn} \quad (2.15)$$

4. *F1-Score*

F1-Score digunakan untuk mengevaluasi performa model melalui kombinasi antara nilai *precision* dan *recall* yang dihitung menggunakan rata-rata harmonik. Metrik ini mencerminkan keseimbangan antara ketepatan prediksi terhadap kelas positif dan kemampuan model dalam mengidentifikasi seluruh data positif (Grandini et al., 2020).

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.16)$$

2.13 Akurasi OCR Secara Keseluruhan

Akurasi OCR secara keseluruhan digunakan untuk mengukur kecakapan model saat mengenali karakter pada sistem OCR. Pengukuran dilakukan dengan menghitung persentase karakter yang berhasil dikenali dengan benar dibandingkan dengan total seluruh karakter yang diuji (Darpito et al., 2025). Semakin tinggi nilai akurasi, maka semakin baik performa model dalam melakukan pengenalan karakter secara tepat. Perhitungan akurasi OCR secara keseluruhan dapat dirumuskan sebagai berikut:

$$Akurasi\ OCR = \frac{Jumlah\ karakter\ yang\ dikenali\ dengan\ benar}{Total\ jumlah\ karakter} \times 100\% \quad (2.17)$$

Keterangan:

Jumlah karakter yang dikenali dengan benar adalah jumlah karakter yang prediksinya sesuai dengan nilai asli.

Total jumlah karakter adalah keseluruhan karakter yang diuji.

Nilai akurasi OCR dinyatakan dalam persentase (%) dengan rentang 0-100%.

2.14 *Blackbox Testing*

Metode pengujian perangkat lunak yang berkonsentrasi ke fungsionalitas sistem tanpa memerlukan pengetahuan tentang logika internal, kode sumber, atau implementasi teknis dari sistem yang diuji adalah *blackbox testing*. Pengujian dilakukan dari sudut pandang pengguna eksternal, sehingga penguji tidak perlu memiliki akses terhadap kode program atau arsitektur internal sistem. Metode ini sangat cocok untuk menguji apakah sistem telah berhasil memenuhi kebutuhan fungsional yang sudah ditetapkan. *Blackbox testing* dapat membantu menemukan kemungkinan masalah yang akan terjadi ketika pengguna menggunakan sistem, seperti kesalahan dalam fungsi, *bug*, atau *error* yang berkaitan dengan interaksi antarmuka. Selain itu, metode ini juga membantu *developer* untuk memastikan bahwa aplikasi bisa berjalan dengan lancar berserta responsif, sehingga pengalaman yang baik bagi pengguna dapat tercapai (Ariyana et al., 2023).

Dalam penerapannya pada aplikasi *mobile OCR*, *blackbox testing* digunakan untuk memastikan bahwa seluruh alur penggunaan mulai dari proses pemindaian teks, pengolahan hasil OCR, hingga penyimpanan data ke menu riwayat (*history*) berjalan dengan lancar sesuai kebutuhan pengguna. Pengujian dilakukan dengan mengoperasikan aplikasi sebagaimana pengguna akhir menggunakannya, seperti mengakses menu pemindaian, membuka halaman hasil, meninjau daftar riwayat, serta memanfaatkan menu-menu pendukung lainnya. Melalui pendekatan ini, pengembang dapat menilai apakah setiap komponen antarmuka bekerja dengan benar, responsif, dan mudah dipahami tanpa perlu mengetahui struktur internal sistem.

Untuk mengetahui tingkat keberhasilan pengujian yang dilakukan, hasil pengujian blackbox testing dapat dihitung menggunakan persentase keberhasilan pengujian sebagai berikut.

$$\text{Persentase Keberhasilan} = \frac{\text{Jumlah Pengujian Berhasil}}{\text{Total Kasus Pengujian}} \times 100\% \quad (2.18)$$

Keterangan:

Persentase Keberhasilan adalah nilai yang menunjukkan tingkat keberhasilan fungsi sistem berdasarkan pengujian yang dilakukan.

Jumlah Pengujian Berhasil adalah jumlah skenario pengujian yang berjalan sesuai dengan hasil yang diharapkan.

Total Kasus Pengujian adalah jumlah keseluruhan skenario pengujian yang dilakukan pada sistem.

BAB III

DESAIN DAN IMPLEMENTASI

3.1 Desain Penelitian

Pada penelitian ini digunakan desain penelitian yang bertujuan untuk mengembangkan aplikasi *mobile* OCR dalam pembacaan meteran air pada PAMDes Pandanrejo. Desain penelitian ini menggunakan pendekatan segmentasi digit yang dikombinasikan dengan model MobileNetV2 untuk mengenali angka pada meteran air.

3.1.1 Tahapan Penelitian

Pada bagian ini dijelaskan tahapan penelitian dalam pengembangan aplikasi *mobile* OCR untuk pembacaan meteran air pada PAMDes Pandanrejo menggunakan pendekatan segmentasi digit dengan model CNN berbasis arsitektur MobileNetV2. Tahapan penelitian didesain mulai dari pengumpulan data, desain sistem, implementasi, evaluasi, hingga analisis hasil dan kesimpulan. Secara lebih jelas, tahapan penelitian ini dapat dilihat pada Gambar 3. 1.



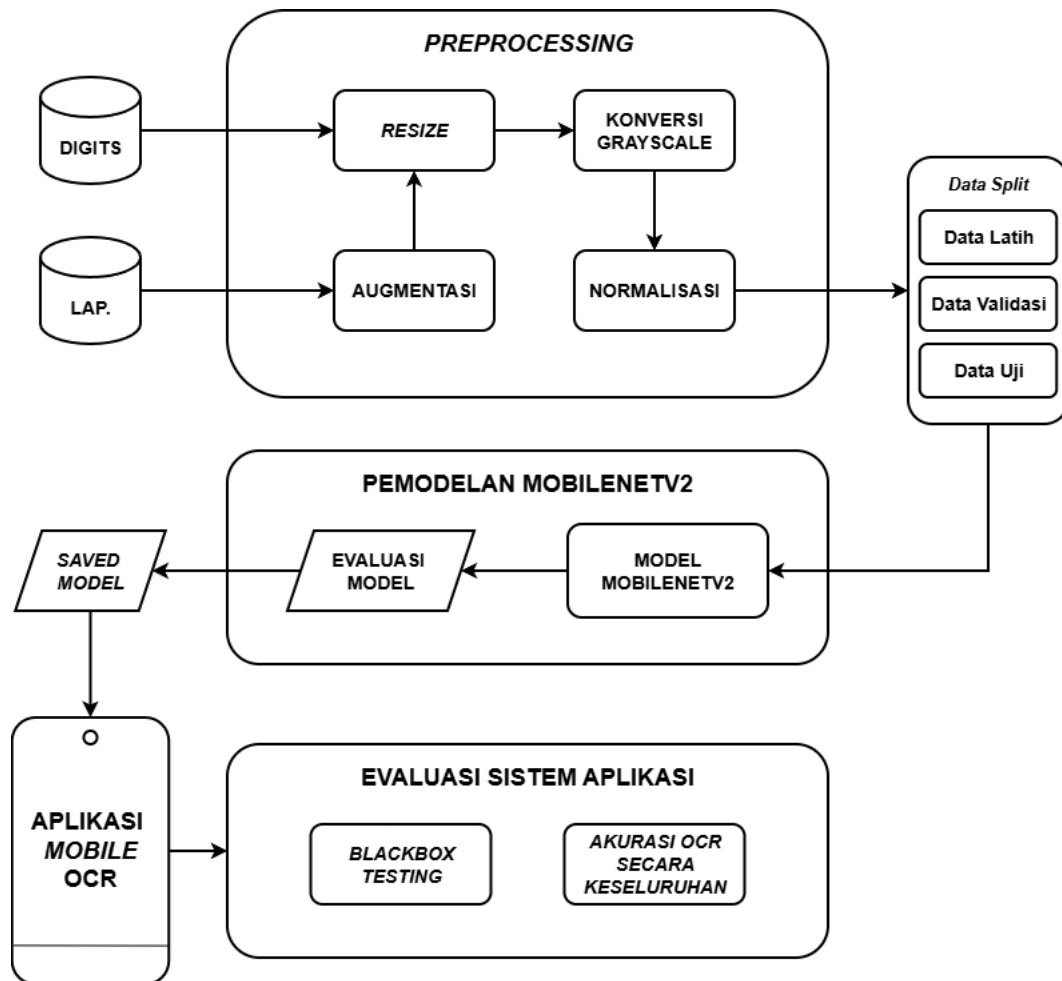
Gambar 3. 1 Tahapan penelitian

3.1.2 Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah *dataset* publik dan beberapa data yang diambil di lapangan. *Dataset* publik yang digunakan ialah *dataset* digit DIGITS yang didapatkan dari Kaggle, yang berisi citra angka dengan berbagai variasi bentuk, gaya, dan *font*. *Dataset* ini berfungsi sebagai data awal dalam pelatihan model untuk mempelajari karakteristik dasar dari digit. Selain itu, penelitian ini juga mengumpulkan sejumlah data nyata dari lapangan berupa foto meteran air milik pelanggan PAMDes Pandanrejo. Pengambilan gambar dilakukan menggunakan kamera ponsel guna merepresentasikan kondisi penggunaan sistem secara riil. Data lapangan tersebut selanjutnya digunakan pada tahap *fine-tuning* model yang sudah dilatih sebelumnya, sehingga model dapat beradaptasi dengan karakteristik digit pada meteran air yang sebenarnya.

3.1.3 Desain Sistem

Tahap ini melakukan perancangan alur kerja keseluruhan sistem yang menggambarkan proses dimulai dengan pengolahan data citra hingga implementasi pada aplikasi *mobile* OCR. Alur desain sistem penelitian ini secara keseluruhan dapat dilihat pada Gambar 3. 2.



Gambar 3. 2 Desain sistem

Berdasarkan Gambar 3. 2, desain sistem diawali dengan tahap *preprocessing* yang berfungsi menyamakan dan menyesuaikan karakteristik data gambar supaya cocok dengan kebutuhan model. Pada tahap ini dilakukan beberapa proses seperti *resize* untuk menyeragamkan ukuran citra, konversi ke *grayscale* untuk menyederhanakan informasi warna, augmentasi data guna menambah variasi data, serta normalisasi untuk menyesuaikan nilai piksel citra.

Selanjutnya, data yang sudah melewati tahap *preprocessing* akan dilakukan pembagian data (*data splitting*) untuk digunakan pada tahap pemodelan

MobileNetV2. Pada tahap ini, dataset dibagi menjadi data latih, data validasi, dan data uji.

Tahap berikutnya ialah proses pemodelan MobileNetV2. Model dilatih menggunakan data latih dan validasi hingga diperoleh performa terbaik. Model yang telah dilatih maka akan dievaluasi untuk mengetahui tingkat akurasi dalam mengenali digit, dan selanjutnya disimpan (*saved model*) untuk digunakan pada tahap implementasi sistem.

Kemudian model MobileNetV2 dengan performa terbaik diintegrasikan pada aplikasi *mobile* OCR sehingga aplikasi dapat melakukan proses pembacaan angka meteran air secara otomatis melalui kamera perangkat *mobile*. Integrasi ini membuat sistem untuk bekerja secara *real-time*, dimana citra yang ditangkap akan diproses dan dikenali secara langsung oleh model.

Tahap terakhir adalah evaluasi sistem, yang dilakukan untuk mengetahui tingkat kinerja dan keakuratan sistem dalam mengenali digit pada meteran air. Evaluasi dilakukan menggunakan pengujian *blackbox testing* serta akurasi OCR secara keseluruhan.

3.1.4 Input Dataset

Input dataset pada penelitian ini merupakan data citra yang digunakan sebagai *inputan* utama dalam pelatihan sekaligus pengujian model OCR. Terdapat dua jenis *dataset* yang digunakan, yaitu dataset publik DIGITS dan data citra meteran air hasil pengambilan di lapangan. Kombinasi antara *dataset* publik dan data lapangan ini berfungsi agar bisa memberikan variasi data yang cukup sehingga

model mampu mempelajari ciri umum digit sekaligus beradaptasi dengan kondisi nyata pada meteran air.

3.1.5 *Preprocessing*

Preprocessing berfungsi untuk menyiapkan data citra digit meteran air agar sesuai untuk proses pelatihan model. Tahap *preprocessing* yang meliputi proses augmentasi data khusus untuk data yang berasal dari pengambilan di lapangan, *resize*, konversi ke *grayscale*, dan normalisasi.

3.1.5.1 Augmentasi Data

Tahap ini melakukan penambahan variasi pada citra supaya kuantitas dan keragaman data *training* bisa meningkat. Augmentasi data hanya diterapkan pada data yang diperoleh dari pengambilan langsung di lapangan, karena jumlah digit yang diperoleh relatif terbatas. Dengan menerapkan augmentasi, data tersebut dapat diperbanyak sekaligus dibuat lebih bervariasi sehingga model bisa mempelajari berbagai kondisi citra yang berbeda. Sebelum dilakukan augmentasi, data lapangan berjumlah 3 citra untuk setiap kelas digit 0 hingga 9. Setelah proses augmentasi dilakukan, jumlah data tersebut meningkat menjadi 100 citra untuk setiap kelas, sehingga total data lapangan menjadi 1000 citra. Sementara itu, *dataset* DIGITS yang diperoleh dari Kaggle tidak dilakukan augmentasi, karena *dataset* tersebut sudah memiliki variasi bentuk, gaya penulisan, dan *font* digit angka yang beragam sehingga cukup merepresentasikan variasi data yang dibutuhkan untuk proses pelatihan model.

Teknik augmentasi yang diterapkan dalam penelitian ini mencakup *brightness adjustment* untuk memodifikasi tingkat kecerahan citra, *contrast adjustment* untuk memodifikasi perbedaan intensitas antara piksel terang dan gelap, *gaussian noise* untuk menambahkan gangguan acak pada citra, *gaussian blur* untuk mensimulasikan kondisi citra yang sedikit kabur, serta *rotation* untuk memutar citra pada sudut tertentu. Penerapan berbagai teknik augmentasi tersebut bertujuan agar model menjadi lebih *robust* meskipun terdapat variasi pada citra.

3.1.5.2 Resize

Tahap ini mengubah seluruh ukuran citra menjadi 224×224 piksel, karena arsitektur MobileNetV2 menggunakan ukuran tersebut sebagai standar *input* pada proses pelatihan maupun inferensi. Proses *resize* bertujuan untuk menyeragamkan dimensi seluruh data citra sehingga bisa diproses dengan konsisten oleh model CNN. Data yang melalui proses *resize* dalam penelitian ini meliputi *dataset* DIGITS serta data digit hasil pengambilan di lapangan yang telah melalui tahap augmentasi data. Dengan menyeragamkan ukuran citra dengan format yang sama maka dapat membantu meningkatkan stabilitas serta kinerja model dalam proses klasifikasi digit.

3.1.5.3 Konversi ke Grayscale

Pada tahap konversi ke *grayscale*, citra yang semula memiliki tiga kanal diubah menjadi citra *grayscale* yang hanya mempunyai satu kanal intensitas. Mekanisme ini dilakukan untuk meringkas informasi pada citra dengan menghapus komponen warna yang tidak terlalu diperlukan dalam proses pengenalan digit.

Dalam tugas klasifikasi digit, model tidak perlu terlalu memperhatikan variasi warna, melainkan lebih berfokus pada pola bentuk angka, perbedaan intensitas piksel, serta kontras antara objek digit dan latar belakang. Dengan mengubah citra menjadi *grayscale*, jumlah dimensi data menjadi lebih sederhana sehingga dapat mengurangi kompleksitas komputasi serta membantu model lebih fokus dalam mempelajari karakteristik bentuk digit.

3.1.5.4 Normalisasi

Tahap ini adalah proses penskalaan nilai piksel pada citra agar berada dalam rentang yang lebih terkontrol sebelum dimasukkan ke dalam model. Dalam penelitian ini digunakan normalisasi *Min-Max*, yaitu mengganti nilai piksel citra yang awalnya berada pada rentang 0 – 255 menjadi ke rentang [0,1]. Proses ini diterapkan dengan cara membagi setiap nilai piksel dengan nilai maksimum piksel yaitu 255. Tujuan dari normalisasi adalah untuk menyamakan skala data. Selain itu, normalisasi juga membantu model dalam memahami pola pada citra dengan lebih baik karena nilai *input* yang digunakan berada pada rentang yang seragam dan tidak terlalu besar.

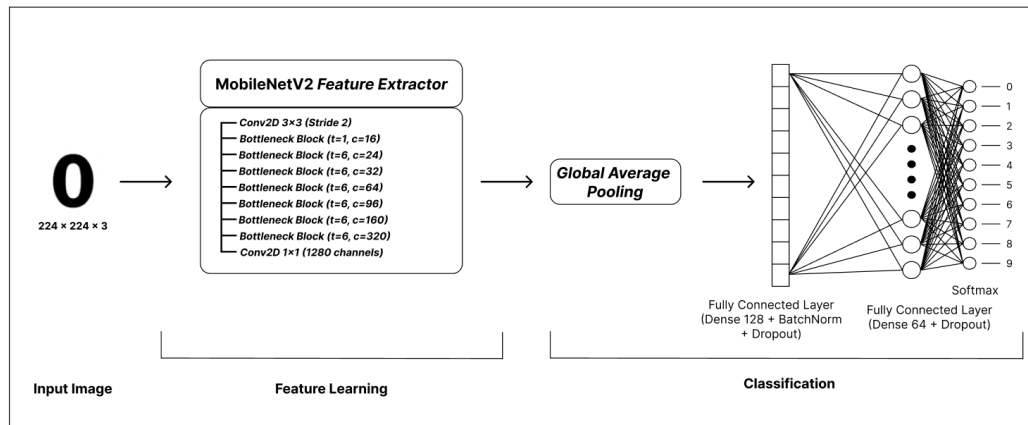
Setelah proses normalisasi selesai, citra yang sebelumnya telah dikonversi menjadi *grayscale* kemudian diubah kembali menjadi citra dengan tiga *channel*. Hal ini dilakukan karena arsitektur MobileNetV2 membutuhkan *input* citra dengan tiga *channel* seperti pada citra RGB. Oleh karena itu, citra *grayscale* yang hanya memiliki satu *channel* akan diduplikasi menjadi tiga *channel* yang identik, sehingga menghasilkan format citra dengan dimensi yang sesuai dengan kebutuhan model. Proses ini memungkinkan model MobileNetV2 tetap dapat digunakan.

3.1.6 *Data Splitting*

Pada tahap ini, *dataset* yang telah melalui proses *preprocessing* kemudian dibagi menjadi beberapa bagian untuk keperluan pelatihan dan evaluasi model. Pembagian data dilakukan dengan proporsi 80% data latih, 10% data validasi, dan 10% data uji. Data latih dipakai pada proses pelatihan model untuk mempelajari pola dan karakteristik digit pada citra. Data validasi dimanfaatkan untuk memonitoring kinerja model selama pelatihan. Sementara itu, data uji dimanfaatkan pada tahap akhir untuk mengevaluasi performa model dalam mengklasifikasikan digit pada data yang tidak pernah digunakan selama proses pelatihan. Pembagian data ini bertujuan agar menghasilkan model yang memiliki kemampuan generalisasi yang baik.

3.1.7 **Pemodelan MobileNetV2**

Tahap ini merupakan proses pemodelan menggunakan arsitektur MobileNetV2 untuk melakukan klasifikasi digit pada citra hasil segmentasi meteran air. Pada tahap ini dibahas beberapa aspek yang meliputi arsitektur model, alur proses pemodelan, hingga evaluasi kinerja model yang digunakan dalam penelitian. Model MobileNetV2 yang digunakan merupakan model *pretrained* dari *dataset* ImageNet, di mana telah mempunyai kemampuan awal dalam mengekstraksi berbagai fitur visual pada citra. Pemanfaatan model *pretrained* ini bertujuan untuk meningkatkan efektivitas proses pelatihan serta membantu model dalam mempelajari pola digit dengan lebih baik, terutama ketika jumlah *dataset* yang digunakan terbatas. Arsitektur MobileNetV2 yang digunakan dalam penelitian ini ditunjukkan pada Gambar 3. 3.



Gambar 3. 3 Arsitektur MobileNetV2

Berdasarkan arsitektur pada Gambar 3. 3, proses pemodelan diawali dari *input* citra berukuran $224 \times 224 \times 3$ yang telah melewati tahap *preprocessing*. Citra tersebut kemudian dimasukkan ke dalam bagian MobileNetV2 *feature extractor* untuk melakukan proses ekstraksi fitur. Pada bagian ini, MobileNetV2 menggunakan beberapa lapisan konvolusi yang terdiri dari Conv2D 3×3 , diikuti dengan beberapa *bottleneck block* yang mengombinasikan konsep *depthwise separable convolution* dan *inverted residual*. Lapisan-lapisan ini berfungsi untuk mengekstraksi berbagai fitur penting dari citra digit, seperti pola bentuk angka, tepi, serta karakteristik visual lainnya yang dapat membedakan setiap digit.

Selanjutnya, fitur yang dihasilkan akan diproses menggunakan *Global Average Pooling* (GAP). Lapisan ini berfungsi untuk mereduksi dimensi fitur dengan cara mengambil nilai rata-rata dari semua feature map yang dihasilkan oleh *feature extractor*. Melalui proses ini, representasi fitur yang semula berbentuk tensor atau matriks multi-dimensi akan diringkas menjadi vektor fitur satu dimensi (1D). Vektor tersebut kemudian dijadikan sebagai *input* untuk lapisan *fully connected* untuk melakukan proses klasifikasi digit.

Selanjutnya, hasil dari GAP akan diteruskan ke bagian klasifikasi yang terdiri dari beberapa *Fully Connected Layer (Dense Layer)*. Pada penelitian ini, bagian klasifikasi pada arsitektur MobileNetV2 dimodifikasi agar cocok dengan kebutuhan penelitian. Hal ini dilakukan karena model MobileNetV2 yang dipakai adalah model *pretrained* dari *dataset* ImageNet yang secara default dirancang untuk melakukan klasifikasi terhadap 1000 kelas objek. Sementara itu, penelitian ini hanya membutuhkan proses klasifikasi terhadap 10 kelas digit, yaitu angka 0 hingga 9. Oleh karena itu, lapisan klasifikasi bawaan diganti dengan arsitektur klasifikasi yang lebih sesuai.

Pada penelitian ini digunakan dua lapisan *dense*, yaitu lapisan dengan 128 neuron yang dilengkapi dengan *Batch Normalization* dan *Dropout*, serta lapisan *dense* kedua dengan 64 neuron yang juga menggunakan *Dropout*. Pelatihan model menjadi lebih stabil melalui penerapan *Batch Normalization*, yang menormalkan distribusi nilai aktivasi pada setiap batch data. Selain itu, penerapan *Dropout* membantu mengurangi *overfitting* dengan cara mengabaikan aktivasi sejumlah neuron secara acak selama proses pembelajaran.

Pada lapisan akhir digunakan fungsi aktivasi *Softmax* untuk menghasilkan probabilitas prediksi terhadap 10 kelas digit (0–9). Kelas yang memiliki nilai probabilitas paling tinggi akan dipilih menjadi hasil prediksi dari model. Melalui arsitektur tersebut, model diharapkan mampu mengenali dan mengklasifikasikan setiap digit hasil segmentasi dengan baik sehingga dapat digunakan dalam proses pembacaan angka pada meteran air dalam aplikasi *mobile* OCR.

3.1.8 Skenario Eksperimen Penentuan Model Terbaik

Pada tahap ini dilakukan skenario eksperimen untuk menentukan model terbaik dengan membandingkan beberapa konfigurasi parameter, khususnya *optimizer* dan *learning rate*. Variasi *learning rate* diterapkan pada tahap *fine-tuning* untuk melihat pengaruhnya terhadap kinerja model. Hasil dari setiap skenario kemudian dianalisis untuk mendapatkan konfigurasi model dengan performa yang paling optimal. Model terbaik yang diperoleh selanjutnya akan diintegrasikan ke dalam aplikasi *mobile* OCR yang dikembangkan.

3.1.9 Desain Sistem Aplikasi *Mobile* OCR

Pada tahap ini dilakukan perancangan sistem aplikasi *mobile* OCR untuk pembacaan meteran air menggunakan model yang telah dibuat. Perancangan sistem disusun berdasarkan metode pengembangan *Waterfall*, yang terdiri dari tahapan berurutan mulai dari *requirements definition* (analisis kebutuhan) hingga *operation and maintenance* (operasi dan pemeliharaan).

3.1.9.1 *Requirements Definition*

Tahap ini berfokus pada pengumpulan dan penentuan kebutuhan sistem. Dalam penelitian ini, proses identifikasi kebutuhan dilaksanakan dengan wawancara dan observasi dengan pengelola PAMDes Pandanrejo untuk mengetahui kebutuhan fungsional. Rincian kebutuhan fungsional selengkapnya ditunjukkan pada Tabel 3. 1

Tabel 3. 1 Kebutuhan Fungsional

No	Requirements	Keterangan
1.	<i>Login</i>	Digunakan untuk memasuki aplikasi oleh petugas lapangan yang terdaftar.
2.	Akses kamera	Digunakan untuk proses pengambilan gambar meteran air menggunakan kamera ponsel pengguna.
3.	OCR	Digunakan untuk ekstraksi digit jumlah pemakaian air pelanggan.
4	<i>Input Baca Meter</i>	Digunakan untuk melakukan pencatatan tagihan pelanggan. Dalam hal ini meliputi pemilihan pelanggan dan hasil pemindaian tagihan oleh OCR.
5.	Histori	Digunakan untuk menyimpan riwayat hasil aktivitas pembacaan.
6.	Profil	Digunakan untuk melihat profil pengguna aplikasi.
7.	Pengaturan	Digunakan untuk membuka menu pengaturan. Dalam hal ini jika ingin mengubah tema aplikasi.
8.	Pengiriman data hasil pembacaan meteran air ke sistem web.	Aplikasi <i>mobile</i> OCR harus mampu mengirimkan hasil pembacaan meteran air ke sistem web eksternal yang digunakan sebagai media simulasi pengujian pengiriman data. Informasi yang dikirimkan meliputi digit angka hasil pembacaan pemakaian air, nomor pelanggan, serta citra meteran air hasil pengambilan kamera.
9.	<i>Logout</i>	Digunakan untuk keluar dari aplikasi.

Berdasarkan Tabel 3. 1 dapat dijelaskan bahwa kebutuhan fungsional yang telah dirumuskan adalah fitur penting yang harus tersedia agar aplikasi dapat berfungsi sesuai dengan tujuan penelitian. Fitur *login* diperlukan untuk menjamin bahwa hanya petugas lapangan yang telah terdaftar yang bisa mengakses aplikasi, sehingga keamanan penggunaan sistem dapat terjaga. Fitur akses kamera digunakan untuk mengambil gambar meteran air yang kemudian menjadi *input* utama pada proses pembacaan digit menggunakan teknologi OCR dengan pendekatan

segmentasi digit dan model CNN MobileNetV2. Selanjutnya, fitur *input* baca meter digunakan untuk mencatat hasil pembacaan meteran air pelanggan yang meliputi pemilihan pelanggan serta hasil pemindaian digit yang diperoleh dari proses OCR.

Selain itu, fitur histori berfungsi untuk menyimpan riwayat aktivitas pembacaan meteran air sehingga petugas dapat melihat kembali data pembacaan yang telah dilakukan sebelumnya. Fitur profil digunakan untuk menampilkan informasi pengguna aplikasi yang sedang aktif. Fitur pengaturan digunakan jika ingin mengubah tema. Sementara itu, fitur pengiriman data hasil pembacaan memungkinkan aplikasi untuk mengirimkan informasi hasil pembacaan meteran air ke sistem web eksternal yang digunakan sebagai media simulasi pengujian pengiriman data. Data yang dikirimkan meliputi hasil pembacaan digit jumlah pemakaian air, nomor pelanggan, serta citra meteran air yang diambil melalui kamera aplikasi. Terakhir, fitur *logout* digunakan untuk keluar dari aplikasi sehingga sesi penggunaan dapat diakhiri dengan aman.

Selain kebutuhan fungsional, pengembangan aplikasi ini juga memerlukan pemenuhan kebutuhan non-fungsional sebagai aspek pendukung kualitas sistem. Kebutuhan non-fungsional mencakup karakteristik yang harus dimiliki aplikasi. Rincian kebutuhan non-fungsional selengkapnya ditunjukkan pada Tabel 3. 2

Tabel 3. 2 Kebutuhan Non-Fungsional

No	Requirements	Keterangan
1.	Akses aplikasi	Aplikasi dapat diakses dan digunakan ketika perangkat/ponsel terhubung ke internet.
2.	User interface (UI)	Tampilan (UI) pada aplikasi mudah dipahami sekaligus digunakan.

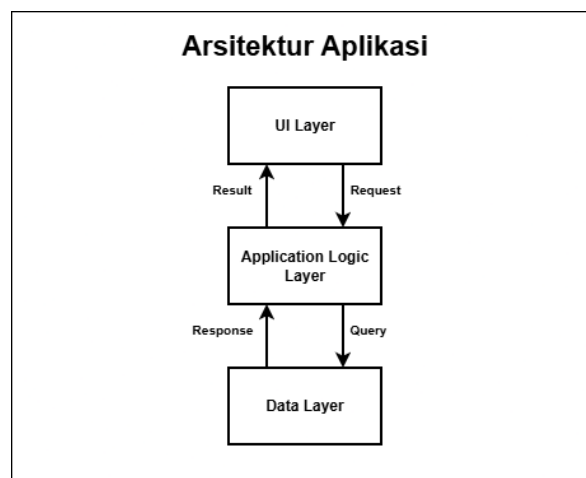
Berdasarkan Tabel 3. 2 kebutuhan non-fungsional yang ditetapkan bertujuan untuk memastikan aplikasi mampu memberikan pengalaman penggunaan yang baik serta mendukung kelancaran operasional di lapangan. Kebutuhan akses aplikasi menekankan bahwa sistem harus dapat dijalankan ketika perangkat terhubung ke internet, sehingga proses pengiriman data pembacaan meteran ke server *backend* aplikasi maupun sistem web eksternal dapat dikerjakan secara *real-time* dan tanpa hambatan. Sementara itu, kebutuhan terkait *user interface* (UI) menegaskan pentingnya tampilan UI yang sederhana, jelas, dan mudah dipahami *user*.

3.1.9.2 System and Software Design

Tahap ini bertujuan merancang solusi teknis berdasarkan kebutuhan yang telah ditetapkan. Perancangan sistem dilakukan secara berurutan, dimulai dari penyusunan struktur arsitektur sistem sebagai fondasi utama. Setelah itu, alur bisnis aplikasi dijelaskan untuk merepresentasikan proses kerja yang terjadi di dalam sistem. Hasil pemodelan proses ini kemudian menjadi dasar dalam merancang struktur basis data serta menggambarkan aliran data yang terjadi antar komponen sistem. Tahap perancangan diakhiri dengan pengembangan rancangan antarmuka pengguna sebagai representasi visual dari fitur dan fungsi yang akan diimplementasikan. Secara keseluruhan, desain sistem harus mampu menunjukkan bagaimana aplikasi akan dibangun secara teknis untuk memenuhi spesifikasi kebutuhan yang telah didefinisikan sebelumnya.

Arsitektur sistem pada penelitian ini dirancang menggunakan pendekatan *three-layer architecture* (arsitektur tiga lapisan) yang terdiri dari UI *Layer*,

Application Logic Layer, dan *Data Layer*. *UI Layer* berfungsi sebagai lapisan antarmuka yang berinteraksi langsung dengan pengguna, di mana setiap *input* atau permintaan layanan akan diteruskan sebagai *request* ke *Application Logic Layer*. Selanjutnya, *Application Logic Layer* bertugas memproses seluruh permintaan tersebut sesuai aturan dan alur bisnis aplikasi, termasuk melakukan pengambilan atau penyimpanan data melalui komunikasi dengan *Data Layer*. Lapisan *Data Layer* berperan sebagai pengelola *database* atau sumber data lainnya dan menyediakan *response* berupa data yang diperlukan oleh *Application Logic Layer*. Hasil pemrosesan kemudian dikembalikan kembali ke *UI Layer* untuk ditampilkan kepada pengguna, sebagaimana dapat dilihat pada Gambar 3. 4.

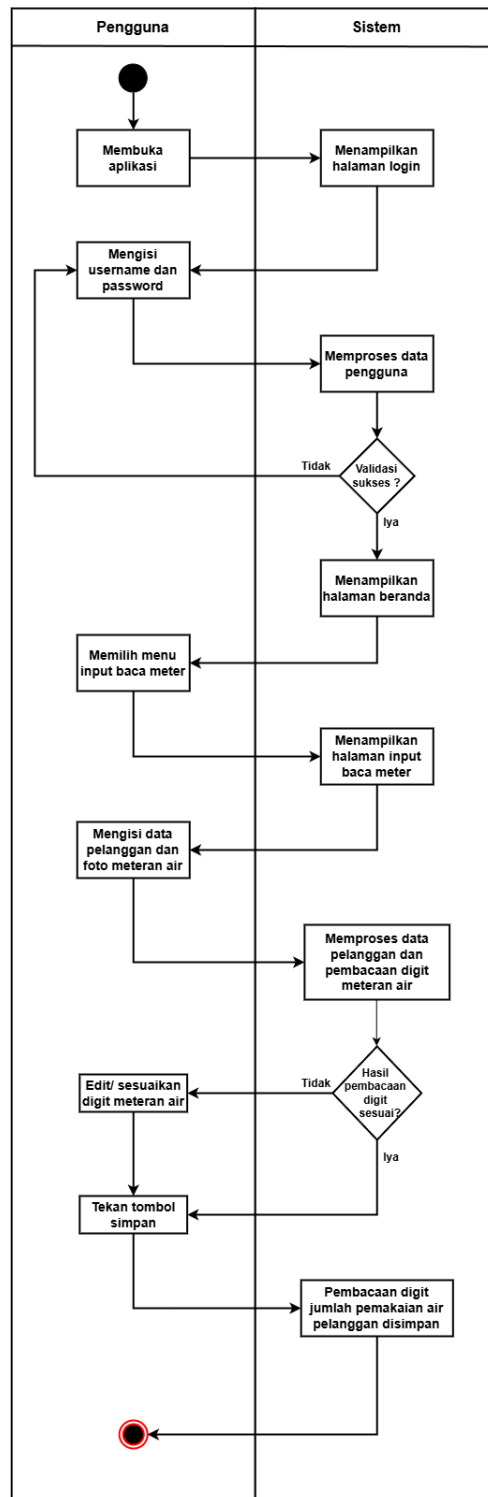


Gambar 3. 4 Arsitektur Aplikasi

Berdasarkan Gambar 3. 4 implementasi arsitektur aplikasi menunjukkan bagaimana setiap lapisan berinteraksi secara dua arah untuk mendukung proses pembacaan meteran air menggunakan OCR. *UI Layer* dibangun menggunakan *React Native (Expo)* yang berfungsi menampilkan antarmuka pengguna sekaligus menerima *input* berupa hasil tangkapan kamera meteran air. Citra yang diperoleh kemudian diteruskan ke *Application Logic Layer* untuk diproses lebih lanjut. Pada

lapisan ini dijalankan seluruh logika inti aplikasi, termasuk pemanggilan model OCR berbasis CNN MobileNetV2 yang sudah dikonversi ke format (*.tflite*) sehingga dapat dijalankan secara efisien pada *mobile device*. Model ini melakukan proses pengenalan digit secara langsung pada citra meteran dan menghasilkan *output* berupa angka pemakaian air. Hasil tersebut kemudian dikembalikan ke UI *Layer* untuk ditampilkan kepada pengguna secara *real-time*. Selain itu, *Application Logic Layer* juga berkomunikasi dengan *Data Layer*, yang pada penelitian ini menggunakan layanan Convex sebagai *backend-as-a-service* untuk menyimpan data hasil pembacaan, riwayat aktivitas, serta informasi pengguna.

Alur bisnis pada aplikasi *mobile* OCR pembacaan meteran air disusun untuk memberikan gambaran umum mengenai urutan proses yang terjadi saat petugas menggunakan aplikasi dalam melakukan pencatatan jumlah pemakaian air pelanggan. Alur tersebut direpresentasikan menggunakan salah satu diagram pada *Unified Modeling Language* (UML), yaitu *activity diagram*, yang menggambarkan langkah-langkah proses secara terstruktur dan runtut. Gambaran lengkap alur bisnis aplikasi ini dapat dilihat pada Gambar 3. 5.

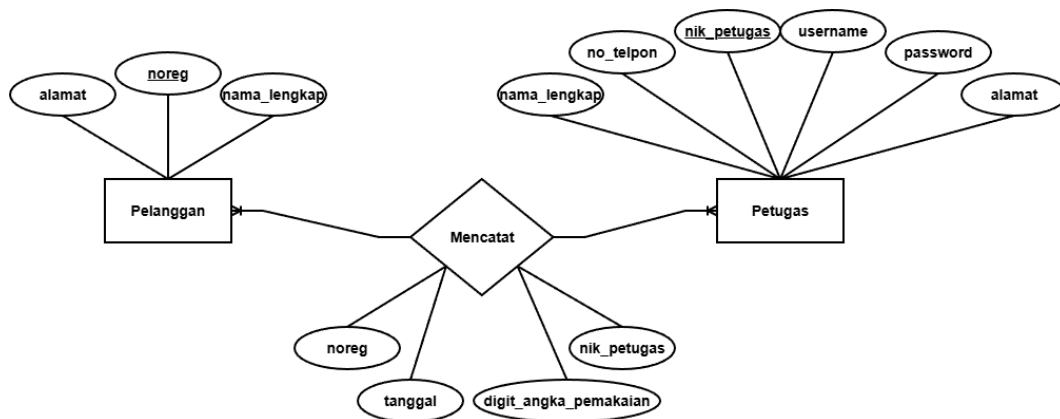


Gambar 3. 5 Alur Bisnis

Berdasarkan Gambar 3. 5, alur bisnis utama pada aplikasi *mobile* OCR pembacaan meteran air dimulai ketika user membuka aplikasi dan diarahkan ke

halaman *login* untuk mengisi *username* serta *password*. Sistem kemudian memvalidasi data tersebut, dan apabila berhasil maka menampilkan halaman beranda. Pengguna selanjutnya memilih menu *input* baca meter yang akan membawa mereka ke halaman pengisian data pelanggan dan unggah foto meteran air. Sistem memproses data tersebut, termasuk melakukan pembacaan digit meteran menggunakan model OCR. Jika hasil pembacaan dianggap tidak sesuai, pengguna dapat melakukan penyesuaian digit sebelum menyimpannya. Setelah pengguna menekan tombol simpan, sistem menyimpan data jumlah pemakaian air pelanggan dan proses pun berakhir.

Dalam proses perancangan sistem, desain *database* merupakan bagian penting yang dijadikan sebagai dasar untuk mengelola dan menyimpan data secara terstruktur agar dapat diakses dan diproses secara efisien. Desain *database* disusun berdasarkan kebutuhan data yang sudah dianalisis sebelumnya, sehingga setiap entitas yang terlibat dalam proses pembacaan meteran air seperti data pengguna, data pelanggan, data hasil pembacaan, serta riwayat aktivitas dapat terhubung secara logis dan mudah dikelola. Hubungan antar entitas dirumuskan untuk memastikan integritas dan konsistensi data, sehingga sistem dapat beroperasi dengan akurat dan dapat diandalkan. Struktur database tersebut direpresentasikan dengan ERD (*Entity Relational Diagram*). Secara lebih jelas ERD pada penelitian ini dapat dilihat pada Gambar 3. 6.

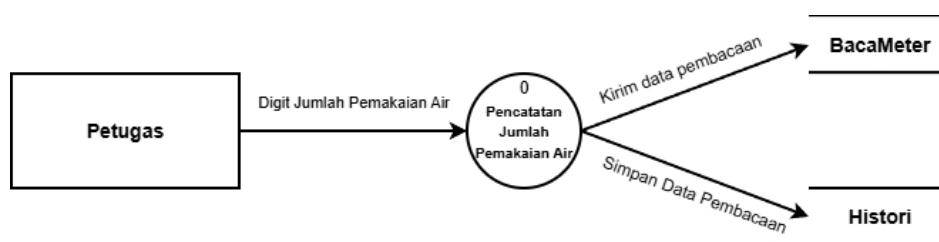


Gambar 3. 6 ERD

Berdasarkan Gambar 3. 6, ERD menggambarkan keterhubungan antara entitas Petugas dan Pelanggan yang dihubungkan melalui relasi Mencatat, yang pada tahap perancangan basis data akan diimplementasikan sebagai tabel tersendiri bernama Histori. Entitas Petugas merepresentasikan pengguna aplikasi yang bertugas melakukan pencatatan pemakaian meter air dan memiliki atribut *nik_petugas* sebagai kunci utama, serta *nama_lengkap*, *no_telpon*, *alamat*, *username*, dan *password* sebagai identitas dan informasi otentikasi. Entitas Pelanggan menyimpan data pengguna layanan air dengan atribut *noreg* sebagai kunci utama, serta *nama_lengkap* dan *alamat*. Relasi Mencatat mencerminkan aktivitas pencatatan pemakaian air yang dilakukan oleh petugas terhadap pelanggan dan memiliki atribut *tanggal* serta *digit_angka_pemakaian* sebagai hasil pembacaan meter air pada waktu tertentu. Hubungan antara entitas Petugas dan Pelanggan bersifat *many-to-many*. Sehingga relasi Mencatat perlu diwujudkan dalam bentuk tabel Histori yang menyimpan keterkaitan antara petugas dan pelanggan beserta data hasil pencatatan pemakaian air.

Untuk memahami bagaimana data bergerak dan diproses di dalam suatu sistem, diperlukan model yang mampu menunjukkan aliran informasi secara jelas dan terstruktur. Pada penelitian ini, *Data Flow Diagram* (DFD) digunakan untuk menggambarkan bagaimana aplikasi *mobile* OCR pembacaan meteran air menerima *input* dari petugas, memproses digit jumlah pemakaian air, serta mendistribusikan hasilnya ke dua tujuan berbeda. Proses pencatatan jumlah pemakaian air menghasilkan dua aliran data, yaitu pertama untuk disimpan ke dalam database internal yang digunakan oleh aplikasi, kemudian untuk dikirim ke sistem web eksternal. Gambaran aliran data tersebut dapat dilihat pada Gambar 3.

7.

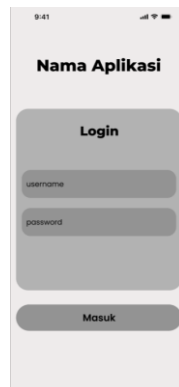


Gambar 3. 7 DFD

Pada tahap perancangan antarmuka pengguna (UI), desain dilakukan untuk memastikan tampilan aplikasi mudah dipahami dan dioperasikan oleh petugas lapangan yang menggunakan perangkat seluler. Perancangan UI menekankan kesederhanaan navigasi, kejelasan elemen visual, serta kemudahan akses terhadap fitur utama seperti *login*, pengambilan gambar meteran air, proses OCR, peninjauan hasil pembacaan, dan pengiriman data ke server. Setiap halaman dirancang agar alur interaksi pengguna berjalan secara terstruktur dan efisien, sehingga memungkinkan petugas untuk melakukan pencatatan jumlah penggunaan air

dengan cepat dan meminimalisir kesalahan dalam proses *input*. Secara lebih jelas, desain antarmuka pengguna pada aplikasi adalah sebagai berikut.

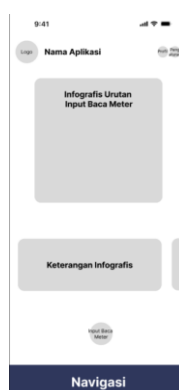
1. Desain UI Halaman *Login*



Gambar 3. 8 Desain UI Login

Gambar 3. 8 menunjukkan desain tampilan halaman *login* pada aplikasi. Pada halaman ini, *user* harus menginputkan *username* dan *password* sebagai proses autentikasi sebelum masuk ke sistem. Setelah data diisi, pengguna dapat menekan tombol masuk untuk melanjutkan ke halaman utama aplikasi.

2. Desain UI Halaman Beranda

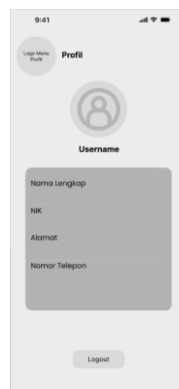


Gambar 3. 9 Desain UI Halaman Beranda

Gambar 3. 9 menunjukkan desain halaman beranda aplikasi setelah pengguna berhasil *login*. Halaman ini menampilkan informasi berupa infografis

urutan proses *input* baca meter serta keterangan terkait urutan tersebut. Pada halaman ini terdapat menu untuk menuju halaman profil dan pengaturan di pojok kanan atas. Sedangkan, pada bagian bawah terdapat tombol *input* baca meter yang digunakan untuk memulai proses pengambilan gambar meteran air. Selain itu, tersedia menu navigasi untuk memudahkan *user* berpindah ke halaman lain dalam aplikasi.

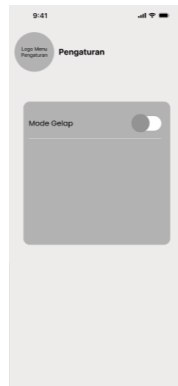
3. Desain UI Halaman Profil



Gambar 3. 10 Desain UI Halaman Profil

Gambar 3. 10 menampilkan desain halaman profil pada aplikasi. Halaman ini menyediakan informasi detail mengenai data petugas, seperti nama lengkap, nik, alamat, dan nomor telepon yang ditampilkan dalam format daftar. Pada bagian atas terdapat ikon profil serta *username* pengguna. Selain itu, tersedia tombol *logout* untuk keluar dari aplikasi dan diarahkan kembali ke halaman *login*.

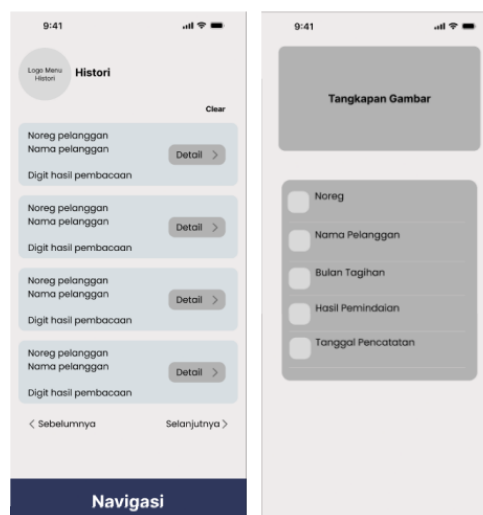
4. Desain UI Halaman Pengaturan



Gambar 3. 11 Desain UI Halaman Pengaturan

Gambar 3. 11 menunjukkan desain halaman pengaturan pada aplikasi. Pada halaman ini pengguna bisa mengubah preferensi tampilan aplikasi, salah satunya melalui fitur mode gelap yang dapat diaktifkan atau dinonaktifkan menggunakan tombol *switch*. Pengaturan ini bertujuan memberikan kenyamanan visual ketika aplikasi digunakan dalam kondisi cahaya minim.

5. Desain UI Halaman Histori



Gambar 3. 12 Desain UI Halaman Histori

Gambar 3. 12 menunjukkan dua antarmuka aplikasi. Antarmuka pertama adalah halaman histori yang menampilkan daftar riwayat tangkapan gambar pembacaan meter, masing-masing memuat noreg dan nama pelanggan serta digit hasil pembacaan. Antarmuka kedua adalah halaman detail dari setiap data histori yang menampilkan gambar hasil tangkapan, noreg dan nama lengkap pelanggan, bulan tagihan, digit hasil pemindaian, serta tanggal rekaman tersebut.

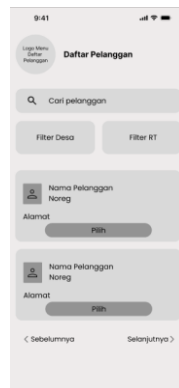
6. Desain UI Halaman *Input Baca Meter*



Gambar 3. 13 Desain UI Halaman Input Baca Meter

Gambar 3. 13 menunjukkan halaman *input* baca meter pada aplikasi. Pada halaman ini, *user* dapat memilih noreg / nama pelanggan dari daftar, mengambil gambar meter menggunakan tombol ambil gambar, dan melihat hasil pembacaan digit pada gambar yang akan ditampilkan secara otomatis. Di bagian bawah ada tombol kembali dan simpan untuk menyelesaikan proses *input* data.

7. Desain UI Halaman Pilih Pelanggan



Gambar 3. 14 Desain UI Halaman Daftar Pelanggan

Gambar 3. 14 menunjukkan desain halaman untuk memilih pelanggan. Pada halaman ini pengguna bisa melakukan pencarian di kolom cari pelanggan, menerapkan filter baik desa maupun rt, hingga langsung memilih pelanggan dengan menekan tombol pilih. Selain itu, pada bagian bawah terdapat *pagination* yang berfungsi beralih ke daftar pelanggan yang lain.

8. Desain UI Halaman Ambil Gambar Digit pada Meteran Air



Gambar 3. 15 Desain UI Halaman Ambil Gambar

Gambar 3. 15 menunjukkan desain tampilan layar ketika melakukan pengambilan gambar. Pada halaman ini terdapat *overlay* berbentuk beberapa kotak yang berfungsi sebagai panduan bagi pengguna untuk memposisikan

digit meteran air agar berada di dalam area yang telah ditentukan. *Overlay* tersebut juga berperan sebagai area segmentasi digit, di mana setiap kotak merepresentasikan satu digit angka yang akan diproses secara terpisah. Citra digit yang berada di dalam masing-masing kotak kemudian akan dipotong (segmentasi) dan diproses lebih lanjut oleh sistem untuk dilakukan klasifikasi menggunakan model MobileNetV2. Dengan adanya *overlay* ini, proses pengambilan gambar dapat menjadi lebih terarah sehingga posisi digit lebih konsisten dan memudahkan proses segmentasi serta pengenalan angka oleh model.

3.1.9.3 *Implementation*

Pada tahap *implementation*, dilakukan pengembangan sistem aplikasi berdasarkan seluruh desain dan kebutuhan yang telah dirumuskan pada tahap sebelumnya. Tahap ini mencakup realisasi seluruh komponen sistem ke dalam bentuk perangkat lunak yang dapat dijalankan, mulai dari pengolahan model hingga pembangunan aplikasi mobile secara utuh.

Proses implementasi diawali dengan pemodelan MobileNetV2 untuk klasifikasi digit menggunakan *dataset* yang telah dipersiapkan. Kemudian, model dikonversi menjadi format *TensorFlow Lite*. Selanjutnya, dilakukan pengembangan aplikasi *mobile* OCR menggunakan *React Native* dengan bantuan *Expo* sebagai *toolchain*, sesuai dengan desain antarmuka dan alur sistem yang telah didesain sebelumnya. Selain itu, sistem *backend* dikembangkan menggunakan *Convex*

sebagai *backend-as-a-service*. *Backend* ini digunakan untuk mengelola data hasil pembacaan meteran air serta mendukung komunikasi antara aplikasi dan basis data.

3.1.9.4 Integration

Pada tahap ini, seluruh komponen sistem, termasuk model OCR, antarmuka pengguna, dan basis data, diintegrasikan menjadi satu kesatuan aplikasi. Proses integrasi ini bertujuan untuk menjamin bahwa setiap komponen dapat saling berkomunikasi dan bekerja secara terkoordinasi dalam menjalankan fungsi aplikasi secara keseluruhan.

3.1.9.5 Operation and Maintenance

Tahap ini berlangsung ketika aplikasi sudah digunakan dalam lingkungan nyata. Pemeliharaan dilakukan guna mengatasi kesalahan yang ditemukan setelah implementasi, menyesuaikan aplikasi dengan perubahan kebutuhan, serta melakukan peningkatan fitur guna menjaga kinerja dan keandalan sistem.

3.1.10 Evaluasi Sistem Aplikasi

Pada tahap ini dilakukan evaluasi terhadap sistem aplikasi *mobile* OCR yang telah dikembangkan untuk mengetahui kinerja dan tingkat keberhasilan sistem dalam membaca digit meteran air. Evaluasi dilakukan menggunakan metode *blackbox testing* untuk menguji fungsionalitas aplikasi berdasarkan kebutuhan pengguna dan memastikan seluruh fitur dapat berjalan dengan baik. Selain itu, digunakan akurasi OCR secara keseluruhan untuk mengukur tingkat kinerja sistem

OCR dalam keberhasilan mengenali digit pada citra meteran air. Hasil evaluasi ini digunakan sebagai acuan dalam menilai performa sistem secara keseluruhan.

3.2 Implementasi

Tahap ini melakukan implementasi dari desain sistem yang telah dibuat pada tahap sebelumnya. Implementasi mencakup pengumpulan data, pengembangan aplikasi *mobile* OCR yang mampu melakukan pembacaan angka meteran air secara otomatis menggunakan kamera perangkat. Pendekatan yang diterapkan dalam penelitian ini ialah segmentasi digit, di mana proses pemisahan setiap digit angka dilakukan berdasarkan area pembacaan yang telah ditentukan pada antarmuka kamera aplikasi. Tata letak (*layout*) pada tampilan kamera berfungsi sebagai pembatas area yang membantu pengguna memposisikan setiap digit pada meteran air sehingga angka yang terbaca berada dalam area yang telah ditentukan. Setiap digit yang diperoleh dari proses segmentasi tersebut kemudian diproses dan diklasifikasikan menggunakan model MobileNetV2 untuk menghasilkan prediksi angka pada masing-masing digit. Selanjutnya, hasil prediksi dari setiap digit angka tersebut akan digabungkan kembali secara berurutan sehingga diperoleh hasil pembacaan atau pemindaian meteran air yang utuh.

3.2.1 Implementasi Pengumpulan Dataset

Dataset DIGITS tersusun dalam sepuluh folder yang merepresentasikan kelas digit dari angka 0 hingga 9. Pada setiap folder terdapat 1016 file gambar yang berisi variasi bentuk, gaya, dan *font* digit yang berbeda-beda, sehingga total *dataset*

ini berjumlah 10.160 data gambar. Contoh data pada *dataset* ini dapat dilihat pada Tabel 3. 3.

Tabel 3. 3 Contoh Data pada Dataset DIGITS

No	Angka	Variasi Bentuk
1.	0	0
2.	0	0
3.	3	3
4.	3	3

Sedangkan, data lapangan yang berhasil dikumpulkan dalam penelitian ini berjumlah 14 gambar meteran air yang diambil langsung dari lingkungan PAMDes Pandanrejo. Data tersebut memerlukan proses pemotongan (*cropping*) secara manual pada setiap digit dari gambar meteran air asli agar dapat dijadikan sebagai data latih. Proses ini dilakukan untuk menghasilkan citra digit tunggal yang sesuai dengan format *dataset* yang digunakan dalam pelatihan model. Dari hasil proses tersebut, diperoleh beberapa citra untuk masing-masing digit. Akan tetapi, pada penelitian ini hanya dipilih sebanyak 3 citra untuk setiap kelas digit 0 hingga 9 agar jumlah data pada setiap kelas tetap seimbang, sehingga total data lapangan berjumlah 30 citra. Contoh hasil pemotongan digit dari data lapangan dapat dilihat pada Tabel 3. 4. Selanjutnya, data lapangan yang digunakan dilakukan proses augmentasi untuk menambah variasi citra pada setiap kelas digit hingga masing-masing kelas memiliki 100 gambar. Dengan demikian, total data lapangan setelah

augmentasi menjadi 1000 citra. Jika digabungkan dengan dataset DIGITS yang berjumlah 10.160 data gambar, maka total keseluruhan data yang digunakan dalam pelatihan model adalah sebanyak 11.160 citra.

Tabel 3. 4 Pemotongan Digit pada Gambar Asli

No	Gambar Asli	Setelah <i>Cropping</i>
1.		
2.		

3.2.2 Implementasi Pemodelan MobileNetV2

Implementasi pemodelan MobileNetV2 dilakukan berdasarkan desain arsitektur yang telah dirancang sebelumnya, dengan menerapkan setiap komponen model mulai dari input layer hingga output layer sesuai konfigurasi yang telah ditentukan.

3.2.2.1 *Input Layer*

Pada tahap *input layer*, model menerima citra *input* yang telah melalui seluruh *preprocessing* sebelumnya. Citra yang dipakai sebagai *input* memiliki ukuran 224×224 piksel dengan 3 *channel*, sehingga membentuk dimensi

$224 \times 224 \times 3$ yang cocok dengan kebutuhan arsitektur MobileNetV2. Ukuran tersebut diperoleh dari proses *resize* yang dilakukan pada tahap *preprocessing*, sedangkan jumlah *channel* diperoleh dari proses duplikasi citra *grayscale* menjadi tiga *channel* agar kompatibel dengan model yang menggunakan bobot *pretrained*.

Sebelum dimasukkan ke dalam model, citra pada dasarnya direpresentasikan sebagai matriks piksel. Jika sebuah citra memiliki tinggi H , lebar W , dan jumlah kanal C , maka representasi matriks citra dapat dituliskan sebagai:

$$I \in R^{H \times W \times C} \quad (3.1)$$

Keterangan:

H : Tinggi citra (*height*).

W : Lebar citra (*width*).

C : Jumlah *channel* warna.

Pada penelitian ini, setelah proses *preprocessing* diperoleh ukuran citra:

$$I \in R^{224 \times 224 \times 3} \quad (3.2)$$

Ketika citra dimasukkan ke dalam model, data tersebut kemudian direpresentasikan dalam bentuk tensor dengan menambahkan dimensi batch size.

Secara matematis, tensor *input* dapat dituliskan sebagai:

$$X \in R^{N \times H \times W \times C} \quad (3.3)$$

Keterangan:

N : Jumlah sampel dalam satu batch (*batch size*).

H : Tinggi citra.

W : Lebar citra.

C : Jumlah *channel* warna.

Sehingga dalam penelitian ini bentuk tensor yang digunakan sebagai *input* model adalah:

$$X \in R^{N \times 224 \times 224 \times 3} \quad (3.4)$$

Tensor tersebut kemudian diproses oleh *feature extractor* MobileNetV2 untuk melakukan ekstraksi fitur dari citra digit.

3.2.2.2 MobileNetV2 Feature Extractor/ Backbone

Pada tahap MobileNetV2 *Feature Extractor (Backbone)*, citra yang telah masuk melalui *input* layer akan diproses untuk melakukan ekstraksi fitur. Bagian ini merupakan inti dari arsitektur MobileNetV2 yang bertugas untuk mempelajari berbagai karakteristik visual dari citra, seperti pola bentuk, tepi, tekstur, serta struktur digit yang terdapat pada gambar meteran air. *Feature extractor* pada MobileNetV2 memiliki beberapa lapisan konvolusi yang dimulai dengan Conv2D 3×3 , kemudian diteruskan dengan serangkaian *bottleneck block* yang mengombinasikan konsep *depthwise separable convolution* dan *inverted residual*.

Lapisan awal MobileNetV2 menggunakan operasi konvolusi 2D (Conv2D) dengan kernel berukuran 3×3 guna mengekstraksi fitur dasar dari citra *input*. Operasi konvolusi ini secara matematis dapat dinyatakan sebagai:

$$Y(i, j, k) = \sum_{m=0}^{K-1} \sum_{n=0}^{K-1} \sum_{c=0}^{C-1} X(i + m, j + n, c) \cdot W(m, n, c, k) \quad (3.5)$$

Keterangan:

X : Citra *input*.

W : Kernel konvolusi.

C : Jumlah *channel input*.

Y : *Feature map* yang dihasilkan.

Setelah lapisan konvolusi awal, MobileNetV2 menerapkan konsep *Depthwise Separable Convolution* untuk meningkatkan efisiensi komputasi. Operasi ini membagi proses konvolusi menjadi dua bagian, yaitu *depthwise*

convolution dan *pointwise convolution*. Pada tahap *depthwise convolution*, setiap *channel* dari citra diproses sendiri-sendiri menggunakan kernel konvolusi yang berbeda. Secara matematis dapat dinyatakan sebagai:

$$Y_k = X_k * K_k \quad (3.6)$$

Keterangan:

- X_k : *Channel* ke- k dari citra *input*.
- K_k : Kernel konvolusi pada *channel* tersebut.
- Y_k : hasil konvolusi (*feature map*) dari *channel* ke- k

Selanjutnya dilakukan *pointwise convolution* dengan kernel berukuran 1×1 untuk menggabungkan informasi antar *channel*:

$$Y(i, j, k) = \sum_{c=1}^C X(i, j, c) \cdot W(1, 1, c, k) \quad (3.7)$$

Keterangan:

- $X(i, j, c)$: Nilai piksel pada posisi (i, j) pada *channel* ke- c .
- $W(1, 1, c, k)$: Bobot kernel konvolusi berukuran 1×1 .
- C : Jumlah *channel* pada *feature map* *input*.
- $Y(i, j, k)$: Nilai piksel pada posisi (i, j) pada *feature map* *output*.

Pendekatan ini dapat mereduksi jumlah parameter dan kompleksitas komputasi dalam jumlah yang besar dibandingkan konvolusi standar, sehingga membuat MobileNetV2 lebih efisien untuk diimplementasikan pada *device* yang memiliki keterbatasan sumber daya seperti perangkat *mobile*.

MobileNetV2 juga menggunakan struktur *Inverted Residual Block*, yaitu blok *bottleneck* yang pertama-tama melakukan ekspansi untuk memperluas jumlah *channel*, kemudian melakukan *depthwise convolution*, dan akhirnya melakukan proyeksi untuk mereduksi kembali jumlah *channel*. Pada tahap terakhir ini diterapkan konsep *Linear Bottleneck*, yaitu tanpa menerapkan fungsi aktivasi *non-linear* seperti ReLU. Hal ini dilakukan karena penggunaan aktivasi *non-linear* pada

dimensi fitur yang rendah dapat menyebabkan hilangnya informasi penting. Oleh karena itu, transformasi pada tahap proyeksi dilakukan secara *linear* agar informasi fitur tetap terjaga.

Selain itu, MobileNetV2 juga menerapkan konsep *skip connection* atau *residual connection* yang menghubungkan *input* blok dengan *output* blok secara langsung. Mekanisme ini membantu menjaga aliran informasi dan mempermudah proses pelatihan jaringan yang dalam. Secara matematis, *residual connection* dapat dituliskan sebagai:

$$y = F(x) + x \quad (3.8)$$

Keterangan:

x : *Input* dari suatu blok.

$F(x)$: Transformasi yang dilakukan oleh lapisan konvolusi dalam blok tersebut.

y : *Output* akhir setelah dijumlahkan dengan *input* awal.

Namun, pada MobileNetV2 *skip connection* hanya digunakan ketika nilai *stride* = 1 dan jumlah *channel input* sama dengan jumlah *channel output*. Apabila terjadi proses *downsampling* dengan *stride* = 2, maka ukuran *feature map* akan berubah sehingga *skip connection* tidak dapat diterapkan.

Setelah melalui serangkaian *bottleneck block* tersebut, pada bagian akhir *feature extractor* digunakan konvolusi 1×1 (Conv2D) untuk meningkatkan kuantitas *channel* fitur menjadi 1280 *channel*. Lapisan ini berfungsi untuk memperkaya representasi fitur kompleks yang sudah dipelajari oleh model dari citra *input* sebelum diteruskan ke tahap *Global Average Pooling* untuk proses klasifikasi selanjutnya.

3.2.2.3 Global Average Pooling

Pada tahap *Global Average Pooling* (GAP), *feature map* yang dihasilkan dari *MobileNetV2 Feature Extractor* akan direduksi dimensinya sebelum diteruskan ke lapisan klasifikasi. Mekanisme GAP dilakukan dengan merata-ratakan nilai pada masing-masing *feature map* yang diperoleh dari lapisan konvolusi sebelumnya. Dengan demikian, setiap *channel* pada *feature map* akan diwakili oleh satu nilai rata-rata, sehingga hasilnya ialah vektor fitur satu dimensi yang lebih ringkas namun tetap menjaga informasi krusial yang telah dipelajari oleh model. Secara matematis, operasi GAP dapat dinyatakan sebagai berikut:

$$y_k = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W x_{i,j,k} \quad (3.9)$$

Keterangan:

- $x_{i,j,k}$: Nilai piksel pada posisi (i, j) pada *feature map channel* ke- k .
- H dan W : Tinggi dan lebar *feature map*.
- y_k : Nilai rata-rata yang merepresentasikan *channel* ke- k pada *output*.

Melalui proses ini, *feature map* dengan dimensi $H \times W \times C$ akan diubah menjadi vektor fitur berukuran $1 \times 1 \times C$ atau vektor satu dimensi sepanjang C . Penggunaan GAP juga membantu mengurangi jumlah parameter pada model, sehingga model menjadi lebih efisien serta dapat mengurangi risiko *overfitting*. *Output* dari lapisan ini kemudian digunakan sebagai *input* pada lapisan *fully connected* (*dense layer*) untuk proses klasifikasi digit pada tahap selanjutnya.

3.2.2.4 Fully Connected Layer

Pada tahap *Fully Connected Layer*, vektor fitur hasil proses GAP akan digunakan untuk melakukan proses klasifikasi. Lapisan ini bertujuan untuk

mengolah representasi fitur yang sudah diekstraksi oleh MobileNetV2 *Feature Extractor* sehingga dapat menentukan kelas digit yang paling sesuai. Dalam arsitektur yang diterapkan pada penelitian ini, lapisan klasifikasi terdiri dari dua *dense layer* yang disusun secara bertahap untuk mempelajari hubungan antar fitur sebelum menghasilkan prediksi akhir.

Secara umum, operasi pada *Fully Connected Layer* dapat dinyatakan sebagai transformasi linear antara vektor *input* dan bobot jaringan, yang dirumuskan sebagai berikut:

$$z = W_x + b \quad (3.10)$$

Keterangan:

x : Vektor *input*.

W : Matrix bobot (*weight*).

b : Bias.

z : Vektor *output* sebelum fungsi aktivasi.

Lapisan pertama pada arsitektur ini adalah *dense layer* dengan 128 neuron yang berfungsi untuk mempelajari kombinasi fitur kompleks dari vektor fitur hasil ekstraksi. Pada lapisan ini juga diterapkan *batch normalization* guna menjaga kestabilan distribusi aktivasi selama pelatihan dilakukan serta mempercepat konvergensi model. Selain itu, teknik *dropout* juga diterapkan sebagai strategi untuk meminimalkan *overfitting* melalui menonaktifkan sejumlah neuron secara acak selama proses pelatihan.

Selanjutnya, fitur yang dihasilkan dilanjutkan ke *dense layer* kedua dengan 64 neuron yang berfungsi untuk melakukan pemetaan fitur lebih lanjut sebelum proses klasifikasi akhir. Pada lapisan ini juga diterapkan *dropout* untuk menjaga kemampuan generalisasi model. Melalui dua lapisan *dense* tersebut, model dapat

mempelajari representasi fitur yang lebih diskriminatif sehingga mampu membedakan setiap digit dengan lebih baik. *Output* dari *layer* ini selanjutnya akan diteruskan ke *Softmax* sebagai lapisan klasifikasi akhir untuk menentukan probabilitas masing-masing kelas digit.

3.2.2.5 *Softmax*

Pada tahap *softmax*, *output* dari *fully connected layer* akan diubah menjadi nilai probabilitas untuk setiap kelas digit. Lapisan ini digunakan sebagai fungsi aktivasi pada tahap akhir model klasifikasi karena mampu mengganti nilai keluaran model menjadi bentuk distribusi probabilitas yang memiliki rentang 0 hingga 1, dengan total probabilitas seluruh kelas bernilai 1. Proses perhitungan probabilitas pada fungsi *softmax* mengikuti rumus yang telah dijelaskan pada persamaan 2.12. Sehingga, model dapat menentukan kelas digit yang paling mungkin berdasarkan nilai probabilitas tertinggi.

Dalam penelitian ini, jumlah kelas yang digunakan adalah 10 kelas digit, yaitu angka 0 hingga 9. Oleh karena itu, lapisan *softmax* akan menghasilkan vektor probabilitas berukuran 10, di mana setiap elemen merepresentasikan kemungkinan citra *input* tergolong pada kelas digit tertentu.

3.2.2.6 Proses Pelatihan

Pada tahap ini dilakukan pelatihan model guna mempelajari pola dan ciri digit pada citra. Pelatihan model dilaksanakan dengan dua tahap, yaitu pelatihan awal (*pretraining*) dan *fine-tuning*. Pada tahap pelatihan awal, model MobileNetV2 dilatih dengan data latih dan data validasi yang berasal dari *dataset* DIGITS.

Dataset ini digunakan karena memiliki variasi bentuk dan gaya penulisan digit yang cukup beragam sehingga dapat membantu model dalam mempelajari karakteristik dasar dari setiap angka. Selama proses pelatihan, model akan meng-*update* bobot jaringan secara berulang dengan memanfaatkan data latih, sedangkan data validasi digunakan untuk memonitoring kinerja model serta mencegah terjadinya *overfitting*.

Setelah model memperoleh pemahaman dasar terhadap pola digit, tahap selanjutnya adalah *fine-tuning* dengan data digit yang berasal dari pengambilan di lapangan. Pada tahap ini, data lapangan yang sudah melewati proses augmentasi dan *preprocessing* dipisah menjadi data latih dan data validasi untuk menyelaraskan model dengan ciri digit pada meteran air yang sebenarnya. *Fine-tuning* berfungsi untuk menambah kecakapan model dalam mengenali digit yang memiliki kondisi pencahayaan, *noise*, serta bentuk digit yang berbeda dibandingkan dengan *dataset* DIGITS. Sehingga, model diharapkan dapat bekerja lebih optimal ketika diimplementasikan pada sistem aplikasi *mobile* OCR.

3.2.2.7 Evaluasi Model

Tahap ini dilakukan setelah proses *pretraining* dan *fine-tuning* dilakukan. Evaluasi dijalankan menggunakan data uji yang tidak dilibatkan selama proses pelatihan model. Dalam penelitian ini, data uji terdiri dari data uji yang berasal dari dataset DIGITS serta data uji yang berasal dari pengambilan di lapangan. Penggunaan kedua jenis data uji tersebut berguna untuk mengetahui kecakapan model dalam mengenali digit baik pada *dataset* umum maupun pada citra digit meteran air yang diperoleh dari kondisi nyata.

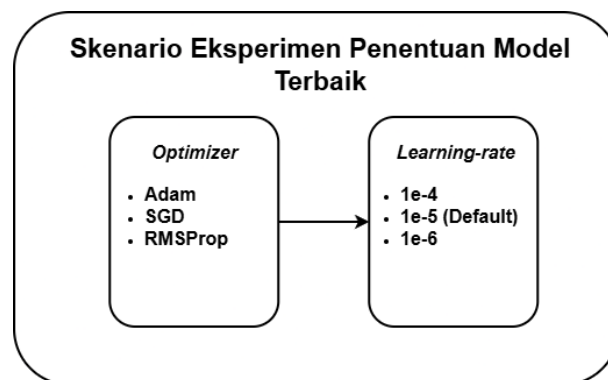
Kinerja model selanjutnya dianalisis dengan sejumlah metrik evaluasi klasifikasi, yaitu akurasi, presisi, *recall*, dan *F1-score*. Setelah evaluasi selesai dilakukan dan model menghasilkan performa yang baik, model tersebut kemudian disimpan (*saved model*). Model yang telah disimpan selanjutnya diimplementasikan ke dalam aplikasi *mobile* OCR agar dapat digunakan untuk mengklasifikasikan digit secara otomatis.

3.2.3 Implementasi Skenario Eksperimen Penentuan Model Terbaik

Skenario eksperimen dirancang untuk memperoleh model terbaik yang akan diintegrasikan pada aplikasi *mobile* OCR guna melakukan pembacaan meteran air. Tahap pertama difokuskan pada perbandingan performa beberapa *optimizer*, yaitu *Adam*, *SGD*, dan *RMSProp*, dengan menggunakan *learning rate* yang sama pada fase *fine-tuning*, untuk menentukan metode optimasi yang paling efektif dalam mempercepat konvergensi dan meningkatkan akurasi model.

Optimizer dengan hasil performa terbaik selanjutnya digunakan pada pengujian variasi *learning rate* yang meliputi 0,0001 (1e-4), 0,00001 (1e-5), dan 0,000001 (1e-6) guna menemukan nilai laju pembelajaran yang paling optimal dalam proses *training*. Variasi *learning rate* tersebut hanya diterapkan pada fase *fine-tuning* menggunakan *dataset* lapangan, sedangkan pada tahap *pretraining* seluruh skenario menggunakan *learning rate* tetap sebesar 0,001 (1e-3). Perbedaan penggunaan *learning rate* ini dilakukan untuk menghindari perubahan bobot yang terlalu besar pada model yang telah dipelajari sebelumnya, sehingga tidak merusak representasi fitur awal yang sudah terbentuk dan memungkinkan proses

penyesuaian yang lebih stabil terhadap data lapangan. Seluruh proses pelatihan pada setiap skenario menggunakan jumlah *epoch* maksimum sebanyak 20. Setiap tahap pengujian dilakukan dengan mempertimbangkan metrik evaluasi, yaitu *accuracy*, *precision*, *recall*, serta *f1-score*, sehingga diperoleh konfigurasi model yang paling optimal untuk diterapkan pada aplikasi *mobile* OCR pembacaan meteran air. Secara keseluruhan, tahapan skenario uji coba dalam penentuan model terbaik divisualisasikan pada Gambar 3. 16.



Gambar 3. 16 Skenario Eksperimen Penentuan Model Terbaik

1. Skenario 1 menggunakan *optimizer Adam* dengan *learning rate* 1e-5 pada fase *fine-tuning*.
2. Skenario 2 menggunakan *optimizer SGD* dengan *learning rate* 1e-5 pada fase *fine-tuning*.
3. Skenario 3 menggunakan *optimizer RMSProp* dengan *learning rate* 1e-5 pada fase *fine-tuning*.
4. Skenario 4 menggunakan *optimizer* terbaik dari skenario sebelumnya dengan *learning rate* 1e-5 pada fase *fine-tuning*.
5. Skenario 5 menggunakan *optimizer* terbaik dengan *learning rate* 1e-4 pada fase *fine-tuning*.

6. Skenario 6 menggunakan *optimizer* terbaik dengan *learning rate* $1e-6$ pada fase *fine-tuning*.

Seluruh skenario eksperimen penentuan model terbaik dalam penelitian ini menggunakan arsitektur MobileNetV2 dengan konfigurasi lapisan klasifikasi yang dibuat konsisten pada setiap percobaan, yaitu *hidden layer* pertama sebanyak 128 *node* dengan fungsi aktivasi ReLU, *hidden layer* kedua sebanyak 64 *node* dengan fungsi aktivasi ReLU, serta *output layer* sebanyak 10 *node* dengan fungsi aktivasi *softmax* yang merepresentasikan kelas digit 0 hingga 9. Selain itu, seluruh skenario juga menerapkan mekanisme regularisasi yang identik, meliputi *batch normalization* dan *dropout* sebesar 0,3 pada *hidden layer* pertama serta *dropout* sebesar 0,2 pada *hidden layer* kedua, guna mencegah *overfitting* selama proses pelatihan.

Seluruh skenario juga dikonfigurasi dengan parameter pelatihan yang seragam. Fungsi *loss* yang digunakan ialah *sparse categorical crossentropy*. Jumlah *epoch* maksimum ditetapkan sebesar 20 *epoch* untuk kedua fase pelatihan. *early stopping* juga digunakan dengan *patience* sebesar 5 *epoch* yang memonitoring nilai *validation loss*. Apabila *validation loss* tidak membaik selama 5 *epoch* terus-menerus, pelatihan dihentikan secara otomatis untuk mencegah *overfitting*. Selain *early stopping*, seluruh skenario juga menerapkan *ReduceLROnPlateau*, yaitu mekanisme yang secara otomatis mereduksi *learning rate* sebesar faktor 0,3 apabila *validation loss* tidak membaik selama 3 *epoch* terus-menerus, dengan batas terendah *learning rate* sebesar $1e-6$. Mekanisme ini bertujuan untuk membantu model keluar dari kondisi stagnasi selama pelatihan tanpa harus menghentikan

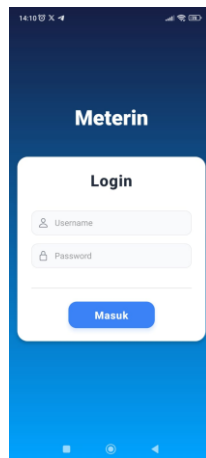
proses secara keseluruhan. Ukuran *batch* yang digunakan pada seluruh skenario adalah 16.

3.2.4 Implementasi Aplikasi *Mobile* OCR

Bagian ini menyajikan hasil implementasi aplikasi yang telah dikembangkan, mencakup realisasi antarmuka pengguna berdasarkan desain yang telah dirancang. Seluruh tampilan dan konfigurasi yang ditampilkan bertujuan untuk memvalidasi keselarasan antara desain yang mendukung operasional sistem secara keseluruhan.

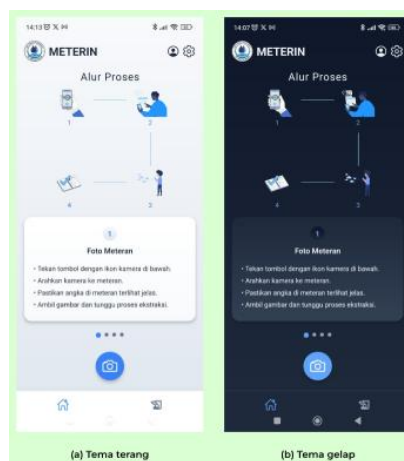
1. Halaman *Login*

Halaman *login* pada aplikasi ditampilkan pada Gambar 3. 17. Halaman ini berfungsi sebagai mekanisme autentikasi untuk melakukan verifikasi identitas *user* sebelum diberikan hak akses ke dalam sistem. Pada tahap ini, pengguna harus memasukkan *username* dan *password* yang telah terdaftar sebagai kredensial autentikasi. Ketika tombol "Masuk" ditekan, sistem akan memvalidasi terhadap data yang diinputkan dengan mencocokkan kredensial tersebut terhadap data yang ada di *database*. Apabila kredensial sesuai, maka *user* akan ditujukan ke halaman beranda aplikasi. Sebaliknya, apabila kredensial tidak cocok, maka sistem akan menolak akses dan meminta *user* untuk menginputkan kembali data yang benar. Halaman ini merupakan bagian penting dari sistem keamanan aplikasi untuk memastikan hanya *user* yang telah terdaftar yang dapat menggunakan aplikasi.

Gambar 3. 17 Halaman *Login*

2. Halaman Beranda

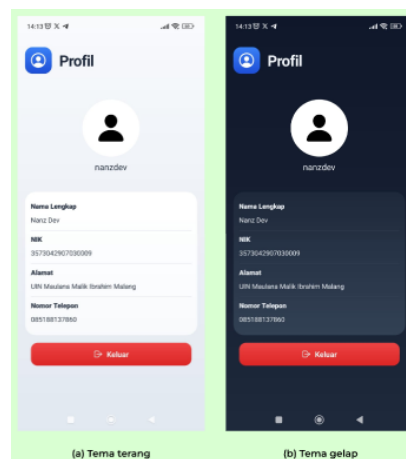
Gambar 3. 18 menampilkan halaman beranda pada aplikasi berfungsi sebagai *dashboard* utama yang memandu pengguna melalui alur kerja inti sistem. Pada bagian atas, header menampilkan logo, nama aplikasi, serta tombol akses ke profil dan pengaturan, yang berfungsi untuk navigasi cepat ke manajemen akun serta konfigurasi preferensi aplikasi. Bagian tengah memuat infografis urutan proses *input* baca meter yang dilengkapi dengan kartu keterangan yang berisi instruksi teknis yang harus dijalankan pengguna. Di bagian bawah, terdapat *navbar* navigasi yang berfungsi pindah antara halaman beranda dan histori.



Gambar 3. 18 Halaman Beranda

3. Halaman Profil

Halaman profil pada Gambar 3. 19 berperan sebagai antarmuka untuk memberikan informasi data diri *user* yang tersimpan di dalam sistem. Halaman ini menyajikan identitas akun berupa *username* di bawah avatar, serta rincian data pribadi yang meliputi nama lengkap, Nomor Induk Kependudukan (NIK), alamat, dan nomor telepon, yang diambil langsung dari *database*. Selain sebagai pusat informasi, halaman ini juga menyediakan tombol 'Keluar' untuk memungkinkan pengguna mengakhiri sesi mereka secara aman dan akan dikembalikan ke halaman *login* apabila diklik.

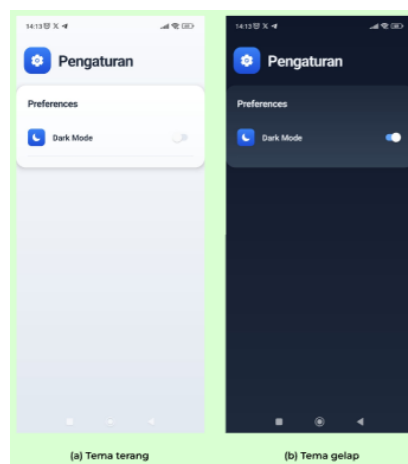


Gambar 3. 19 Halaman Profil

4. Halaman Pengaturan

Halaman pengaturan pada Gambar 3. 20 menyediakan fitur untuk mengelola preferensi tampilan aplikasi melalui opsi "*Dark Mode*" yang terletak pada bagian *Preferences*. Fitur ini didesain untuk menghasilkan fleksibilitas kepada *user* dalam menyesuaikan tema aplikasi sesuai dengan kenyamanan visual dan preferensi pribadi, sekaligus memberikan kesan yang lebih modern pada antarmuka aplikasi. Ketika *toggle dark mode* diaktifkan, sistem akan secara otomatis mengubah skema

warna pada sebagian besar halaman aplikasi dari tema terang (*light theme*) menjadi tema gelap (*dark theme*), seperti yang ditunjukkan pada perbedaan tampilan antara gambar (a) dan (b).

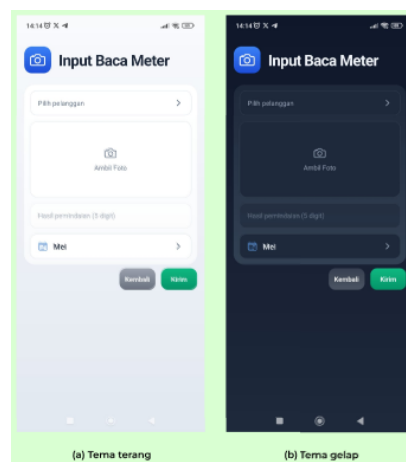


Gambar 3. 20 Halaman Pengaturan

5. Halaman *Input* Baca Meter

Gambar 3. 21 menampilkan halaman *input* baca meter. berfungsi sebagai antarmuka utama untuk melakukan pencatatan penggunaan air pelanggan. Alur kerja dimulai dengan pengguna memilih target pelanggan pencatatan melalui kolom "Pilih pelanggan" yang akan mengalihkan ke halaman daftar pelanggan. Selanjutnya, area "Ambil Foto" mengaktifkan kamera perangkat guna mengambil gambar digit meteran air pelanggan. Setelah gambar tersebut diambil maka model akan memprosesnya, dan setelah selesai, hasil pemindaian akan otomatis muncul pada kolom hasil pemindaian. Kolom tersebut awalnya bersifat *read-only*, namun akan menjadi dapat diedit setelah prediksi keluar untuk memungkinkan koreksi manual jika hasil deteksi model OCR kurang akurat. *Field* pemilihan bulan tagihan secara otomatis terisi sesuai bulan berjalan guna mempercepat proses *input*. Apabila seluruh data telah diverifikasi, pengguna dapat menekan tombol "Kirim" untuk

menyimpan data pada *database (Convex)* sekaligus meneruskannya ke sistem web eksternal atau memilih tombol "Kembali" untuk membatalkan proses dan keluar dari halaman tersebut.

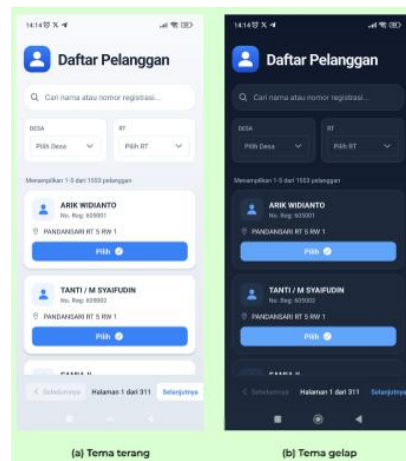


Gambar 3. 21 Halaman Input Baca Meter

6. Halaman Daftar Pelanggan

Halaman daftar pelanggan pada Gambar 3. 22 berperan sebagai antarmuka untuk menyajikan dan memilih data pelanggan yang akan dilakukan pencatatan meteran, di mana data pelanggan diambil langsung dari *database*. Pada bagian atas halaman, tersedia mekanisme penyaringan data melalui kolom pencarian berbasis nama atau nomor registrasi, serta *dropdown* filter untuk wilayah "Desa" dan "RT". Ketika filter diisi atau diaktifkan, sistem akan secara otomatis melakukan *query* ulang dan hanya menampilkan data pelanggan yang relevan dengan kriteria yang ditentukan. Setiap entri dalam daftar dilengkapi tombol "Pilih" yang, apabila diklik, akan mengembalikan navigasi ke halaman *input* baca meter sekaligus membawa data identitas pelanggan yang telah terpilih untuk mengisi *form* pencatatan. Di bagian bawah, terintegrasi komponen paginasi yang membatasi jumlah data yang dimuat per halaman. Mekanisme ini sengaja diterapkan untuk mengoptimalkan

proses pengambilan data secara bertahap, sehingga dapat mengurangi frekuensi permintaan ke *database* berbasis *cloud* (*Convex*) dan secara efektif menekan biaya operasional serta penggunaan *bandwidth*.



Gambar 3. 22 Halaman Daftar Pelanggan

7. Halaman Ambil Gambar

Gambar 3. 23 menampilkan halaman ambil gambar. Halaman ini berfungsi sebagai antarmuka khusus untuk mengambil dan mengekstraksi digit angka pada meteran air melalui pemotretan langsung. Antarmuka ini dilengkapi dengan *overlay* tetap berupa lima bingkai kotak yang berfungsi sebagai panduan segmentasi visual, di mana pengguna diharuskan mengarahkan kamera agar kelima digit pada meteran tepat berada di dalam masing-masing kotak. Ketika tombol kamera ditekan, sistem secara otomatis melakukan *cropping* pada area yang tercakup oleh bingkai tersebut sesuai dimensinya, sehingga menghasilkan lima potongan gambar berisi digit tunggal yang telah terisolasi dari latar belakang di luar area tangkapan. Potongan gambar tersebut kemudian diolah secara independen oleh model OCR guna menghasilkan prediksi per digit. Setelah proses inferensi selesai, hasil prediksi dari kelima digit akan digabungkan secara berurutan, secara otomatis mengisi kolom

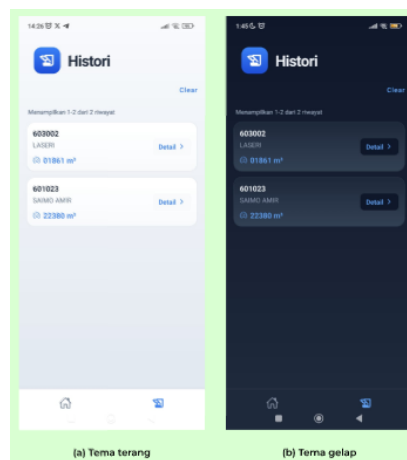
hasil pemindaian, dan sistem akan mengalihkan navigasi kembali ke halaman *input* baca meter untuk proses verifikasi serta penyimpanan data selanjutnya.



Gambar 3. 23 Halaman Ambil Gambar

8. Halaman Histori

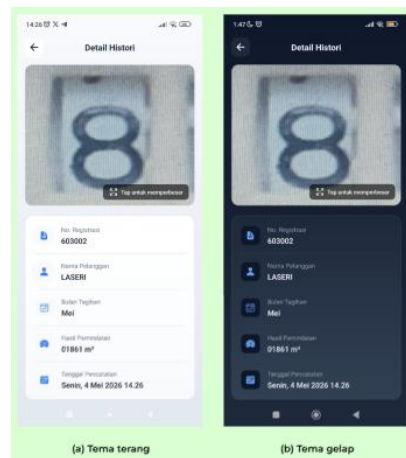
Halaman histori pada Gambar 3. 24 berfungsi untuk merekam dan menampilkan riwayat aktivitas pencatatan meteran yang telah diselesaikan oleh petugas. Pada bagian atas, tersedia tombol "*Clear*" yang memungkinkan pengguna menghapus seluruh entri histori. Daftar histori disajikan dalam bentuk *card* yang memuat informasi ringkas setiap transaksi, meliputi nomor registrasi, nama pelanggan, serta nilai digit hasil pemindaian, di mana setiap data histori pada *card* dilengkapi tombol "*Detail*" yang akan mengarahkan *user* kepada halaman detail histori untuk melihat informasi pencatatan secara lebih rinci.



Gambar 3. 24 Halaman Histori

9. Halaman Detail Histori

Halaman detail histori pada Gambar 3. 25 berfungsi sebagai antarmuka untuk menampilkan informasi lengkap dari setiap aktivitas pencatatan meteran yang telah tersimpan dalam *database*. Bagian atas halaman menyajikan hasil tangkapan gambar pada saat proses pendataan berlangsung, yang dilengkapi interaksi perbesaran untuk memudahkan pengecekan visual terhadap angka meteran asli. Di bawahnya, sistem menampilkan rincian data transkrip yang meliputi nomor registrasi, nama pelanggan, periode bulan tagihan, nilai hasil pemindaian dalam satuan meter kubik, serta waktu tanggal dan jam pencatatan dilakukan.



Gambar 3. 25 Halaman Detail Histori

3.2.5 Implementasi Evaluasi Sistem Aplikasi

Pengujian dilakukan menggunakan pendekatan *blackbox testing*. Pendekatan ini digunakan karena penelitian berfokus pada pengujian kesesuaian fungsi aplikasi terhadap kebutuhan *user* dan alur bisnis yang telah dirancang. Untuk menjamin semua fitur pada aplikasi bekerja sesuai dengan perilaku yang diharapkan, disusun skenario pengujian sebagai acuan dalam pelaksanaan *blackbox testing*. Skenario pengujian tersebut dibuat berdasarkan kebutuhan fungsional serta alur bisnis yang telah diimplementasikan pada aplikasi. Rincian skenario pengujian *blackbox testing* dapat dilihat pada Tabel 3. 5.

Tabel 3. 5 Skenario *Blackbox Testing*

ID	USE CASE	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
TC01	UC01	<i>Login Pengguna</i>	Pengguna memasukkan username beserta password yang benar.	Sistem berhasil memverifikasi identitas user dan halaman beranda.
TC02	UC01	<i>Login Gagal</i>	Pengguna menginputkan username atau password yang salah.	Sistem memunculkan notifikasi kesalahan dan tetap berada di halaman login.
TC03	UC01	<i>Login Form Kosong</i>	Pengguna tidak mengisi <i>username</i> maupun <i>password</i>	Sistem memunculkan notifikasi kesalahan dan tetap berada di halaman <i>login</i>

ID	USE CASE	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
TC04	UC02	Akses Kamera	Pengguna menuju halaman input baca meter dan menekan tombol ambil gambar	Kamera terbuka dan siap digunakan untuk mengambil foto digit angka jumlah pemakaian pada meteran air.
TC05	UC02	Ambil Gambar Meteran	Pengguna mengambil foto meteran air dengan jelas.	Gambar tersimpan sementara dan hasil pemindaian OCR ditampilkan pada halaman input baca meter
TC06	UC03	Verifikasi Digit	Pengguna membandingkan hasil OCR dengan digit angka pada meteran	Pengguna dapat menyetujui hasil atau melakukan koreksi manual
TC07	UC04	Penyimpanan Data	Hasil pembacaan disimpan.	Data tersimpan ke <i>database</i> dan muncul pada daftar yang berada di halaman histori.
TC08	UC04	Pengiriman Data <i>Input</i> Baca Meter	Data <i>input</i> baca meter dikirim ke sistem web eksternal	Sistem aplikasi <i>mobile</i> berhasil mengirimkan data <i>input</i> baca meter ke sistem web eksternal.
TC09	UC05	Tampilkan Histori	Pengguna membuka menu histori.	Daftar pembacaan yang telah disimpan tampil dengan benar.
TC10	UC05	Bersihkan Histori	Pengguna menekan tombol <i>clear</i> .	Catatan histori terhapus dan daftar menjadi kosong.
TC11	UC05	Akses Detail Histori	Pengguna menekan tombol detail pada daftar data histori	Detail histori tampil dengan menampilkan informasi yang lebih rinci.
TC12	UC06	Perubahan Tema	Pengguna memilih tema terang/gelap di menu pengaturan.	Tema aplikasi berubah sesuai pilihan pengguna.
TC13	UC07	Tampilkan Profil	Pengguna membuka menu profil.	Informasi pengguna tampil sesuai data yang terdaftar.
TC14	UC08	Tampilkan Data Pelanggan	Pengguna membuka menu daftar pelanggan	Daftar data pelanggan PAMDes Pandanrejo tampil sesuai data yang terdaftar
TC15	UC08	Filter Pencarian Pelanggan	Pengguna menggunakan filter pencarian untuk mencari pelanggan.	Daftar data pelanggan yang muncul hanya berdasarkan dengan yang dicari
TC16	UC08	Penggunaan Filter Desa	Pengguna menggunakan filter pilihan desa untuk mencari pelanggan sesuai desa tertentu	Daftar data pelanggan yang muncul hanya berdasarkan dengan desa yang sesuai
TC17	UC08	Penggunaan Filter RT	Pengguna menggunakan filter pilihan rt untuk mencari	Daftar data pelanggan yang muncul hanya

ID	USE CASE	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
			pelanggan sesuai rt tertentu	berdasarkan dengan rt yang sesuai
TC18	UC08	Pilih Pelanggan	Pengguna memilih pelanggan dengan menekan tombol pilih pada daftar pelanggan	Pengguna kembali halaman <i>input</i> baca meter dengan membawa informasi pelanggan
TC19	UC09	<i>Logout</i>	Pengguna menekan tombol keluar.	Sistem mengakhiri sesi user, kemudian mengarahkan kembali ke halaman <i>login</i> .

Selain pengujian *blackbox* untuk fungsionalitas aplikasi, dilakukan pula evaluasi performa model OCR menggunakan akurasi OCR. Pengujian ini dilakukan dengan membandingkan hasil pembacaan digit oleh model OCR terhadap nilai asli pada meteran air pelanggan di lapangan.

Untuk memperjelas proses evaluasi akurasi OCR, dilakukan simulasi perhitungan dengan membandingkan hasil pembacaan model terhadap nilai asli yang telah diketahui. Simulasi tersebut ditunjukkan pada Tabel 3. 6.

Tabel 3. 6 Simulasi Evaluasi Model OCR

No.	Nilai Asli	Hasil Pembacaan OCR
1.	01634	01634
2.	02375	02375
3.	03459	03459
4.	10234	10234
5.	12846	1284
6.	03452	03452
7.	20123	20123
8.	21234	21234
9.	22375	2375
10.	30122	30122

Berdasarkan Tabel 3. 6 setiap data pada tabel terdiri dari 5 digit angka, sehingga total digit angka nilai asli yang diuji adalah 50. Pada kolom hasil pembacaan OCR dapat diketahui bahwa terdapat dua kesalahan pembacaan berupa digit yang tidak terdeteksi, sehingga jumlah digit yang berhasil dikenali dengan

benar adalah sebanyak 48 digit angka. Sehingga nilai akurasi OCR dengan menggunakan persamaan 2.17 dapat dihitung:

$$Akurasi\ OCR = \frac{48}{50} \times 100\% = 96\% \quad (3.11)$$

Hasil perhitungan tersebut menunjukkan bahwa model OCR yang dikembangkan mampu mengenali digit angka pada meteran air dengan tingkat akurasi sebesar 96%.

3.2.6 *Experimental Setup*

Pada penelitian ini, proses pengembangan sistem dilakukan pada dua lingkungan komputasi yang berbeda, yaitu komputer dengan spesifikasi cukup tinggi yang dimanfaatkan untuk proses pelatihan model, serta laptop pribadi yang digunakan untuk pengembangan dan implementasi aplikasi *mobile*. Pemisahan lingkungan ini dilakukan untuk mengoptimalkan kinerja pada masing-masing tahap, di mana proses pelatihan model membutuhkan jumlah besar terhadap sumber daya komputasi sehingga tidak memungkinkan dilakukan secara optimal pada laptop pribadi karena keterbatasan spesifikasi perangkat. Sementara itu, pengembangan aplikasi difokuskan pada laptop karena lebih mendukung proses integrasi dan pengujian sistem secara keseluruhan.

Spesifikasi perangkat komputer yang digunakan pada pelatihan model disajikan pada Tabel 3. 7. Perangkat ini dilengkapi dengan GPU yang mumpuni. Dengan dukungan spesifikasi tersebut, proses *training* dan *fine-tuning* dapat dilakukan secara lebih efisien dalam waktu yang relatif singkat.

Tabel 3. 7 Tabel Spesifikasi Komputer untuk Pelatihan Model

No.	Komponen	Spesifikasi
1	CPU	Intel Core i5-12400F (2.50 GHz)
2	GPU	NVIDIA GeForce RTX 3070 Ti
3	RAM	32 GB
4	STORAGE	SSD ± 22 TB
5	OS	Windows 11 64-bit

Spesifikasi laptop yang dimanfaatkan untuk pengembangan aplikasi *mobile* disajikan pada Tabel 3. 8. Perangkat ini dipergunakan dalam proses implementasi, integrasi sistem, serta *debugging* berdasarkan hasil pengujian, seperti pengujian antarmuka pengguna, fungsionalitas fitur, dan integrasi model ke dalam aplikasi apabila masih ditemukan kesalahan. Meskipun memiliki spesifikasi yang lebih rendah dibandingkan komputer untuk pelatihan model, laptop ini sudah memadai untuk menjalankan *tools* pengembangan seperti *React Native* dan *Expo*.

Tabel 3. 8 Tabel Spesifikasi Laptop untuk Pengembangan Aplikasi

No.	Komponen	Spesifikasi
1	CPU	Intel Core i5-1135G7 (2.40 GHz)
2	GPU	NVIDIA GeForce MX350, Intel Iris Xe Graphics
3	RAM	20 GB
4	STORAGE	SSD 256 GB dan HDD 1 TB
5	OS	Windows 11 64-bit

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

Pada bab ini dijelaskan hasil dari penelitian yang telah dilakukan. Hasil penelitian meliputi proses pengujian skenario eksperimen penentuan model terbaik, pengujian fungsionalitas sistem dengan menggunakan *blackbox testing*, serta evaluasi akurasi sistem OCR untuk mengetahui kemampuan model dalam mengenali digit angka pada meteran air pelanggan.

4.1.1 Hasil Pengujian Skenario Penentuan Model Terbaik

Pada penelitian ini dilakukan perbandingan hasil pengujian dari beberapa skenario pelatihan model yang telah dirancang sebelumnya. Perbandingan difokuskan pada penggunaan *optimizer* dan variasi *learning rate* terhadap performa model. Setiap skenario dievaluasi menggunakan metrik evaluasi untuk mengidentifikasi konfigurasi model yang paling optimal.

4.1.1.1 Perbandingan *Optimizer*

Pada tahap ini dilakukan perbandingan penggunaan beberapa *optimizer*, antara lain *Adam*, *SGD*, dan *RMSProp*, dengan nilai parameter *learning rate* pada fase *fine-tuning* sebesar $1e-5$ serta kuantitas *epoch* sebanyak 20 pada masing-masing skenario. Berdasarkan Tabel 4. 1, diperoleh bahwa *optimizer Adam* memiliki waktu pelatihan tercepat, yaitu selama 2 menit 9 detik, dibandingkan

dengan *SGD* selama 2 menit 25 detik dan *RMSProp* selama 2 menit 33 detik. Hasil ini menunjukkan bahwa *Adam* mampu mencapai konvergensi lebih cepat dibandingkan *optimizer* lainnya.

Tabel 4. 1 Hasil *Training* Berdasarkan *Optimizer*

No.	Skenario	<i>Optimizer</i>	<i>Learning rate</i>	<i>Epoch</i>	Waktu
1	Skenario 1	<i>Adam</i>	1e-5	20	2m 9d
2	Skenario 2	<i>SGD</i>	1e-5	20	2m 25d
3	Skenario 3	<i>RMSProp</i>	1e-5	20	2m 33d

Hasil evaluasi penggunaan *optimizer* terhadap performa model terdapat pada Tabel 4. 2. Berdasarkan tabel tersebut, dapat dilihat bahwa *optimizer Adam* pada Skenario 1 menghasilkan performa terbaik, yaitu memperoleh nilai *accuracy* 96,51%, *precision* 96,63%, *recall* 96,51%, dan *F1-score* 96,52%.

Sementara itu, *optimizer RMSProp* pada Skenario 3 menghasilkan performa yang cukup baik dengan *accuracy* 93,91%, namun masih berada di bawah *Adam*. Adapun *optimizer SGD* pada Skenario 2 menghasilkan performa terendah dengan *accuracy* 90,95%, yang mengindikasikan bahwa model masih memiliki tingkat kesalahan klasifikasi yang lebih banyak dibandingkan skenario lainnya. Berdasarkan hasil tersebut, dapat disimpulkan bahwa *optimizer Adam* memberikan kinerja paling optimal dibandingkan *optimizer* lainnya, sehingga dipilih untuk digunakan pada tahap selanjutnya.

Tabel 4. 2 Hasil Evaluasi Berdasarkan *Optimizer*

No.	Skenario	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
1	Skenario 1	96.51%	96,63%	96,51%	96,52%
2	Skenario 2	90.95%	93,25%	90,95%	91,38%
3	Skenario 3	93,91%	94,37%	93,91%	93,95%

4.1.1.2 Perbandingan *Learning Rate*

Pada tahap ini dilakukan perbandingan variasi *learning rate* pada fase *fine-tuning*, yaitu $1e-5$, $1e-4$, dan $1e-6$, dengan menggunakan *optimizer* terbaik yang diperoleh pada tahap sebelumnya, yaitu *Adam*. Setiap skenario diuji dengan jumlah *epoch* yang sama, yaitu 20, untuk memastikan perbandingan yang adil. Berdasarkan Tabel 4. 3, diperoleh bahwa penggunaan *learning rate* $1e-4$ pada Skenario 5 menghasilkan waktu pelatihan tercepat, yaitu selama 2 menit 4 detik, diikuti oleh *learning rate* $1e-5$ pada Skenario 4 dengan waktu 2 menit 9 detik, serta *learning rate* $1e-6$ pada Skenario 6 dengan waktu paling lama, yaitu 2 menit 26 detik. Hasil ini mengindikasikan bahwa nilai *learning rate* lebih besar cenderung mempercepat proses konvergensi model, sedangkan nilai *learning rate* yang kecil membutuhkan waktu pelatihan yang lebih lama. Oleh sebab itu, variasi nilai *learning rate* memberikan pengaruh terhadap efisiensi waktu pelatihan model.

Tabel 4. 3 Hasil Training berdasarkan *Learning Rate*

No.	Skenario	Optimizer	Learning rate	Epoch	Waktu
1	Skenario 4	Adam	$1e-5$	20	2m 9d
2	Skenario 5	Adam	$1e-4$	20	2m 4d
3	Skenario 6	Adam	$1e-6$	20	2m 26d

Hasil evaluasi performa model berdasarkan variasi *learning rate* terdapat pada Tabel 4. 4. Berdasarkan tabel tersebut, *learning rate* $1e-5$ pada Skenario 4 menghasilkan kinerja terbaik dengan nilai *accuracy* 96,51%, *precision* 96,63%, *recall* 96,51%, dan *f1-score* 96,52%. Sementara itu, *learning rate* $1e-4$ pada Skenario 5 mengindikasikan performa yang sangat mendekati dengan *accuracy* 95,34%, sedangkan *learning rate* $1e-6$ pada Skenario 6 menghasilkan kinerja terendah. Hal ini mengindikasikan bahwa waktu pelatihan yang lebih lama tidak

selalu menjamin peningkatan performa model, terbukti pada *learning rate* $1e-6$ yang memiliki waktu pelatihan paling lama namun menghasilkan akurasi yang paling rendah. Perbedaan hasil yang relatif tipis antara *learning rate* $1e-5$ dan $1e-4$ mengindikasikan bahwa kedua nilai tersebut berada pada rentang yang cukup optimal dalam proses pembelajaran model, di mana model mampu mencapai konvergensi dengan baik. Oleh sebab itu, *learning rate* $1e-5$ dinobatkan sebagai nilai yang paling optimal karena memberikan performa terbaik.

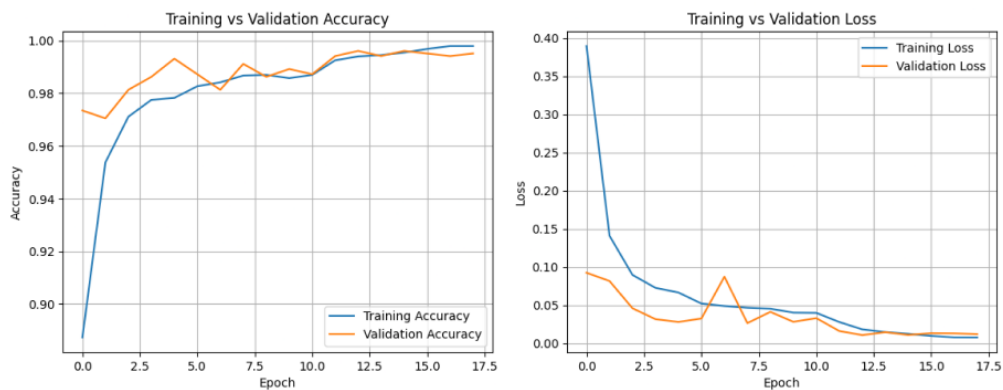
Tabel 4. 4 Hasil Evaluasi Berdasarkan *Learning Rate*

No.	Skenario	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
1	Skenario 4	96.51%	96,63%	96,51%	96,52%
2	Skenario 5	95.34%	95,80%	95,34%	95,32%
3	Skenario 6	88,89%	90,86%	88,89%	89,27%

4.1.1.3 Hasil Skenario 1

Pada skenario ini digunakan *optimizer Adam* dengan *learning rate* sebesar $1e-3$ pada fase *pretraining* dan $1e-5$ pada fase *fine-tuning*, serta jumlah *epoch* sebanyak 20. Berdasarkan Gambar 4. 1 yang menunjukkan hasil *pretraining* menggunakan *dataset DIGITS*, terlihat bahwa grafik akurasi *training* maupun *validation* mengalami kenaikan yang besar pada beberapa awal *epoch*, lalu cenderung stabil hingga akhir pelatihan. Nilai *training accuracy* tertinggi menggapai 99,79%, sedangkan *validation accuracy* tertinggi menggapai 99,61%, dengan selisih (*gap*) sebesar 0,18%. *Gap* yang relatif kecil menandakan bahwa model memiliki kecakapan generalisasi yang baik dan tidak terjadi *overfitting* signifikan. Selain itu, grafik *loss* memperlihatkan penurunan yang curam pada awal pelatihan dan terus berkurang hingga menggapai *training loss* terendah senilai 0,0077 dan *validation loss* terendah senilai 0,0108, dengan *gap* sebesar 0,0031.

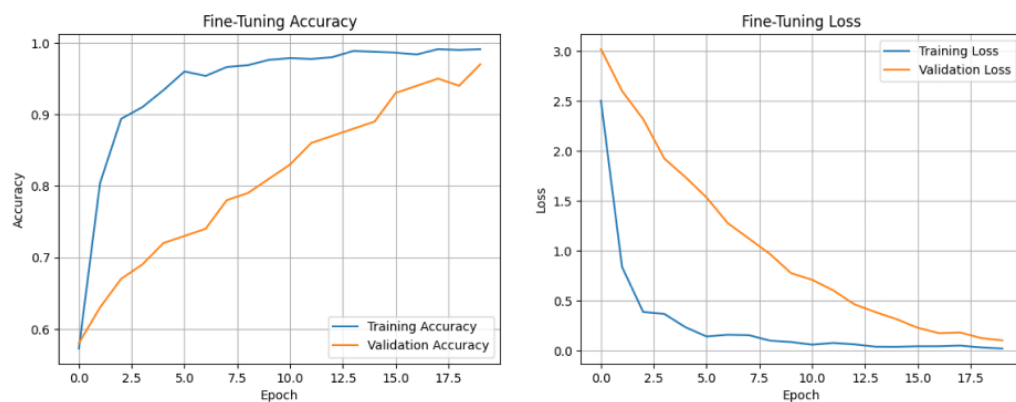
Nilai *loss* rendah serta selisih nilai kecil antara *training* dan *validation* mengindikasikan kesalahan prediksi model semakin kecil seiring bertambahnya *epoch* dan proses pembelajaran berlangsung secara stabil.



Gambar 4. 1 Grafik Proses *Pretraining Adam*

Berdasarkan Gambar 4. 2 yang menunjukkan proses *fine-tuning* dengan *learning rate* $1e-5$ menggunakan data lapangan, memperlihatkan grafik *accuracy* pada data pada data *training* mengalami peningkatan dengan sangat cepat pada awal pelatihan dan kemudian cenderung stabil hingga akhir *epoch*. Nilai *training accuracy* tertinggi mencapai 99,12%, sedangkan *validation accuracy* tertinggi mencapai 97,00%, dengan selisih (*gap*) sebesar 2,12%. Meskipun *validation accuracy* meningkat secara bertahap dan terus mendekati *training accuracy*, *gap* tersebut mengindikasikan bahwa kemampuan generalisasi model pada data lapangan masih sedikit lebih rendah dibandingkan pada data *pretraining*. Dari sisi *loss*, nilai *training loss* mendapat penurunan yang besar dari 2,5041 pada awal pelatihan menjadi 0,0206 pada akhir pelatihan. Sementara itu, *validation loss* juga turun secara konsisten dari 3,0223 menjadi 0,1026, dengan selisih (*gap*) sebesar 0,0820 terhadap *training loss*. Penurunan *loss* pada kedua data menunjukkan bahwa

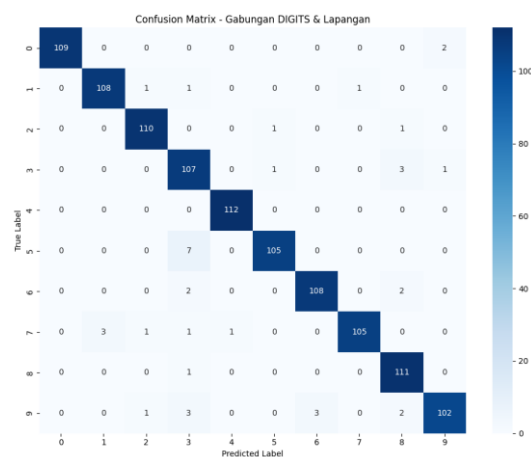
kesalahan model semakin kecil sejalan dengan bertambahnya *epoch*. Namun, dibandingkan dengan tahap *pretraining*, performa pada tahap *fine-tuning* masih lebih rendah, yang ditunjukkan oleh *validation accuracy* senilai 97,00% dan *validation loss* senilai 0,1026, sedangkan pada tahap *pretraining* model mampu mencapai *validation accuracy* 99,61% dengan *validation loss* 0,0108. Perbedaan tersebut mengindikasikan bahwa data lapangan memiliki karakteristik yang lebih kompleks dan bervariasi dibandingkan *dataset* DIGITS, sehingga memberikan tantangan yang lebih besar bagi model dalam melakukan generalisasi.



Gambar 4. 2 Grafik Proses *Fine-tuning Adam*

Performa model pada skenario 1 dapat dikatakan sangat baik, yang terdapat pada Tabel 4. 2, yaitu dengan nilai *accuracy* senilai 96,51%, *precision* 96,63%, *recall* 96,51%, serta *f1-score* 96,52%. Hasil tersebut menandakan bahwa model mampu mengklasifikasikan dengan tingkat ketepatan dan konsistensi yang tinggi. Selain itu, Gambar 4. 3 yang menampilkan *confusion matrix* memperlihatkan bahwa mayoritas prediksi berada pada diagonal utama. Sebagai contoh, kelas digit 0 memiliki 109 prediksi benar, digit 1 sebanyak 108 prediksi benar, digit 2 sebanyak 110 prediksi benar, dan beberapa kelas lainnya juga menunjukkan jumlah prediksi

benar yang dominan. Kesalahan klasifikasi hanya terjadi pada sejumlah kecil data yang tersebar pada beberapa kelas tertentu, seperti digit 5 yang beberapa kali diprediksi sebagai digit 3 serta digit 9 yang sesekali diprediksi sebagai digit 3 atau digit 6. Namun, kuantitas kesalahan yang dihasilkan relatif kecil daripada prediksi benar (*true negative* dan *true positive* tetap dominan), sehingga tidak memberikan dampak yang berarti terhadap performa keseluruhan model.

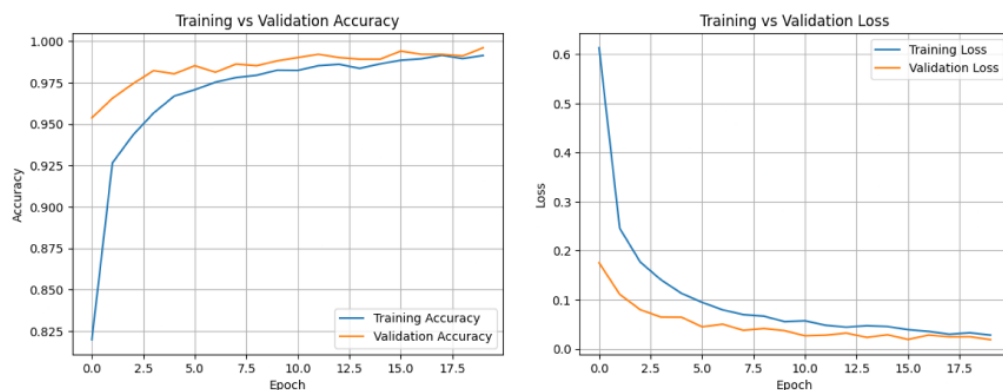


Gambar 4. 3 *Confusion Matrix* Adam

4.1.1.4 Hasil Skenario 2

Pada skenario ini digunakan *optimizer SGD* dengan *learning rate* sebesar $1e-3$ pada fase *pretraining* dan $1e-5$ pada fase *fine-tuning*, serta jumlah *epoch* sebanyak 20. Berdasarkan Gambar 4. 4 yang memperlihatkan hasil *pretraining* dengan *learning rate* $1e-3$, yang menunjukkan bahwa grafik *accuracy* pada data *training* maupun *validation* mendapat kenaikan yang cukup cepat pada awal epoch, kemudian berangsur stabil hingga akhir pelatihan. Nilai *training accuracy* tertinggi menggapai 99,15%, sedangkan *validation accuracy* tertinggi menggapai 99,61%, dengan *gap* sebesar 0,46%. Selisih yang relatif kecil tersebut mengindikasikan

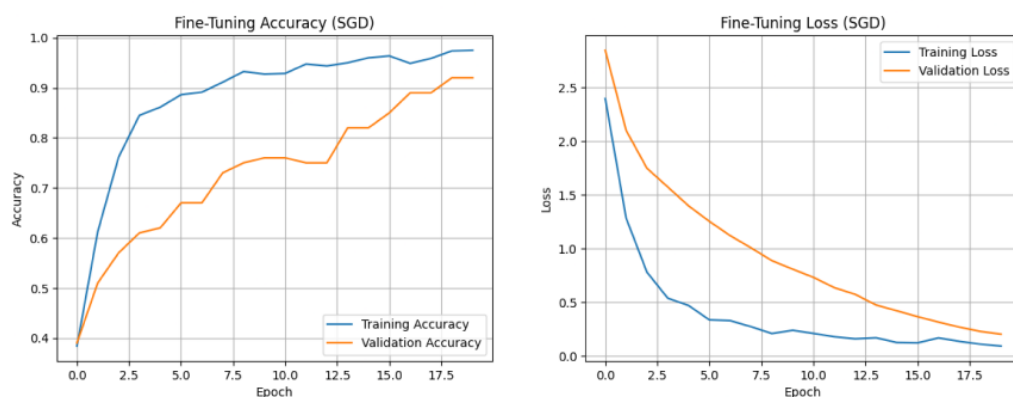
performa model pada data *training* dan *validation* tetap konsisten sehingga model mampu melaksanakan generalisasi dengan baik. Selain itu, grafik *loss* memperlihatkan penurunan yang besar pada awal pelatihan dan terus menurun hingga mencapai *training loss* terendah sebesar 0,0282 dan *validation loss* terendah senilai 0,0187, dengan *gap* senilai 0,0095. Nilai *loss* yang rendah pada kedua data mengindikasikan bahwa kesalahan prediksi model semakin berkurang seiring bertambahnya *epoch*. Secara keseluruhan, pola grafik *accuracy* dan *loss* memperlihatkan bahwa proses *pretraining* menggunakan optimizer SGD berlangsung secara stabil serta menghasilkan model dengan kemampuan generalisasi yang baik tanpa adanya indikasi ketidakstabilan.



Gambar 4. 4 Grafik Proses *Pretraining* SGD

Berdasarkan Gambar 4. 5 yang menunjukkan proses *fine-tuning* dengan *learning rate* $1e-5$ menggunakan data lapangan, terlihat bahwa grafik *accuracy* pada data *training* meningkat dengan cepat pada awal epoch, kemudian cenderung stabil hingga akhir. Nilai *training accuracy* tertinggi mencapai 97,50%, sedangkan *validation accuracy* tertinggi mencapai 92,00%, dengan *gap* sebesar 5,50%. Sementara itu, *accuracy* pada data *validation* juga terjadi peningkatan secara bertahap, namun masih di bawah *training accuracy*. Selisih tersebut

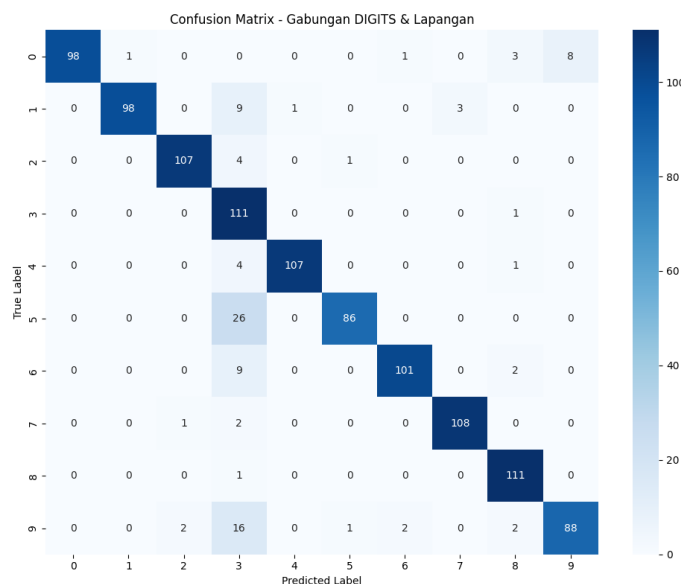
mengindikasikan bahwa kemampuan generalisasi model terhadap data lapangan masih belum sebaik performanya pada data *training*. Dari sisi *loss*, nilai *training loss* terjadi penurunan yang besar dari 2,3969 pada awal pelatihan menjadi senilai 0,0908 pada akhir pelatihan. *Validation loss* juga mengindikasikan tren penurunan yang konsisten dari 2,8470 menjadi 0,2020, dengan *gap* sebesar 0,1112 terhadap *training loss*. Penurunan *loss* pada kedua data mengindikasikan bahwa kesalahan prediksi model semakin sedikit selama proses pelatihan. Namun, dibandingkan dengan tahap *pretraining*, performa pada tahap *fine-tuning* masih lebih rendah. Hal ini ditunjukkan oleh *validation accuracy* senilai 92,00% dan *validation loss* senilai 0,2020, yang mengindikasikan bahwa model masih mengalami rintangan dalam melakukan generalisasi pada data lapangan yang memiliki karakteristik lebih bervariasi dibandingkan *dataset* DIGITS.



Gambar 4. 5 Grafik Proses *Fine-tuning* SGD

Performa model pada skenario 2 dapat dikatakan cukup baik, yang terdapat dilihat pada Tabel 4. 2 dengan nilai *accuracy* 90,95%, *precision* 93,25%, *recall* 90,95%, dan *f1-score* 91,38%. Meskipun demikian, nilai tersebut masih lebih rendah dibandingkan dengan skenario 1 yang menunjukkan performa lebih optimal.

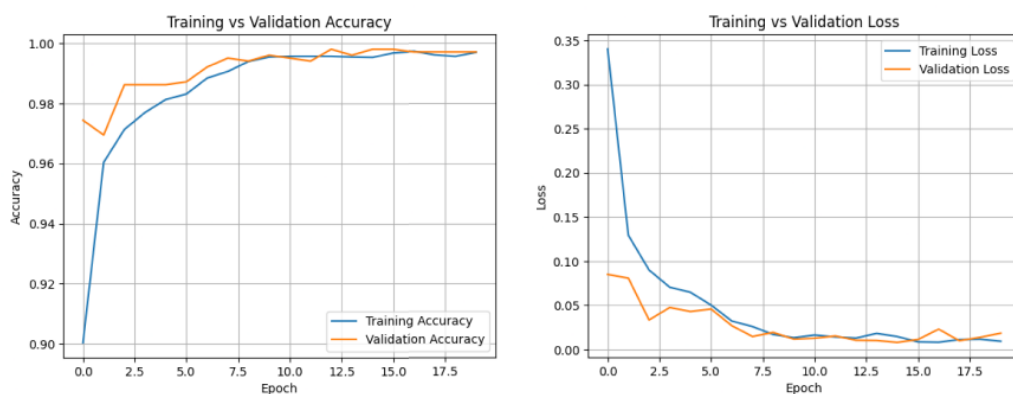
Hal ini juga tercermin pada Gambar 4. 6 yang menampilkan *confusion matrix*, di mana meskipun sebagian besar prediksi masih berada pada posisi diagonal utama yang menunjukkan klasifikasi benar, terlihat adanya peningkatan kuantitas kesalahan klasifikasi pada beberapa kelas tertentu. Sebagai contoh, kelas digit 5 hanya memiliki 86 prediksi benar dan sebanyak 26 data salah diprediksi sebagai digit 3. Selain itu, digit 9 juga mengalami cukup banyak kesalahan klasifikasi, yaitu sebanyak 16 data diprediksi sebagai digit 3. Kesalahan klasifikasi juga terlihat pada digit 1 dan digit 6 yang beberapa kali diprediksi sebagai digit 3. Kondisi ini mengindikasikan bahwa model mengalami kesulitan membedakan pola antar kelas tertentu. Meskipun model pada skenario 2 masih mampu melakukan klasifikasi dengan cukup baik, performanya cenderung menurun dibandingkan skenario 1, terutama dalam hal konsistensi prediksi antar kelas.



Gambar 4. 6 *Confusion Matrix* SGD

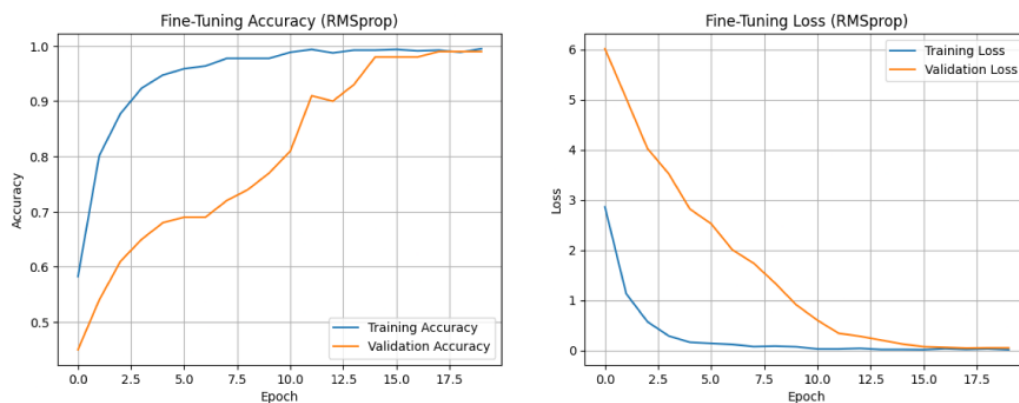
4.1.1.5 Hasil Skenario 3

Pada skenario ini digunakan *optimizer RMSProp* dengan *learning rate* sebesar $1e-3$ pada fase *pretraining* dan $1e-5$ pada fase *fine-tuning*, serta jumlah *epoch* sebanyak 20. Berdasarkan Gambar 4. 7 yang menunjukkan hasil proses pretraining dengan *learning rate* $1e-3$, terlihat bahwa grafik akurasi pada data *training* meningkat dengan cepat pada awal pelatihan dan kemudian cenderung stabil hingga akhir. Pola yang serupa juga terlihat pada data *validation*, di mana akurasi terus meningkat dan mengikuti tren yang sejalan dengan akurasi *training*. Nilai *training accuracy* tertinggi yang diperoleh mencapai 99,73%, sedangkan *validation accuracy* tertinggi mencapai 99,80%, dengan *gap* sebesar 0,07%. *Gap* yang sangat kecil mengindikasikan bahwa model mampu melakukan generalisasi dengan baik tanpa menunjukkan gejala *overfitting* yang berarti. Dari sisi *loss*, nilai *training loss* mendapat penurunan yang tajam pada awal pelatihan dan terus menurun hingga mencapai nilai terendah senilai 0,0082, sedangkan *validation loss* terendah mencapai 0,0080, dengan *gap* senilai 0,0002. Nilai *loss* yang sangat rendah dan mendekati nilai optimal, yaitu 0, menunjukkan bahwa kesalahan prediksi model semakin kecil.



Gambar 4. 7 Grafik Proses *Pretraining RMSProp*

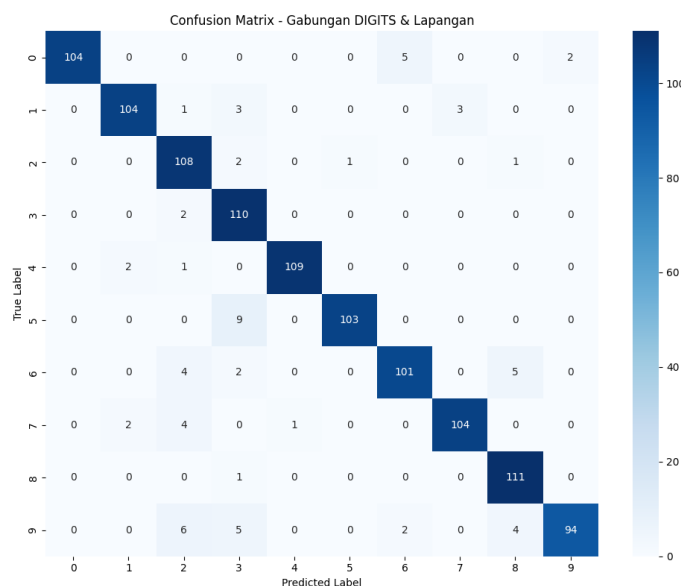
Berdasarkan Gambar 4. 8 yang menunjukkan proses *fine-tuning* dengan *learning-rate* $1e-5$ menggunakan data lapangan, memperlihatkan bahwa nilai *accuracy* pada data *training* meningkat signifikan selama proses *training* dan kemudian cenderung stabil pada nilai yang tinggi. Sementara itu, *accuracy* pada data *validation* juga menunjukkan peningkatan yang konsisten hingga mendekati nilai *training accuracy* pada akhir pelatihan. Nilai *training accuracy* paling tinggi diperoleh mencapai 99,50%, sedangkan *validation accuracy* tertinggi mencapai 99,00%, dengan *gap* sebesar 0,50%. *Gap* yang relatif kecil mengindikasikan bahwa model mampu melakukan generalisasi dengan baik tanpa menunjukkan gejala *overfitting* yang berarti. Dari sisi *loss*, nilai *training loss* terus berkurang hingga mencapai nilai terendah sebesar 0,0145, sedangkan *validation loss* terendah mencapai 0,0479, dengan *gap* senilai 0,0334.



Gambar 4. 8 Grafik Proses *Fine-tuning RMSProp*

Performa model pada skenario 3 dapat dikatakan baik yang terdapat pada Tabel 4. 2, yaitu nilai *accuracy* 93,91%, *precision* 94,37%, *recall* 93,91%, dan *f1-score* 93,95%. Jika dibandingkan, hasil ini lebih baik daripada skenario 2, namun masih berada di bawah performa optimal pada skenario 1. Hal ini juga tercermin

pada *confusion matrix* pada Gambar 4. 9, di mana mayoritas prediksi terkonsentrasi pada diagonal utama yang menandakan klasifikasi benar untuk hampir seluruh kelas digit. Sebagai contoh, digit 3 memiliki 110 prediksi benar, digit 4 sebanyak 109 prediksi benar, dan digit 8 sebanyak 111 prediksi benar. Namun, masih terjadi beberapa kesalahan klasifikasi yang tersebar pada kelas tertentu. Kesalahan yang cukup terlihat terjadi pada digit 5 yang beberapa kali diprediksi sebagai digit 3 sebanyak 9 data. Selain itu, digit 9 juga beberapa kali diprediksi sebagai digit 2, digit 3, dan digit 8. Beberapa kesalahan kecil lainnya juga terlihat pada digit 6 yang diprediksi sebagai digit 2 dan digit 8, serta digit 7 yang sesekali diprediksi sebagai digit 2. Distribusi kesalahan ini menunjukkan bahwa model sudah mampu mengenali pola dengan cukup baik dan memiliki kecakapan generalisasi yang lebih baik dibandingkan skenario 2, meskipun masih sedikit di bawah skenario 1 dalam hal ketepatan dan kestabilan prediksi antar kelas.



Gambar 4. 9 *Confuion Matrix* RMSProp

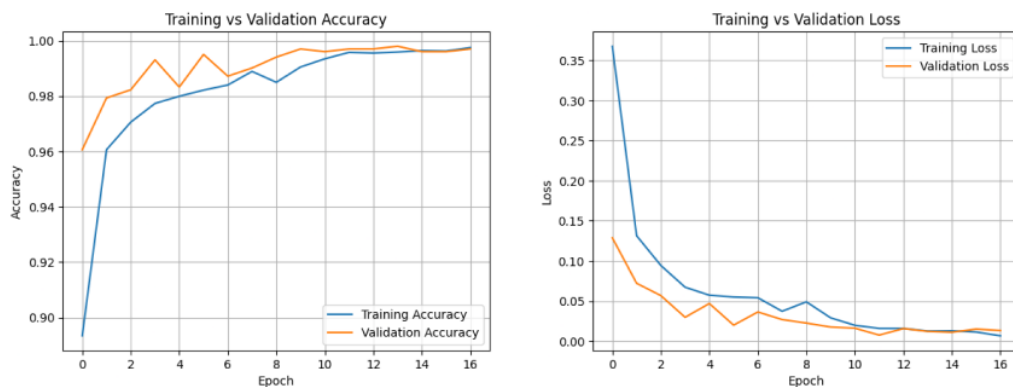
4.1.1.6 Hasil Skenario 4

Pada Skenario 4, digunakan *optimizer* terbaik yang diperoleh dari hasil perbandingan pada skenario sebelumnya, yaitu *Adam*, dengan konfigurasi *learning rate* sebesar $1e-3$ pada fase *pretraining* dan $1e-5$ pada fase *fine-tuning*, serta jumlah *epoch* sebanyak 20. Namun, konfigurasi ini pada dasarnya identik dengan yang telah diterapkan pada skenario 1, sehingga proses pelatihan, hasil evaluasi, serta pola yang ditunjukkan pada grafik *accuracy*, *loss*, maupun *confusion matrix* akan menghasilkan nilai dan visualisasi yang sama. Dengan kata lain, tidak ada perbedaan (sama) antara performa skenario 4 dan skenario 1, di mana model tetap memperoleh hasil yang optimal dengan tingkat *accuracy*, *precision*, *recall*, dan *f1-score* tertinggi daripada skenario lainnya. Hal ini menegaskan kembali bahwa kombinasi *optimizer Adam* dengan *learning rate* $1e-5$ pada fase *fine-tuning* merupakan konfigurasi yang paling efektif dalam penelitian ini.

4.1.1.7 Hasil Skenario 5

Pada Skenario 5, dilakukan proses *pretraining* menggunakan *optimizer* terbaik, yaitu *Adam* dengan *learning rate* sebesar $1e-3$ pada fase *pretraining* dan $1e-4$ pada fase *fine-tuning*, serta jumlah *epoch* sebanyak 20. Berdasarkan Gambar 4. 10, memperlihatkan bahwa nilai *accuracy* pada data *training* bertambah dengan cepat pada awal pelatihan dan kemudian cenderung stabil. Sementara itu, *accuracy* pada data *validation* juga menandakan peningkatan yang konsisten dan mengikuti pola *training accuracy*, meskipun terjadi sedikit fluktuasi pada beberapa tahap pelatihan. Nilai *training accuracy* tertinggi yang diperoleh mencapai 99,75%, sedangkan *validation accuracy* tertinggi mencapai 99,80%, dengan *gap* sebesar

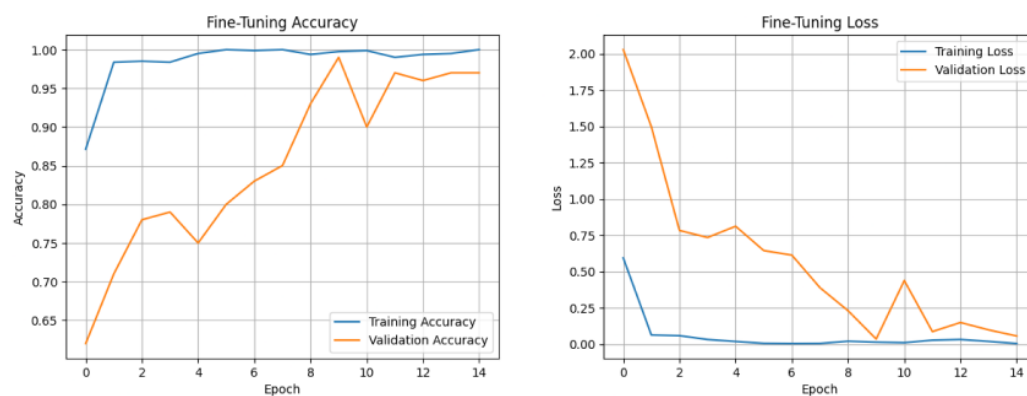
0,05%. Dari sisi *loss*, nilai *training loss* terus berkurang hingga mencapai nilai terendah sebesar 0,0066, sedangkan *validation loss* terendah mencapai 0,0075, dengan *gap* senilai 0,0009. Nilai *loss* yang sangat rendah dan mendekati nilai optimal, yaitu 0, menunjukkan bahwa kesalahan prediksi model berkurang seiring berlangsungnya proses pembelajaran.



Gambar 4. 10 Grafik Proses *Pretraining* Skenario 5

Berdasarkan Gambar 4. 11 yang menunjukkan proses *fine-tuning* pada Skenario 5 dengan optimizer *Adam* dan *learning rate* $1e-4$, memperlihatkan bahwa nilai *accuracy* pada data *training* bertambah dengan sangat cepat dan cenderung stabil pada nilai yang sangat tinggi selama proses pelatihan. Sementara itu, *accuracy* pada data *validation* juga terjadi peningkatan yang cukup signifikan, meskipun masih menunjukkan fluktuasi pada beberapa tahap pelatihan dan tidak selalu mengikuti pola *training accuracy*. Nilai *training accuracy* tertinggi yang diperoleh mencapai 100,00%, sedangkan *validation accuracy* tertinggi mencapai 99,00%, dengan *gap* sebesar 1,00%. Dari sisi *loss*, nilai *training loss* terus menurun hingga mencapai nilai terendah sebesar 0,0038, sedangkan *validation loss* terendah mencapai 0,0347, dengan *gap* senilai 0,0309. Namun, *gap* yang relatif lebih besar dibandingkan skenario sebelumnya, disertai dengan fluktuasi pada *validation*

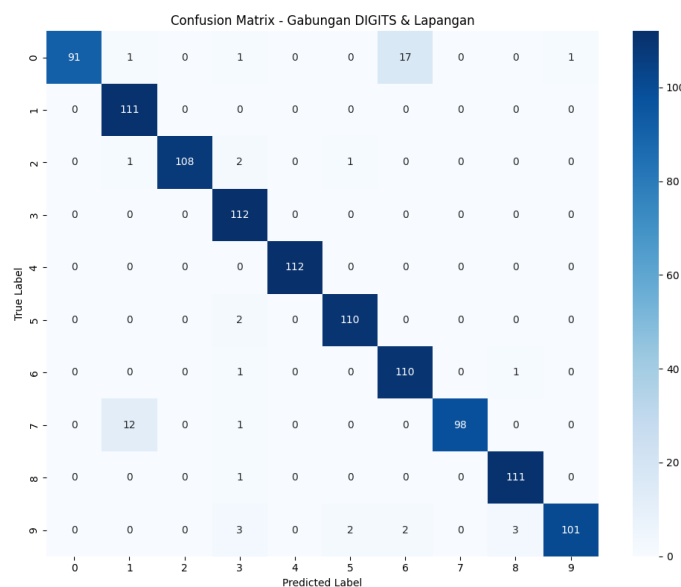
accuracy dan *validation loss*, mengindikasikan adanya kecenderungan *overfitting* ringan, di mana model menandakan performa yang sangat baik pada data *training* tetapi belum sepenuhnya konsisten pada data *validation*. Kondisi ini dipengaruhi oleh *learning rate* yang digunakan relatif besar sehingga model belajar dengan cepat terhadap data *training*. Secara keseluruhan, proses *fine-tuning* menghasilkan performa yang baik, tetapi masih menunjukkan indikasi *overfitting* ringan yang terlihat dari perbedaan performa antara data *training* dan *validation*.



Gambar 4. 11 Grafik Proses *Fine-tuning* Skenario 5

Performa model pada skenario 5 tergolong baik yang terdapat pada Tabel 4. 4, dengan nilai *accuracy* 95,34%, *precision* 95,80%, *recall* 95,34%, dan *f1-score* 95,32%. Hasil ini menunjukkan bahwa performanya hampir mendekati skenario 1 dan 4 yang merupakan konfigurasi terbaik, serta lebih unggul dibandingkan skenario 2 dan 3 yang menggunakan *optimizer SGD* dan *RMSProp*. Hal ini juga tercermin pada *confusion matrix* pada Gambar 4. 12, di mana mayoritas berada pada diagonal utama yang menandakan klasifikasi benar untuk hampir seluruh kelas digit. Sebagai contoh, digit 1, digit 3, dan digit 4 masing-masing memiliki lebih dari 110 prediksi benar, sedangkan digit 5 dan digit 6 masing-masing memiliki 110

prediksi benar. Sedangkan, kesalahan yang paling terlihat terjadi pada digit 0 yang beberapa kali diprediksi sebagai digit 6 sebanyak 17 data. Selain itu, digit 7 juga beberapa kali diprediksi sebagai digit 1 sebanyak 12 data. Beberapa kesalahan kecil lainnya terlihat pada digit 9 yang sesekali diprediksi sebagai digit 3, digit 5, digit 6, dan digit 8. Hal ini menunjukkan bahwa model pada skenario 5 sudah mampu melakukan klasifikasi dengan baik dan memiliki kestabilan prediksi yang cukup tinggi pada sebagian besar kelas digit.

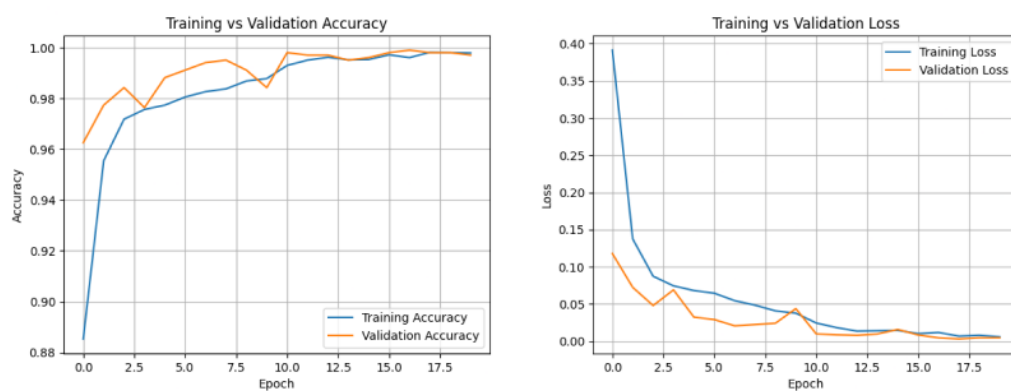


Gambar 4. 12 *Confusion Matrix* Skenario 5

4.1.1.8 Hasil Skenario 6

Pada Skenario 6, dilakukan proses *pretraining* menggunakan *optimizer Adam* dengan *learning rate* sebesar $1e-3$ pada fase *pretraining* dan $1e-6$ pada fase *fine-tuning*, serta jumlah *epoch* sebanyak 20. Berdasarkan Gambar 4. 13, terlihat bahwa nilai *accuracy* pada data *training* meningkat secara konsisten selama proses pelatihan dan mencapai nilai yang sangat tinggi pada akhir pelatihan. Sementara itu, *accuracy* pada data *validation* juga memperlihatkan tren peningkatan yang baik,

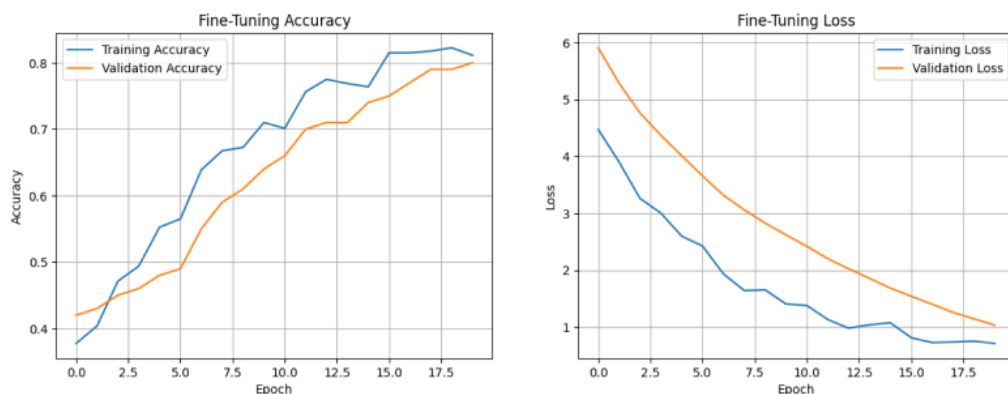
bahkan pada beberapa tahap memiliki nilai yang lebih tinggi dibandingkan *training accuracy*, sebelum akhirnya stabil pada akhir proses pelatihan. Nilai *training accuracy* tertinggi yang diperoleh mencapai 99,82%, sedangkan *validation accuracy* tertinggi mencapai 99,90%, dengan *gap* sebesar 0,08%. Dari sisi *loss*, nilai *training loss* terus berkurang hingga mencapai nilai terendah sebesar 0,0058, sedangkan *validation loss* terendah mencapai 0,0030, dengan *gap* senilai 0,0028.



Gambar 4. 13 Grafik Proses *Pretraining* Skenario 6

Berdasarkan Gambar 4. 14 yang menunjukkan proses *fine-tuning* pada Skenario 6 dengan optimizer *Adam* dan *learning rate* sebesar $1e-6$, memperlihatkan bahwa nilai *accuracy* pada data *training* meningkat secara bertahap selama proses pelatihan, namun laju peningkatannya relatif lambat dibandingkan skenario sebelumnya. *Accuracy* pada data *validation* juga menandakan tren peningkatan yang stabil dan mengikuti pola *training accuracy*, meskipun nilainya cenderung berada di bawah *training accuracy* sepanjang pelatihan. Nilai *training accuracy* tertinggi yang diperoleh mencapai 82,25%, sedangkan *validation accuracy* tertinggi mencapai 80,00%, dengan *gap* sebesar 2,25%. Meskipun *gap* tersebut masih tergolong kecil dan tidak menunjukkan indikasi *overfitting* yang signifikan, nilai akurasi yang diperoleh masih lebih rendah dibandingkan skenario lainnya.

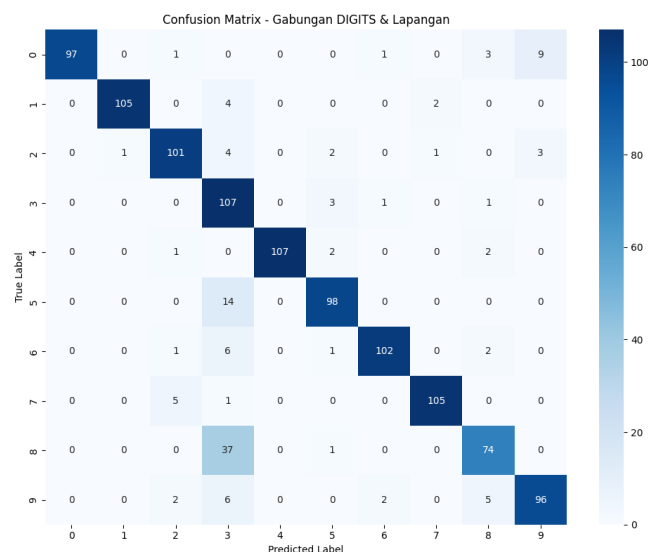
Dari sisi *loss*, nilai *training loss* terus menurun hingga mencapai nilai terendah sebesar 0,7153, sedangkan *validation loss* terendah mencapai 1,0382, dengan *gap* sebesar 0,3229. Nilai *loss* yang masih cukup tinggi dan belum mendekati nilai optimal, yaitu 0, menunjukkan bahwa model belum mencapai konvergensi secara maksimal. Selain itu, penurunan *loss* yang berlangsung secara perlahan mengindikasikan bahwa proses optimasi berjalan lambat sehingga model belum mampu mempelajari pola data secara optimal dalam jumlah *epoch* yang diberikan. Secara keseluruhan, penggunaan *learning rate* $1e-6$ kurang efektif karena memperlambat proses optimasi sehingga performa model yang dihasilkan belum optimal.



Gambar 4. 14 Grafik Proses *Fine-tuning* Skenario 6

Performa model pada Skenario 6 merupakan yang paling rendah dibandingkan seluruh skenario yang diuji, yang terdapat dilihat pada Tabel 4. 4 dengan nilai *accuracy* sebesar 88,89%, *precision* 90,86%, *recall* 88,89%, dan *f1-score* 89,27%. Nilai ini berada di bawah skenario 2 dan 3, serta terpaut cukup jauh dari skenario 1 dan 4 yang menunjukkan performa terbaik. Hal ini mengindikasikan bahwa konfigurasi pada skenario 6 kurang optimal dalam meningkatkan

kemampuan model. Kondisi tersebut juga tercermin pada *confusion matrix* pada Gambar 4. 15, di mana meskipun sebagian besar prediksi masih berada pada diagonal utama, jumlah kesalahan klasifikasi terlihat lebih banyak dan lebih tersebar dibandingkan skenario lainnya. Kesalahan klasifikasi yang paling signifikan terlihat pada digit 8, di mana sebanyak 37 data salah diprediksi sebagai digit 3, sehingga jumlah prediksi benar pada digit 8 hanya mencapai 74 data. Selain itu, digit 5 juga beberapa kali diprediksi sebagai digit 3 sebanyak 14 data, sedangkan digit 6 beberapa kali diprediksi sebagai digit 3 sebanyak 6 data. Kesalahan lain juga terlihat pada digit 0 yang beberapa kali diprediksi sebagai digit 9 sebanyak 9 data. Banyaknya kesalahan klasifikasi tersebut menunjukkan bahwa model belum mampu membedakan pola antar kelas dengan baik, sehingga kuantitas *false positive* dan *false negative* menjadi lebih tinggi dibandingkan skenario lainnya. Hal tersebut sejalan dengan penggunaan *learning rate* yang sangat kecil ($1e-6$), sehingga proses pembelajaran berlangsung terlalu lambat dan model belum mencapai konvergensi maksimal, yang pada akhirnya menyebabkan performa menjadi yang paling rendah di antara seluruh skenario yang diuji.

Gambar 4. 15 *Confusion Matrix* Skenario 6

Berdasarkan hasil *confusion matrix* dari seluruh skenario yang telah diuji, terlihat adanya pola kesalahan klasifikasi yang berulang. Kesalahan prediksi yang paling dominan terjadi pada pasangan digit 5 dan 3, digit 9 dan 3. Selain itu, kesalahan klasifikasi pada pasangan digit 8 dan 3, digit 0 dan 6, digit 7 dan 1 juga ditemukan pada hasil pengujian, meskipun frekuensinya berbeda pada setiap skenario. Pola tersebut terlihat secara konsisten pada hasil *confusion matrix* masing-masing skenario pengujian.

Selain faktor kemiripan visual, jumlah dan sebaran kesalahan pada *confusion matrix* juga dipengaruhi secara signifikan oleh konfigurasi *optimizer* dan *learning rate* yang digunakan pada setiap skenario. Skenario 1 dan 4 yang menggunakan *optimizer* Adam dengan *learning rate* $1e-5$ menghasilkan *confusion matrix* dengan kesalahan paling minimal, karena mekanisme *adaptive learning rate* pada Adam memungkinkan pembaruan bobot yang lebih presisi pada setiap parameter sehingga model mampu membedakan pola antar kelas dengan lebih baik. Sebaliknya, Skenario 2 yang menggunakan SGD menunjukkan jumlah kesalahan

yang relatif besar dan terkonsentrasi, dengan 26 data digit 5 salah diprediksi sebagai digit 3, karena SGD tidak memiliki mekanisme adaptif sehingga kurang efektif dalam mengoptimalkan bobot untuk membedakan pasangan digit yang memiliki kemiripan visual. Skenario 3 yang menggunakan RMSProp menunjukkan performa yang lebih baik dibandingkan Skenario 2, namun masih terdapat beberapa kesalahan klasifikasi, di mana digit 5 diprediksi sebagai digit 3 sebanyak 9 data, yang mengindikasikan bahwa meskipun RMSProp memiliki mekanisme adaptif, kemampuannya dalam membedakan pasangan digit dengan kemiripan visual belum seoptimal Adam. Adapun Skenario 5 yang menggunakan *optimizer* terbaik, yaitu Adam dengan *learning rate* yang lebih besar dibandingkan Skenario 1 dan 4, yaitu $1e-4$, menunjukkan pola kesalahan yang sedikit berbeda, di mana kesalahan terbesar terjadi pada digit 0 yang diprediksi sebagai digit 6 sebanyak 17 data dan digit 7 yang diprediksi sebagai digit 1 sebanyak 12 data. Hal ini mengindikasikan bahwa penggunaan *learning rate* yang lebih besar menghasilkan performa yang tidak seoptimal konfigurasi dengan *learning rate* $1e-5$. Sementara itu, Skenario 6 dengan *learning rate* $1e-6$ memperlihatkan pola kesalahan yang paling tersebar dengan 37 data digit 8 salah diprediksi sebagai digit 3, yang menandakan bahwa *learning rate* yang terlalu kecil menyebabkan model gagal beradaptasi secara optimal selama proses *fine-tuning*.

4.1.2 Hasil Pengujian *Blackbox Testing*

Pengujian fungsionalitas sistem aplikasi *mobile* OCR dilakukan menggunakan metode *blackbox testing*. Berdasarkan hasil pengujian fungsionalitas aplikasi *mobile* menggunakan metode *blackbox testing*, seluruh skenario yang telah

dibuat menunjukkan hasil yang cocok dengan yang diharapkan. Setiap fitur utama aplikasi, mulai dari proses autentikasi pengguna, pengambilan dan pemrosesan gambar melalui kamera, verifikasi hasil OCR, hingga penyimpanan dan pengiriman data, dapat berjalan dengan baik tanpa ditemukan kesalahan fungsional. Selain itu, fitur pendukung seperti histori, pengaturan tema, profil pengguna, dan *logout* juga berfungsi secara optimal. Rincian hasil pengujian dengan menggunakan *blackbox testing* ditunjukkan pada Tabel 4. 5.

Tabel 4. 5 Hasil Pengujian Aplikasi *Mobile OCR* dengan *Blackbox Testing*

ID	Use Case	Skenario Pengujian	Hasil yang Diharapkan	Keterangan
TC01	UC01	Pengguna login menggunakan <i>username</i> dan <i>password</i> yang benar.	Sistem sukses dalam melakukan autentikasi dan mengarahkan ke halaman beranda	<i>Valid</i>
TC02	UC01	Pengguna login menggunakan <i>username</i> dan <i>password</i> yang salah.	Sistem memunculkan notifikasi kesalahan dan tetap berada di halaman <i>login</i>	<i>Valid</i>
TC03	UC01	Pengguna <i>login</i> dengan tidak mengisi <i>username</i> dan <i>password</i> .	Sistem memunculkan notifikasi kesalahan dan tetap berada di halaman <i>login</i>	<i>Valid</i>
TC04	UC02	Pengguna login tanpa mengisi <i>username</i> dan <i>password</i> .	Kamera terbuka dan siap digunakan untuk mengambil foto digit angka jumlah pemakaian pada meteran air.	<i>Valid</i>
TC05	UC02	Pengguna mengambil gambar digit pada meteran air	Gambar tersimpan sementara dan hasil pemindaian OCR ditampilkan pada halaman <i>input</i> baca meter	<i>Valid</i>
TC06	UC03	Pengguna membandingkan hasil OCR dengan digit pada meteran	Pengguna dapat menyetujui hasil atau melakukan koreksi manual	<i>Valid</i>
TC07	UC04	Hasil pembacaan disimpan	Data tersimpan ke <i>database</i> dan muncul pada halaman histori	<i>Valid</i>
TC08	UC04	Data <i>input</i> baca meter dikirim ke sistem web eksternal	Sistem berhasil mengirimkan data ke sistem web eksternal	<i>Valid</i>
TC09	UC05	Pengguna membuka menu histori	Daftar pembacaan yang telah disimpan tampil.	<i>Valid</i>
TC10	UC05	Pengguna menekan tombol <i>clear</i>	Catatan histori terhapus dan daftar menjadi kosong.	<i>Valid</i>

ID	Use Case	Skenario Pengujian	Hasil yang Diharapkan	Keterangan
TC11	UC05	Pengguna menekan tombol detail pada daftar data histori	Detail histori tampil dengan menampilkan informasi yang lebih rinci.	<i>Valid</i>
TC12	UC06	Pengguna memilih tema gelap/ terang di menu pengaturan.	Tema aplikasi berubah sesuai pilihan	<i>Valid</i>
TC13	UC07	Pengguna membuka menu profil.	Informasi pengguna tampil sesuai data yang terdaftar.	<i>Valid</i>
TC14	UC08	Pengguna membuka menu daftar pelanggan	Daftar data pelanggan PAMDes Pandanrejo tampil sesuai data yang terdaftar	<i>Valid</i>
TC15	UC08	Pengguna menggunakan filter pencarian untuk mencari pelanggan.	Daftar data pelanggan yang muncul hanya berdasarkan dengan yang dicari	<i>Valid</i>
TC16	UC08	Pengguna menggunakan filter pilihan desa untuk mencari pelanggan sesuai desa tertentu	Daftar data pelanggan yang muncul hanya berdasarkan dengan desa yang sesuai	<i>Valid</i>
TC17	UC08	Pengguna menggunakan filter pilihan rt untuk mencari pelanggan sesuai rt tertentu	Daftar data pelanggan yang muncul hanya berdasarkan dengan rt yang sesuai	<i>Valid</i>
TC18	UC08	Pengguna memilih pelanggan dengan menekan tombol pilih pada daftar pelanggan	Pengguna kembali halaman <i>input</i> baca meter dengan membawa informasi pelanggan	<i>Valid</i>
TC19	UC09	Pengguna menekan tombol keluar.	Sesi diakhiri oleh sistem dan mengembalikan ke halaman <i>login</i>	<i>Valid</i>

Berdasarkan hasil pengujian yang telah dilakukan, seluruh skenario uji pada aplikasi menunjukkan status berhasil (*valid*) tanpa ditemukan kegagalan pada setiap kasus pengujian. Hal ini menunjukkan bahwa seluruh fungsi yang diuji telah berjalan sesuai dengan yang diharapkan. Adapun tingkat keberhasilan pengujian dapat dihitung menggunakan persamaan 2.18. Berdasarkan hasil pengujian, jumlah kasus uji yang berhasil adalah sebanyak 19 dari total 19 kasus pengujian, sehingga diperoleh perhitungan sebagai berikut:

$$\text{Persentase Keberhasilan} = \frac{19}{19} \times 100\%$$

$$\text{Persentase Keberhasilan} = 100\%$$

Hal ini mengindikasikan bahwa aplikasi yang dikembangkan telah memenuhi aspek fungsionalitas yang diharapkan dan siap digunakan dalam kondisi operasional.

4.1.3 Hasil Evaluasi Akurasi OCR dalam Pengenalan Digit

Evaluasi terhadap kinerja sistem OCR dilakukan menggunakan akurasi pengenalan digit untuk mengukur tingkat keberhasilan model dalam mengenali digit pada meteran air. Pengujian melalui perbandingan hasil prediksi OCR dengan nilai asli yang terdapat pada meteran air untuk setiap sampel data. Perbandingan tersebut bertujuan untuk mengetahui sejauh mana sistem mampu mengenali digit angka secara tepat serta mengidentifikasi kesalahan yang masih muncul pada proses pembacaan digit. Hasil perbandingan tersebut secara lebih rinci dapat dilihat pada Tabel 4. 6.

Tabel 4. 6 Perbandingan Hasil Prediksi Sistem OCR dengan Nilai Asli pada Meteran Air

No.	Nilai Asli pada Meteran Air	Hasil Prediksi OCR
1	00030	70030
2	00174	00174
3	00252	00254
4	00256	00251
5	00276	00278
6	00295	00295
7	00349	00349
8	00394	00394
9	00396	00396
10	00479	00479
11	00523	00529
12	00575	00575
13	00613	00613
14	00798	00198
15	00809	00909

No.	Nilai Asli pada Meteran Air	Hasil Prediksi OCR
16	00891	00891
17	00947	00949
18	01082	01082
19	01252	01250
20	01294	64494
21	01308	01306
22	01342	01346
23	01430	01439
24	01458	01458
25	01468	01464
26	01861	01861
27	01913	04911
28	01937	01937
29	02242	02242
30	05735	05734
31	05914	05914

Berdasarkan hasil pengujian pada Tabel 4. 6, dilakukan identifikasi pada jumlah digit yang berhasil dikenali dengan benar dan digit yang mengalami kesalahan. Pengujian dilakukan terhadap 31 meteran air, di mana setiap meteran memiliki 5 digit angka, sehingga total digit yang diuji adalah 155 digit. Dari total tersebut, sebanyak 136 digit berhasil dikenali dengan benar. Sehingga, perhitungan akurasi pengenalan digit dilakukan sebagai berikut:

$$Akurasi\ OCR = \frac{136}{155} \times 100\%$$

$$Akurasi\ OCR = 87,74\%$$

Berdasarkan hasil perhitungan tersebut, diperoleh nilai akurasi OCR sebesar 87,74% dalam mengenali digit pada meteran air.

4.2 Pembahasan

Pada bagian ini akan dibahas mengenai hasil yang telah diperoleh pada tahap sebelumnya, baik dari sisi pengembangan model maupun implementasi sistem aplikasi. Pembahasan difokuskan pada analisis performa model berdasarkan

variasi skenario pelatihan yang telah dilakukan, seperti pengaruh pemilihan *optimizer* dan nilai *learning rate* terhadap hasil yang diperoleh. Selanjutnya, dilakukan pembahasan terhadap hasil implementasi sistem serta evaluasi kinerja aplikasi yang meliputi aspek fungsionalitas dan akurasi sistem OCR.

Berdasarkan hasil uji coba yang sudah dilakukan pada berbagai skenario pelatihan, diperoleh perbedaan performa model yang cukup signifikan pada setiap konfigurasi yang digunakan. Variasi tersebut terlihat dari nilai *accuracy*, *precision*, *recall*, dan *f1-score* yang dihasilkan, di mana beberapa skenario mampu mendapatkan hasil yang lebih optimal daripada yang lainnya. Perbedaan performa ini menunjukkan bahwa pemilihan parameter pelatihan, seperti jenis *optimizer* dan nilai *learning rate*, memiliki pengaruh signifikan pada kecakapan model dalam mengenali pola digit pada citra meteran air.

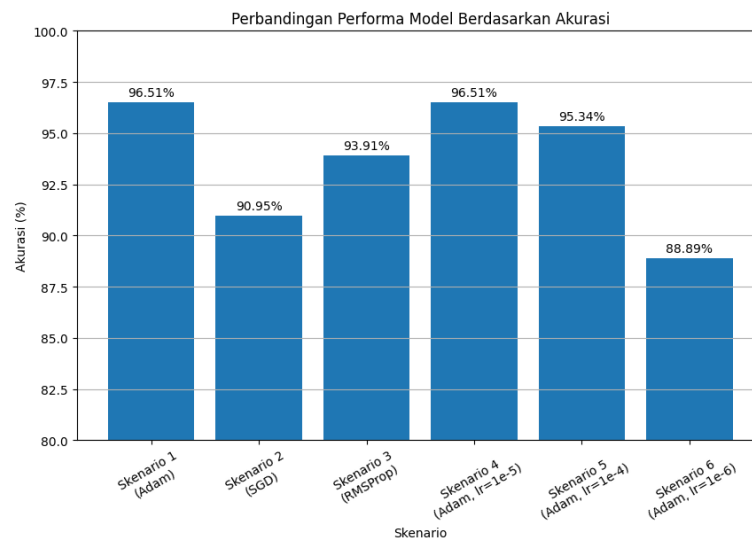
Pemilihan *optimizer* pada proses pelatihan model memberikan pengaruh yang cukup signifikan terhadap performa yang dihasilkan. Berdasarkan hasil pengujian, *optimizer Adam* menghasilkan performa terbaik daripada *SGD* dan *RMSProp*, yang ditunjukkan oleh nilai akurasi dan metrik evaluasi lainnya yang lebih tinggi. Hal ini disebabkan karena Adam memiliki mekanisme adaptif dalam mengatur *learning rate* untuk setiap parameter, sehingga proses pembelajaran menjadi lebih stabil dan konvergen lebih cepat. Sementara itu, *SGD* cenderung menunjukkan proses pembelajaran yang lebih lambat, sedangkan *RMSProp* meskipun bersifat adaptif, pada pengujian ini belum mampu mendapatkan performa yang lebih baik dibandingkan dengan *Adam*.

Pengaruh nilai *learning rate* pada fase *fine-tuning* juga memberikan dampak yang signifikan terhadap performa model yang dihasilkan. Berdasarkan hasil pengujian, penggunaan *learning rate* sebesar $1e-5$ yang dikombinasikan dengan *optimizer Adam* mampu menghasilkan performa terbaik dibandingkan variasi lainnya. Selain itu, *learning rate* sebesar $1e-4$ juga menandakan performa yang hampir mendekati hasil tersebut. Sebaliknya, penggunaan *learning rate* yang terlalu kecil, yaitu $1e-6$, cenderung menghasilkan performa yang lebih rendah. Hal ini disebabkan karena nilai *learning rate* yang terlalu kecil mengakibatkan proses pembelajaran berjalan sangat lambat, sehingga model tidak mampu menyesuaikan bobot secara optimal selama proses *fine-tuning*. Oleh karena itu, pemilihan nilai *learning rate* yang tepat pada tahap *fine-tuning* menjadi faktor krusial dalam memperoleh performa model yang optimal.

Analisis terhadap confusion matrix pada setiap skenario menunjukkan adanya pola kesalahan klasifikasi yang berulang pada pasangan digit tertentu, yaitu digit 5 dan 3, digit 9 dan 3, digit 8 dan 3, digit 0 dan 6, serta digit 7 dan 1. Pola tersebut muncul secara konsisten pada beberapa skenario pengujian. Selain itu, perbedaan konfigurasi optimizer dan learning rate menghasilkan tingkat kesalahan klasifikasi yang berbeda. Berdasarkan hasil pengujian, kombinasi optimizer Adam dengan *learning rate* $1e-5$ menghasilkan jumlah kesalahan klasifikasi yang paling rendah, sedangkan penggunaan *optimizer SGD* dan *learning rate* $1e-6$ menghasilkan jumlah kesalahan yang lebih tinggi.

Secara keseluruhan, perbandingan performa dari seluruh skenario yang telah diuji dapat dilihat pada Gambar 4. 16, di mana Skenario 1 dan Skenario 4

menunjukkan performa tertinggi, sedangkan Skenario 6 memperoleh performa terendah di antara seluruh skenario yang diuji.



Gambar 4. 16 Perbandingan Performa Seluruh Skenario

Hasil pengujian fungsionalitas dengan metode *blackbox testing* memperlihatkan bahwa semua fitur pada aplikasi dapat berjalan sesuai dengan yang diharapkan. Setiap skenario pengujian yang telah dirancang berhasil dieksekusi tanpa ditemukan kesalahan. Seluruh kasus uji memperoleh status *valid*, yang menandakan bahwa sistem mampu merespons setiap input pengguna dengan benar sesuai dengan fungsinya. Sehingga aplikasi yang dikembangkan sudah memenuhi aspek fungsionalitas dan dapat diterapkan dengan baik dalam mendukung proses pembacaan meteran air.

Hasil evaluasi akurasi sistem OCR mengindikasikan bahwa model mampu mengenali sebagian besar digit dengan cukup baik ketika diimplementasikan pada aplikasi *mobile*. Berdasarkan pengujian akurasi OCR yang dilakukan menggunakan digit pada meteran air pelanggan di lapangan, diperoleh nilai akurasi OCR sebesar 87,74%, yang menandakan bahwa sebagian besar digit yang diuji berhasil dikenali

dan diprediksi dengan benar oleh sistem. Hasil tersebut juga mengindikasikan bahwa model mampu melakukan pengenalan digit meteran air secara cukup konsisten meskipun citra yang digunakan berasal dari kondisi lingkungan lapangan yang beragam. Meskipun demikian, sistem OCR masih sering mengalami kesalahan prediksi, terutama ketika kondisi kaca pada meteran air tidak jernih. Hal ini disebabkan oleh adanya goresan yang dapat mengganggu bentuk digit, embun yang menyebabkan citra menjadi buram (*blur*), serta kotoran seperti debu yang telah menempel dalam waktu lama sehingga menimbulkan *noise* pada citra. Sebaliknya, apabila kondisi kaca pada meteran air dalam keadaan jernih dan bersih, sistem OCR cenderung mampu menghasilkan prediksi digit yang lebih akurat dan sesuai dengan nilai aslinya.

4.3 Integrasi Islam

Pada bagian ini dilakukan integrasi antara hasil penelitian dengan nilai-nilai ajaran Islam sebagai landasan dalam memahami dan memaknai pemanfaatan teknologi. Integrasi ini bertujuan untuk menunjukkan bahwa pengembangan sistem berbasis teknologi, termasuk dalam penelitian ini, tidak terlepas dari nilai-nilai keislaman yang mengatur hubungan manusia dengan Allah serta hubungan manusia dengan sesama.

4.3.1 *Muamalah Ma'a Allah*

Muamalah Ma'a Allah merupakan konsep yang mengatur hubungan antara manusia dengan Allah SWT, yang mencakup aspek keimanan, ketaatan, serta kesadaran bahwa setiap aktivitas yang dilakukan oleh manusia berada dalam

pengawasan-Nya. Dalam konteks ini, segala bentuk usaha dan pekerjaan, termasuk pengembangan teknologi, dapat dipandang sebagai bagian dari ibadah apabila dilakukan dengan niatan yang baik dan berdasar pada prinsip-prinsip yang telah ditetapkan dalam ajaran islam.

Sebagaimana firman Allah SWT dalam Al-Qur'an surah *Az-Zalzalah* ayat 7-8 yang menyatakan bahwa setiap perbuatan manusia, sekecil apa pun, akan mendapatkan balasan yang setimpal.

فَمَنْ يَعْمَلْ مِثْقَالَ ذَرَّةٍ خَيْرًا يَرَهُ ۖ وَمَنْ يَعْمَلْ مِثْقَالَ ذَرَّةٍ شَرًّا يَرَهُ ۝٨

“Barangsiapa yang mengerjakan kebaikan seberat zarah, dia akan melihat (balasan)-nya. Dan barangsiapa yang mengerjakan kejahatan seberat zarah, dia akan melihat (balasan)-nya pula.” (QS. Al-Zalzalah’: 7-8).

Menurut Imam Al-Qurthubi dalam tafsirnya *“Al-Jāmi’ li Ahkām al-Qur’an”*, ayat ini menjelaskan bahwa seluruh amal perbuatan manusia akan diperlihatkan dan diberikan balasan secara adil oleh Allah SWT, tanpa ada satu pun yang terlewat, baik perbuatan kecil maupun besar. Hal ini menegaskan adanya prinsip pertanggungjawaban yang menyeluruh atas setiap tindakan manusia di hadapan Allah SWT.

Keterkaitan konsep ini dengan penelitian yang dilakukan terletak pada pentingnya menjaga nilai kejujuran dan amanah dalam pengembangan sistem OCR untuk pembacaan meteran air. Sistem yang dibangun harus mampu memberikan hasil yang akurat dan dapat dipercaya, sehingga tidak merugikan pihak mana pun. Selain itu, proses pengembangan dan implementasi sistem juga harus dilakukan

secara bertanggung jawab, dengan memastikan bahwa teknologi yang digunakan memberikan manfaat dan tidak menimbulkan dampak negatif.

4.3.2 *Muamalah Ma'a An-Naas*

Muamalah Ma'a An-Naas merupakan konsep yang mengatur hubungan antara manusia dengan sesama manusia dalam kehidupan sosial. Konsep ini menitikberatkan pentingnya nilai keadilan, kejujuran, amanah, serta memberikan kemudahan dan kemaslahatan bagi orang lain dalam setiap aktivitas. Dalam konteks perkembangan teknologi, penerapan nilai-nilai Muamalah Ma'a An-Naas dapat diwujudkan melalui pengembangan sistem yang tidak hanya berfungsi secara teknis, tetapi juga memberikan manfaat nyata bagi masyarakat serta meminimalkan potensi kesalahan yang dapat merugikan pihak tertentu.

Sebagaimana firman Allah SWT dalam Al-Qur'an surah *An-Nahl* ayat 90 yang memerintahkan untuk berlaku adil, berbuat kebajikan, dan memberikan kepada sesama.

إِنَّ اللَّهَ يَأْمُرُ بِالْعَدْلِ وَالْإِحْسَنِ وَإِيتَايَ ذِي الْقُرْبَىٰ وَيَنْهَىٰ عَنِ الْفَحْشَاءِ وَالْمُنْكَرِ وَالْبَغْيِ يَعِظُكُمْ لَعَلَّكُمْ تَذَكَّرُونَ ٩٠

“Sesungguhnya Allah menyuruh (kamu) berlaku adil dan berbuat kebajikan, memberi kepada kaum kerabat, dan Allah melarang dari perbuatan keji, kemungkaran dan permusuhan. Dia memberi pengajaran kepadamu agar kamu dapat mengambil pelajaran.” (QS. *An-Nahl*: 90).

Menurut Muhammad Nasib Ar-Rifa'i dalam kitab “*Ringkasan Tafsir Ibnu Katsir*”, Allah SWT menjelaskan bahwa Dia memerintahkan hamba-hamba-Nya untuk berlaku adil, yaitu bersikap seimbang dan tidak berlebihan, serta menganjurkan untuk berbuat ihsan. Dalam ayat tersebut terdapat tiga bentuk perintah, yaitu berlaku adil, berbuat ihsan, serta memberikan bantuan kepada kaum

kerabat. Di sisi lain, ayat ini juga memuat tiga larangan, yaitu perbuatan keji (*fahsyah*), kemungkaran (*munkar*), dan tindakan permusuhan (*baghy*). Hal ini menandakan bahwa ajaran Islam sangat menekankan pentingnya nilai keadilan, kebaikan, dan kepedulian dalam kehidupan sosial.

Keterkaitan konsep tersebut dengan penelitian ini terletak pada tujuan pengembangan sistem OCR dalam membaca meteran air, yaitu untuk membantu proses pencatatan data penggunaan air pelanggan. Dengan adanya sistem ini, proses pencatatan yang sebelumnya dilakukan secara manual dapat dipermudah, sehingga dapat menghemat waktu dan mengurangi potensi kesalahan pencatatan. Selain itu, penerapan sistem ini juga merupakan bentuk upaya dalam membantu sesama manusia, khususnya petugas dan pelanggan, melalui peningkatan kualitas layanan yang lebih cepat, tepat, dan transparan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa aplikasi *mobile* OCR berhasil dikembangkan untuk membantu proses pembacaan digit meteran air dan pencatatan hasil pembacaan pada PAMDes Pandanrejo. Selain itu, hasil penelitian menunjukkan bahwa variasi parameter pelatihan model memengaruhi kemampuan sistem dalam melakukan pengenalan digit meteran air. Penggunaan konfigurasi yang tepat menghasilkan performa pengenalan yang lebih baik dibandingkan konfigurasi lainnya, sehingga berkontribusi terhadap peningkatan kualitas pembacaan meteran air.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, penelitian selanjutnya disarankan untuk mengembangkan sistem dengan memanfaatkan teknologi yang lebih lanjut, baik dari sisi model maupun tahap *preprocessing*. Hal ini dapat dilakukan, misalnya, dengan menambahkan metode segmentasi berbasis kontur (*contour-based segmentation*) atau teknik pengolahan citra lainnya guna meningkatkan akurasi pembacaan, khususnya pada kondisi data lapangan yang lebih kompleks.

DAFTAR PUSTAKA

- Abdi, N. M., Aisyah, S., & Arnia, D. F. (2011). Peningkatan Kualitas Citra Digital Menggunakan Metode Super Resolusi Pada Domain Spasial. *Jurnal Rekayasa Elektrika*, 9(3).
- Akay, M., Du, Y., Sershen, C. L., Wu, M., Chen, T. Y., Assassi, S., Mohan, C., & Akay, Y. M. (2021). Deep Learning Classification of Systemic Sclerosis Skin Using the MobileNetV2 Model. *IEEE Open Journal of Engineering in Medicine and Biology*, 2, 104–110. <https://doi.org/10.1109/OJEMB.2021.3066097>
- Alfajri, M. D., Suroso, & Salamah, I. (2025). Implementasi Sistem Parkir Cerdas Berbasis IoT dan QR Code dengan OCR untuk Segmentasi Plat Nomor Kendaraan. *SemanTIK*, 11(2).
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1). <https://doi.org/10.1186/s40537-021-00444-8>
- Arafat, F. A., Pamungkas, D. P., & Kasih, P. (2025). Pemodelan Klasifikasi Penyakit Daun Tembakau Dengan Arsitektur MobileNetV2. *INOTEK*, 9, 2549–7952.
- Ario, D. A., & Prasetyo, H. B. (2017). PENERAPAN OPTIMISASI PRUNING DAN CLUSTERING TENSORFLOW LITE PADA EMBEDDED SYSTEM UNTUK SISTEM SPEECH TO TEXT. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 1(1), 2548–2964. <http://j->
- Ariyana, R. Y., Susanti, E., Ath-Thaariq, M. R., & Apriadi, R. (2023). Penerapan Uji Fungsionalitas Menggunakan Black Box Testing pada Game Motif Batik Khas Yogyakarta. *JUMINTAL: Jurnal Manajemen Informatika Dan Bisnis Digital*, 2(1), 33–43. <https://doi.org/10.55123/jumintal.v2i1.2371>
- Arnita, Marpaung, F., Aulia, F., Suryani, N., & Nabila, R. C. (2022). *COMPUTER VISION DAN PENGOLAHAN CITRA DIGITAL* (A. B. Surya, Ed.). PUSTAKA AKSARA. www.pustakaaksara.co.id
- Azhary, M. Z. R., & Ismail, A. R. (2024). Comparative Performance of Different Convolutional Neural Network Activation Functions on Image Classification. *International Journal on Perceptive and Cognitive Computing*, 10(2), 118–122. <https://doi.org/10.31436/ijpcc.v10i2.490>
- Azmi, K., Defit, S., & Putra Indonesia YPTK Padang Jl Raya Lubuk Begalung-Padang-Sumatera Barat, U. (2023). Implementasi Convolutional Neural

- Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat. *Jurnal Unitek*, 16(1), 2023.
- Cholissodin, I., Sutrisno, Soebroto, A. A., Hasanah, U., & Febiola, Y. I. (2020). *AI, Machine Learning & Deep Learning (Teori & Implementasi)*. <http://bit.ly/3piOnnU>
- Cossio, M. (2023). *Augmenting Medical Imaging: A Comprehensive Catalogue of 65 Techniques for Enhanced Data Analysis*. <http://arxiv.org/abs/2303.01178>
- Darpito, M. N., Firdausy, K., & Fadlil, A. (2025). Perbandingan Unjuk Kerja Library Optical Character Recognition (OCR) dalam Pengenalan Teks pada Dokumen Digital. *JIP (Jurnal Informatika Polinema)*, 11(3).
- Dijaya, R. (2023). *BUKU AJAR PENGOLAHAN CITRA DIGITAL* (M. Tanzil & Mahardika, Eds.). UMSIDA PRESS. <https://doi.org/https://doi.org/10.21070/2023/978-623-464-075-5>
- Goel, P., & Ganatra, A. (2023). Handwritten Gujarati Numerals Classification Based on Deep Convolution Neural Networks Using Transfer Learning Scenarios. *IEEE Access*, 11, 20202–20215. <https://doi.org/10.1109/ACCESS.2023.3249787>
- Grandini, M., Bagli, E., & Visani, G. (2020). *Metrics for Multi-Class Classification: an Overview*. <http://arxiv.org/abs/2008.05756>
- Hanif, A. R., Nasrullah, E., & Setyawan, F. X. A. (2023). DETEKSI KARAKTER PLAT NOMOR KENDARAAN DENGAN MENGGUNAKAN METODE OPTICAL CHARACTER RECOGNITION (OCR). *JITET (Jurnal Informatika Dan Teknik Elektro Terapan)*, 11(1). <https://doi.org/http://dx.doi.org/10.23960/jitet.v11i1.2897>
- Hardi, N., & Sundari, J. (2022). Pengenalan Telapak Tangan Menggunakan Convolutionall Neural Network (CNN). *Jurnal Rekayasa Perangkat Lunak*, 4(1). <http://jurnal.bsi.ac.id/index.php/reputasi>
- Hicks, S. A., Strümke, I., Thambawita, V., Hammou, M., Riegler, M. A., Halvorsen, P., & Parasa, S. (2022). On Evaluation Metrics for Medical Applications of Artificial Intelligence. *Scientific Reports*, 12(1). <https://doi.org/10.1038/s41598-022-09954-8>
- Jain, G., Mittal, D., Thakur, D., & Mittal, M. K. (2020). A Deep Learning Approach to Detect Covid-19 Coronavirus with X-Ray Images. *Biocybernetics and Biomedical Engineering*, 40(4), 1391–1405. <https://doi.org/10.1016/j.bbe.2020.08.008>
- Jang, B., Kim, M., Harerimana, G., Kang, S. U., & Kim, J. W. (2020). Bi-LSTM Model to Increase Accuracy in Text Classification: Combining Word2vec CNN and Attention Mechanism. *Applied Sciences (Switzerland)*, 10(17). <https://doi.org/10.3390/app10175841>

- Karthick, K., Ravindrakumar, K. B., Francis, R., & Ilankannan, S. (2019). Steps Involved in Text Recognition and Recent Research in OCR; A Study. In *International Journal of Recent Technology and Engineering (IJRTE)* (Number 8).
- Khotimah, S. H., Aulia, M., Athirah, N. A. binti S., & Mujahid, A. (2025). TANGGUNGJAWAB INDERA MANUSIA DALAM MENERIMA INFORMASI: QS. AL-ISRA' AYAT 36. *MUSHAF JOURNAL : Jurnal Ilmu Al Quran Dan Hadis*, 5(2), 2809–3712. <https://www.brilio.net/ragam/ulasan-buku-thinking-fast-and-slow-oleh-daniel-kahneman-240912x.html>
- Kusmacherusanni, N. D., & Ismanto, S. U. (2024). Tata Kelola Badan Usaha Milik Desa (BUMdes) Mekarsari, Kabupaten Garut Dalam Meningkatkan Perekonomian Masyarakat Desa. *INNOVATIVE: Journal Of Social Science Research*, 4, 2925–2942.
- Lathif, M. M., & Novianto, S. (2025). Turtle Dove Classification Using CNN Algorithm With MobileNetV2 Transfer Learning. *Journal of Applied Informatics and Computing (JAIC)*, 9(4), 1701–1708. <http://jurnal.polibatam.ac.id/index.php/JAIC>
- Lehtonen, O. (2024). *On-Device Machine Learning Inference using Cross-Platform Frameworks*. <http://www.cs.helsinki.fi/>
- Lestari, I. S., Jumadi, & Lukman, N. (2024). IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK DENGAN PRE-TRAINED MODEL MOBILENETV2 UNTUK DETEKSI KOLESTEROL. *Rabit : Jurnal Teknologi Dan Sistem Informasi Univrab*, 9(2), 173–183. <https://doi.org/10.36341/rabit.v9i2.4732>
- Liu, Y., Pu, H., & Sun, D. W. (2021). Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing complex food matrices. In *Trends in Food Science and Technology* (Vol. 113, pp. 193–204). Elsevier Ltd. <https://doi.org/10.1016/j.tifs.2021.04.042>
- Lukman, L. A., & Fuad, E. (2026). STUDI LITERATUR : PERBANDINGAN ARSITEKTUR CNN UNTUK KLASIFIKASI TUMOR OTAK. *Jurnal Mahasiswa Teknik Informatika*, 10(3).
- Mahardika, F., Kabul, A. P., & Saputra, D. I. S. (2017). IMPLEMENTASI METODE WATERFALL PADA PROSES DIGITALISASI CITRA ANALOG. *Jurnal Ilmiah Pendidikan Teknik Elektro*, 2(1), 63–72. <https://doi.org/https://dx.doi.org/10.30870/volt.v2i1.948>
- Mesran, Yahya, S. R., Nugroho, F., & Windarto, A. P. (2024). Investigating the Impact of ReLU and Sigmoid Activation Functions on Animal Classification Using CNN Models. *Jurnal RESTI*, 8(1), 111–118. <https://doi.org/10.29207/resti.v8i1.5367>

- Miftahuddin, Y., & Zaelani S, F. (2022). Perbandingan Metode Efficientnet-B3 dan Mobilenet-V2 Untuk Identifikasi Jenis Buah-buahan Menggunakan Fitur Daun. *Jurnal Ilmiah Teknologi Informasi Terapan*, 9(1).
- Mulyadi, J., & Veronika, D. N. M. (2025). Penerapan Optical Character Recognition (OCR) dalam Pengambilan Teks Judul Buku untuk Digitalisasi Arsip Perpustakaan. *JSAI : Journal Scientific and Applied Informatics*, 08(2), 447–454. <https://doi.org/10.36085>
- Naufal, M. F., & Kusuma, S. F. (2021). PENDETEKSI CITRA MASKER WAJAH MENGGUNAKAN CNN DAN TRANSFER LEARNING. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIK)*, 8(6), 1293–1300. <https://doi.org/10.25126/jtiik.202185201>
- Nusantara, B. S., & Akbar, M. (2024). Klasifikasi Penyakit Tuberculosis Berdasarkan Citra Rontgen Thorax Menggunakan Multi-Scale Convolutional Neural Network. *Sudo Jurnal Teknik Informatika*, 3(1), 1–8. <https://doi.org/10.56211/sudo.v3i1.483>
- Paksi, Y. E. E., & Prihartono, E. (2019). Sistem Monitoring Pemakaian Air Pdam Tirta Kencana Kota Samarinda Berbasis Arduino. *JIMP-Jurnal Informatika Merdeka Pasuruan*, 4(2).
- Putra, E. D., Utami, M., & Rifqo, M. H. (2021). Identifikasi Text Meteran Air Menggunakan Metode Run-Length Smearing Algorithm (Rlsa). *JUKOMIKA - (JURNAL ILMU KOMPUTER DAN INFORMATIKA)*, 4(2).
- Rahman, Moch. M., Triana, F. P., Salsabila, Z. S., Akbar, R. F., Alfatih, S. M., Mahdi, P. Z. Al, & Utami, H. A. (2025). RANCANG BANGUN DAN IMPLEMENTASI APLIKASI PEMBAYARAN PAM DESA PANDANREJO BERBASIS WEB DENGAN METODOLOGI SCRUM. *Jurnal Rekayasa Sistem Informasi Dan Teknologi*.
- Ramadhan, J. A., Haniva, D. T., & Suharso, A. (2023). Systematic Literature Review Penggunaan Metodologi Pengembangan Sistem Informasi Waterfall, Agile, dan Hybrid. *JIEET (Journal Information Engineering and Educational Technology)*, 07(01).
- Ramchandani, M., Khandare, H., Singh, P., Rajak, P., Suryawanshi, N., Jangde, A. S., Arya, L., Kumar, P., & Sahu, M. (2022). Survey: Tensorflow in Machine Learning. *Journal of Physics: Conference Series*, 2273(1). <https://doi.org/10.1088/1742-6596/2273/1/012008>
- Rehman, A., Mohamad, D., & Sulong, G. (2009). Implicit Vs Explicit based Script Segmentation and Recognition: A Performance Comparison on Benchmark Database. *Int. J. Open Problems Compt. Math*, 2(3). www.i-csrs.org
- Riad, R., Teboul, O., Grangier, D., & Zeghidour, N. (2022). Learning strides in convolutional neural networks. *ICLR*. <http://arxiv.org/abs/2202.01653>

- Sainyakit, J. R. R., & Bisilisin, F. Y. (2025). KLASIFIKASI PENGGUNAAN METERAN AIR MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) PADA PERUMDA AIR MINUM KOTA KUPANG BERBASIS CITRA DIGITAL. *Jurnal Mahasiswa Teknik Informatika*, 9(4), 5773.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *IEEE Explore*.
- Saputra, D., Silvana, H., & Khoerunnisa, L. (2024). Advantages and disadvantages of technological implementation in archive systems: A systematic literature review. *Record and Library Journal*, 10(2), 388–412. <https://doi.org/10.20473/rlj.V10-I2.2024.388-412>
- Saputri, I. W. N., Harahap, L. S., & Baihaqi, A. (2025). Konversi Citra Rgb Ke Grayscale Menggunakan Python. *Jurnal JUPITER*, 845–855.
- Sari, R., Musa Adi, I., & Hidayati, A. (2023). Personal Track Your Cash: Prototipe Aplikasi Pembacaan Struk 0020 Belanja Menggunakan OCR dan Google Vision. *SNIV: SEMINAR NASIONAL INOVASI VOKASI*, 2(1).
- Shawon, M. T. R., Tanvir, R., & Alam, Md. G. R. (2022). Bengali Handwritten Digit Recognition using CNN with Explainable AI. *IEEE Xplore*. <http://arxiv.org/abs/2212.12146>
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1). <https://doi.org/10.1186/s40537-019-0197-0>
- Suci, B. D., Musfiroh, Sefriani, S. P., Sumanto, Pakpahan, R., & Budiawan, I. (2026). KLASIFIKASI PENYAKIT DAUN TANAMAN BERBASIS CITRA MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK DATA AUGMENTATION. In *Seminar Nasional Riset dan Inovasi Teknologi (SEMNAS RISTEK)*. <https://www.kaggle.com/datasets/rashikrahm>
- Tanjung, D. A., Lestari, T. A., Harahap, L. S., & Kesuma, M. R. (2025). Implementasi Algoritma Convolutional Neural Network (CNN) Untuk Pengenalan Angka Tulisan Tangan. *Data Sciences Indonesia (DSI)*, 5(2), 93–100. <https://doi.org/10.47709/dsi.v5i2.7357>
- Thanki, J. D., Davda, P. D., & Swaminarayan, P. (2021). A Review on OCR Technology. *Journal of Emerging Technologies and Innovative Research (JETIR)*, 8(4). www.jetir.org
- Toyib, M., Pratama, T. D. K., & Aqil, I. (2024). Penerapan Algoritma CNN Untuk Mendeteksi Tulisan Tangan Angka Romawi dengan Augmentasi Data. *Algoritma : Jurnal Matematika, Ilmu Pengetahuan Alam, Kebumihan Dan Angkasa*, 2(3), 108–120. <https://doi.org/10.62383/algoritma.v2i3.69>
- Ulum, M. R. B., Rahmat, B., & Swari, M. H. P. (2024). Implementasi Metode CNN Dan K-Nearest Neighbor Untuk Klasifikasi Tingkat Kematangan Tanaman

Cabai Rawit. *Jurnal Informatika Dan Sains Teknologi*, 1(3), 112–123.
<https://doi.org/10.62951/modem.v1i3.131>

Wardani, K. R. R., & Leonardi, L. (2023). Klasifikasi Penyakit pada Daun Anggur menggunakan Metode Convolutional Neural Network. *Jurnal Tekno Insentif*, 17(2), 112–126. <https://doi.org/10.36787/jti.v17i2.1130>

Wei, H., Liu, C., Chen, J., Wang, J., Kong, L., Xu, Y., Ge, Z., Zhao, L., Sun, J., Peng, Y., Han, C., & Zhang, X. (2024). *General OCR Theory: Towards OCR-2.0 via a Unified End-to-end Model*.
<https://doi.org/https://doi.org/10.48550/arXiv.2409.01704>

Zeiler, M. D., & Fergus, R. (2013). *Visualizing and Understanding Convolutional Networks*. <http://arxiv.org/abs/1311.2901>

Zhang, H., Chen, M., Ni, M., Chen, L., Zhang, Y., & Hao, X. (2024). Automated feature map padding and transfer circuit for CNN inference. *IEICE Electronics Express*, 21(22). <https://doi.org/10.1587/elex.21.20240559>

Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M., & Parmar, M. (2024). A review of convolutional neural networks in computer vision. *Artificial Intelligence Review*, 57(4). <https://doi.org/10.1007/s10462-024-10721-6>

LAMPIRAN

Lampiran 1. Dokumentasi proses pencatatan manual.

