



Universitas Nusantara PGRI Kediri

UPT. PERPUSTAKAAN, PUBLIKASI DAN INOVASI

Alamat: Kampus 1, Jl. KH. Ahmad Dahlan No.76 Kota Kediri 64112
Telp. (0354) 771576, (0354) 771503, (0354) 771495, Fax. (0354) 771576
Website: <http://ppi.unpkediri.ac.id/> Email: perpustakaan@unpkediri.ac.id

SURAT KETERANGAN BEBAS SIMILARITY

Ketua UPT Perpustakaan, Publikasi dan Inovasi Universitas Nusantara PGRI Kediri menerangkan bahwa mahasiswa dengan identitas berikut:

Nama Mahasiswa : Ilham Khefi Ramadhanu
NPM : 2213020120
Program Studi : S1-Teknik Informatika

Judul Karya Ilmiah:

“PENGEMBANGAN SISTEM PELAPORAN KERUSAKAN JALAN DENGAN KLASIFIKASI DAN DETEKSI MENGGUNAKAN RESNET-50 DAN YOLOV12 BERBASIS ANDROID”

Dinyatakan sudah memenuhi syarat batas maksimal 30% *similarity* sesuai dengan ketentuan yang berlaku pada setiap subbab naskah Laporan **Tugas Akhir/Skripsi/Tesis** yang disusun.

Demikian Surat Keterangan ini kami berikan untuk dapat dipergunakan seperlunya.



Kediri, 18 Juni 2026
Ka UPT PPI,



Dr. Abdul Aziz Hunaifi, M.A

Turnitin LLCC

2213020120_Ilham Khefi Ramadhanu

 TI-04

Document Details

Submission ID**trn:oid::2945:394030777****Submission Date****Jun 18, 2026, 2:49 PM GMT+7****Download Date****Jun 18, 2026, 2:53 PM GMT+7****File Name****2213020120_Ilham Khefi Ramadhanu.pdf****File Size****2.3 MB****89 Pages****16,892 Words****105,913 Characters**

22% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Top Sources

- 11%  Internet sources
- 9%  Publications
- 20%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Top Sources

- 11% Internet sources
- 9% Publications
- 20% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Student papers	Universitas Khairun on 2026-05-11	<1%
2	Student papers	Universitas Gunadarma on 2025-12-08	<1%
3	Student papers	STT PLN on 2025-08-11	<1%
4	Student papers	Perpustakaan on 2025-08-07	<1%
5	Internet	repository.telkomuniversity.ac.id	<1%
6	Internet	jurnal.uinsu.ac.id	<1%
7	Internet	docplayer.info	<1%
8	Student papers	Universitas Islam Indonesia on 2025-07-14	<1%
9	Student papers	Universitas Islam Indonesia on 2025-09-17	<1%
10	Internet	dspace.uil.ac.id	<1%
11	Internet	kc.umh.ac.id	<1%

12	Internet	repository.its.ac.id	<1%
13	Student papers	UIN Syarif Hidayatullah Jakarta on 2023-07-30	<1%
14	Student papers	Telkom University on 2025-07-07	<1%
15	Student papers	Universitas Putera Batam on 2026-01-29	<1%
16	Internet	123dok.com	<1%
17	Student papers	LL Dikti IX Turnitin Consortium on 2026-04-28	<1%
18	Student papers	Telkom University on 2026-01-22	<1%
19	Internet	eprints.upj.ac.id	<1%
20	Internet	id.123dok.com	<1%
21	Internet	repository.unissula.ac.id	<1%
22	Student papers	UPN Veteran Yogyakarta on 2026-03-02	<1%
23	Student papers	Universitas Muhammadiyah Buton on 2026-06-05	<1%
24	Student papers	Universitas Islam Riau on 2026-03-26	<1%
25	Student papers	UIN Raden Intan Lampung on 2026-05-11	<1%

26	Student papers	UPN Veteran Yogyakarta on 2026-05-25	<1%
27	Internet	eprints.uny.ac.id	<1%
28	Student papers	Perpustakaan on 2025-08-06	<1%
29	Student papers	Perpustakaan on 2025-09-02	<1%
30	Student papers	Universitas Pamulang on 2026-05-04	<1%
31	Internet	conf.unnes.ac.id	<1%
32	Publication	Ade Wardani, Miftahurrahma Rosyda. "Implementasi Aplikasi Laporjalanku untu...	<1%
33	Publication	Arkananta Emier, Daffa Arief, Aqila Adam, Jubel Hiero Oktovan Lumban Gaol et al...	<1%
34	Student papers	Universitas Islam Indonesia on 2026-06-15	<1%
35	Student papers	Universitas Muhammadiyah Makassar on 2026-04-29	<1%
36	Student papers	Universitas Putera Batam on 2025-02-05	<1%
37	Internet	repositori.usu.ac.id	<1%
38	Student papers	Boston University on 2026-05-13	<1%
39	Publication	Dila Aura Putri, Ivana Lucia Kharisma, Somantri Somantri. "Implementasi Model ...	<1%

40	Publication	Takyudin Takyudin, Muhammad Sandi Rais, Jonni Adi Putra, Ali Hamsar. "Metode ...	<1%
41	Publication	Taryono Ibnu Affan, Dian Ade Kurnia -, Irfan Ali -. "APLIKASI CARSECURE MENGGU...	<1%
42	Student papers	Telkom University on 2025-07-04	<1%
43	Student papers	Universitas Islam Riau on 2026-04-30	<1%
44	Student papers	Universitas Islam Riau on 2026-05-13	<1%
45	Internet	jurnal.unimed.ac.id	<1%
46	Internet	ojs.uajy.ac.id	<1%
47	Publication	Muhammad Adrezo, Muhamad Erian Ardiansyah. "Deteksi Jenis Kelamin Berdasa...	<1%
48	Student papers	Telkom University on 2026-01-12	<1%
49	Student papers	Universitas Dian Nuswantoro on 2026-01-09	<1%
50	Student papers	Universitas Islam Riau on 2026-05-12	<1%
51	Student papers	Universitas Sumatera Utara on 2025-05-23	<1%
52	Publication	Afifah Dwi Ramadhani, Arva Zaki Fanadzan, Mochammad Mobed Bachtiar, Aji Sap...	<1%
53	Student papers	Telkom University on 2025-07-08	<1%

54	Student papers	Universitas Tarumanagara on 2026-01-02	<1%
55	Publication	Yozi Heru Maulana, Nurhasan Nugroho, Gelard Untirtha Pratama. "Perancangan ...	<1%
56	Internet	ejurnal.methodist.ac.id	<1%
57	Student papers	UPN Veteran Yogyakarta on 2026-03-31	<1%
58	Student papers	Universitas Negeri Surabaya The State University of Surabaya on 2026-04-21	<1%
59	Student papers	Universitas Pancasila on 2026-03-04	<1%
60	Internet	www.coursehero.com	<1%
61	Student papers	LL Dikti IX Turnitin Consortium on 2025-09-29	<1%
62	Student papers	Politeknik Negeri Lhokseumawe Turnitin on 2026-02-10	<1%
63	Student papers	Sriwijaya University on 2026-04-28	<1%
64	Internet	www.pnpm-alu.org	<1%
65	Student papers	Institut Bisnis dan Teknologi Indonesia (INSTIKI) on 2025-06-08	<1%
66	Student papers	Sriwijaya University on 2026-05-04	<1%
67	Student papers	Telkom University on 2025-07-02	<1%

68	Student papers	Universitas Gunadarma on 2025-12-08	<1%
69	Student papers	Universitas Pancasila on 2025-08-10	<1%
70	Student papers	Universitas Tarumanagara on 2025-12-14	<1%
71	Student papers	Universitas Tarumanagara on 2025-12-24	<1%
72	Student papers	UPN Veteran Jawa Timur on 2025-06-25	<1%
73	Student papers	Universitas Islam Riau on 2025-11-21	<1%
74	Student papers	Universitas Islam Riau on 2026-05-18	<1%
75	Student papers	Universitas Jember on 2025-12-08	<1%
76	Student papers	Universitas Kristen Duta Wacana on 2026-05-06	<1%
77	Student papers	Universitas Malikussaleh on 2026-01-05	<1%
78	Student papers	Universitas Putera Batam on 2026-02-02	<1%
79	Internet	digilib.unila.ac.id	<1%
80	Internet	dspace.daffodilvarsity.edu.bd:8080	<1%
81	Internet	holidincom.blogspot.com	<1%

82	Internet	repository.ub.ac.id	<1%
83	Internet	repository.umsu.ac.id	<1%
84	Student papers	Universitas Islam Riau on 2025-09-02	<1%
85	Student papers	Universitas Muhammadiyah Purwokerto on 2025-11-19	<1%
86	Internet	digilibadmin.unismuh.ac.id	<1%
87	Internet	eprints.iain-surakarta.ac.id	<1%
88	Internet	repository.unpkediri.ac.id	<1%
89	Student papers	Academic Library Consortium on 2025-05-24	<1%
90	Student papers	Perpustakaan on 2025-08-08	<1%
91	Student papers	Universitas Islam Indonesia on 2026-05-12	<1%
92	Student papers	Universitas Muria Kudus on 2026-04-15	<1%
93	Publication	Vanness Bee, Ery Hartati. "Aplikasi Absensi Pengenalan Wajah dengan Mengguna...	<1%
94	Internet	ejurnal.seminar-id.com	<1%
95	Internet	informatika.unpkediri.ac.id	<1%

96	Internet	repository.um-surabaya.ac.id	<1%
97	Publication	Achmad Rafi, Jeffri Alfa Razaq. "RANCANG BANGUN APLIKASI PEMESANAN MENU ...	<1%
98	Publication	Murtiwiayati Murtiwiayati, Risha Fitriaswhalia Arini, Leli Safitri, Santi Widiyanti, Iwa...	<1%
99	Student papers	UPN Veteran Yogyakarta on 2026-06-02	<1%
100	Student papers	Universitas Diponegoro on 2019-01-28	<1%
101	Student papers	Universitas Diponegoro on 2025-03-05	<1%
102	Student papers	Universitas Katolik Musi Charitas on 2026-02-09	<1%
103	Student papers	Universitas Muhammadiyah Sukabumi on 2026-01-19	<1%
104	Student papers	Universitas Pancasila on 2026-03-02	<1%
105	Internet	brainly.com	<1%
106	Student papers	itera on 2025-02-04	<1%
107	Internet	lawumedia.com	<1%
108	Internet	repository.iainpalopo.ac.id	<1%
109	Student papers	Fakultas Ekonomi Universitas Indonesia on 2017-05-05	<1%

110	Student papers	IAIN Purwokerto on 2025-05-14	<1%
111	Publication	Muhammad Mauladi, Dedy Hermanto. "Klasifikasi Motif Kain Jumputan Palembang...	<1%
112	Student papers	Trisakti University on 2026-01-09	<1%
113	Student papers	Universitas Islam Bandung on 2026-01-21	<1%
114	Student papers	Universitas Islam Riau on 2025-10-31	<1%
115	Student papers	Universitas Putera Batam on 2018-12-01	<1%
116	Student papers	Universitas Tarumanagara on 2025-12-24	<1%
117	Internet	books.digitalkbshah.com	<1%
118	Internet	pt.scribd.com	<1%
119	Internet	www.enigmacamp.com	<1%
120	Publication	Doms Upuy, Victor Eric Pattiradjawane. "Deteksi Tahap Perkembangan Putik Saw...	<1%
121	Publication	Fadila Ananda Kartika Hidayat, Neny Sulistianingsih, Rifqi Hammad. "Perbanding...	<1%
122	Publication	Khusnul Khotimah, Martanto Martanto, Arif Rinaldi Dikananda, Ahmad Rifa'i. "AN...	<1%
123	Publication	Panji Pangestu Saputra, Hasbi Firmansyah, Rizki Prasetyo Tulodo, Priyo Haryok...	<1%

124	Student papers	Politeknik Negeri Lhokseumawe Turnitin on 2026-01-07	<1%
125	Student papers	School of Business and Management ITB on 2025-07-14	<1%
126	Student papers	Universitas Brawijaya on 2021-06-11	<1%
127	Student papers	Universitas Bung Hatta on 2022-02-18	<1%
128	Student papers	Universitas Islam Malang on 2025-07-16	<1%
129	Internet	jurusan.tik.pnj.ac.id	<1%
130	Internet	repository.nusamandiri.ac.id	<1%
131	Publication	Ahmad Munawir, Farhan Maulana, Wahyu Amaldi. "Implementasi Augmented Re...	<1%
132	Publication	Devi Fajar Wati, Lila Setiyani, Deden Moch Alfiansyah, M. Jembar Jomantara, De...	<1%
133	Publication	Ika Arthalia Wulandari, Mohammad Bintang Ramadhan, Muhammad Ridwan, Mu...	<1%
134	Student papers	Politeknik Manufaktur Negeri Bangka Belitung on 2025-11-06	<1%
135	Publication	Rahmawati Rahmawati, Nurhikma Arifin, Wawan Firgiawan. "Perbandingan SVM ...	<1%
136	Student papers	UIN Siber Syekh Nurjati Cirebon on 2026-06-15	<1%
137	Student papers	UPN Veteran Yogyakarta on 2026-04-30	<1%

138	Student papers	Universitas Andalas on 2024-07-13	<1%
139	Student papers	Universitas Islam Riau on 2025-07-17	<1%
140	Student papers	Universitas Tarumanagara on 2025-12-19	<1%
141	Student papers	Universitas Tarumanagara on 2026-01-04	<1%
142	Internet	ejournal.itn.ac.id	<1%
143	Internet	repository.uksw.edu	<1%
144	Internet	vokasi.unair.ac.id	<1%
145	Publication	Asykur Ahmad, Syarifah Rahmah, Agung Fatwanto. "Perancangan penambahan fi...	<1%
146	Publication	Evi Syuriani Harahap, Dinda Rumapea, Wulan Pertiwi Br Sinulingga, Angelica Try ...	<1%
147	Student papers	FAKULTAS ILMU KOMPUTER on 2026-01-11	<1%
148	Publication	Fitri Wibowo, Freska Rolansa, Tri Bowo Atmojo. "Sistem Informasi Inventaris Unt...	<1%
149	Student papers	Indian School of Business on 2025-05-31	<1%
150	Publication	Muhammad Sidiq Pramono, Aditya Permana Wibowo. "PENERAPAN CONVOLUTIO...	<1%
151	Student papers	STT PLN on 2025-08-11	<1%

152	Student papers	Sriwijaya University on 2021-08-04	<1%
153	Student papers	Telkom University on 2025-07-08	<1%
154	Student papers	UIN Sultan Syarif Kasim Riau on 2021-02-10	<1%
155	Student papers	UPN Veteran Jawa Timur on 2025-09-11	<1%
156	Student papers	UPN Veteran Yogyakarta on 2026-06-04	<1%
157	Student papers	Universitas Bina Sarana Informatika on 2026-06-11	<1%
158	Student papers	Universitas Brawijaya on 2019-07-29	<1%
159	Student papers	Universitas Brawijaya on 2019-12-30	<1%
160	Student papers	Universitas Jember on 2025-06-16	<1%
161	Student papers	Universitas Kristen Duta Wacana on 2026-05-07	<1%
162	Student papers	Universitas Muhammadiyah Palembang on 2025-07-18	<1%
163	Student papers	Universitas Muria Kudus on 2016-03-29	<1%
164	Student papers	Universitas Pamulang on 2026-06-06	<1%
165	Student papers	Universitas Tarumanagara on 2025-12-18	<1%

166	Internet	doku.pub	<1%
167	Internet	eprints.undip.ac.id	<1%
168	Internet	journal.global.ac.id	<1%
169	Internet	lib.unnes.ac.id	<1%
170	Internet	projekter.aau.dk	<1%
171	Internet	repository.unbara.ac.id	<1%
172	Publication	Ade Kurniawan, Imamulhakim Syahid Putra. "Pengembangan Sistem Informasi D...	<1%
173	Publication	Alfia Meilani Putri. "RANCANG BANGUN WEBSITE E-COMMERCE "RELOVED" UNTU...	<1%
174	Publication	Aprilian Gevindo, Syafri Arlis. "Smoker Detection Using the You Only Look Once (Y...	<1%
175	Student papers	College of Banking and Financial Studies on 2025-12-15	<1%
176	Student papers	Fakultas Teknik on 2025-07-23	<1%
177	Publication	Hari Santoso, Shelly Wati, Sugesti Sugesti, Sobiyanto Sobiyanto. "PERANCANGAN ...	<1%
178	Student papers	Management & Science University on 2025-01-14	<1%
179	Publication	Manda Agustriya, Munirul Ula, Kurniawati -. "Analisis Kinerja Algoritma Klasifikas...	<1%

180	Publication	Mhd Zahir Az Zikri, Muhammad Yasin Simargolang. "Klasifikasi Tingkat Tindak Pi...	<1%
181	Student papers	Politeknik Kesehatan Kemenkes Padang on 2026-04-21	<1%
182	Student papers	Politeknik Manufaktur Negeri Bangka Belitung on 2025-07-03	<1%
183	Student papers	Politeknik Negeri Bandung on 2017-07-31	<1%
184	Publication	Susi Handayani, Fajrizal, Taslim, Dafwen Toresa, Syahril. "Peningkatan Perform...	<1%
185	Student papers	Tarumanagara University on 2024-06-26	<1%
186	Student papers	Tarumanagara University on 2024-12-11	<1%
187	Student papers	UIN Sultan Syarif Kasim Riau on 2021-07-10	<1%
188	Student papers	UPN Veteran Yogyakarta on 2026-01-06	<1%
189	Student papers	UPN Veteran Yogyakarta on 2026-02-27	<1%
190	Student papers	UPN Veteran Yogyakarta on 2026-05-04	<1%
191	Student papers	UPN Veteran Yogyakarta on 2026-06-02	<1%
192	Student papers	Universitas Bengkulu on 2024-08-05	<1%
193	Student papers	Universitas Bina Sarana Informatika on 2026-06-11	<1%

194	Student papers	Universitas Brawijaya on 2018-07-19	<1%
195	Student papers	Universitas Brawijaya on 2023-12-05	<1%
196	Student papers	Universitas Budi Luhur on 2025-07-15	<1%
197	Student papers	Universitas Esa Unggul on 2025-07-23	<1%
198	Student papers	Universitas Indonesia on 2026-06-11	<1%
199	Student papers	Universitas Islam Riau on 2026-05-12	<1%
200	Student papers	Universitas Jambi on 2026-02-07	<1%
201	Student papers	Universitas Jember on 2025-06-17	<1%
202	Student papers	Universitas Khairun on 2026-06-08	<1%
203	Student papers	Universitas Kristen Duta Wacana on 2026-05-07	<1%
204	Student papers	Universitas Muhammadiyah Prof. Dr. Hamka on 2026-04-07	<1%
205	Student papers	Universitas Muhammadiyah Sukabumi on 2025-08-03	<1%
206	Student papers	Universitas Muria Kudus on 2016-09-27	<1%
207	Student papers	Universitas Pendidikan Indonesia on 2022-05-18	<1%

208	Student papers	Universitas Pendidikan Indonesia on 2025-03-13	<1%
209	Student papers	Universitas Putera Batam on 2026-02-02	<1%
210	Student papers	Universitas Sangga Buana YPKP on 2026-02-13	<1%
211	Student papers	Universitas Sebelas Maret on 2022-06-25	<1%
212	Student papers	Universitas Sumatera Utara on 2025-12-22	<1%
213	Student papers	Universitas Tarumanagara on 2025-12-22	<1%
214	Student papers	Universitas Tarumanagara on 2025-12-29	<1%
215	Internet	eprints.unram.ac.id	<1%
216	Internet	eprints.walisongo.ac.id	<1%
217	Internet	es.scribd.com	<1%
218	Internet	journal.anc-aryantonurconsulting.com	<1%
219	Internet	repositori.uin-alauddin.ac.id	<1%
220	Internet	repositori.umsu.ac.id	<1%
221	Internet	repository.dinamika.ac.id	<1%

222	Internet	repository.uinsu.ac.id	<1%
223	Internet	text-id.123dok.com	<1%
224	Student papers	UPN Veteran Jakarta on 2026-04-29	<1%
225	Student papers	UPN Veteran Yogyakarta on 2026-05-05	<1%
226	Student papers	Universitas Muhammadiyah Prof. Dr. Hamka on 2026-04-08	<1%
227	Student papers	UIN Raden Intan Lampung on 2026-05-11	<1%
228	Student papers	UPN Veteran Yogyakarta on 2026-05-05	<1%
229	Student papers	UPN Veteran Yogyakarta on 2026-06-04	<1%
230	Student papers	Universitas Islam Indonesia on 2026-05-12	<1%
231	Student papers	Universitas Islam Negeri Raden Fatah on 2026-03-12	<1%
232	Student papers	Universitas Islam Riau on 2026-04-28	<1%
233	Student papers	Universitas Islam Riau on 2026-06-15	<1%
234	Student papers	Universitas Khairun on 2026-05-04	<1%
235	Student papers	Universitas Muhammadiyah Prof. Dr. Hamka on 2026-04-07	<1%

236

Student papers

Universitas Muhammadiyah Semarang on 2026-02-23

<1%

237

Student papers

Universitas Pancasila on 2026-03-05

<1%

238

Internet

jurnal.untag-sby.ac.id

<1%

BAB I

PENDAHULUAN

A. Latar Belakang

Jalan raya merupakan salah satu infrastruktur utama yang berperan penting dalam mendukung aktivitas ekonomi, sosial, dan mobilitas masyarakat. Kualitas jalan yang baik dapat meningkatkan konektivitas antarwilayah, memperlancar distribusi barang, serta memperkuat pertumbuhan sektor industri dan pariwisata. Namun, kondisi jalan di berbagai wilayah sering kali mengalami penurunan akibat faktor lingkungan, cuaca, beban kendaraan berlebih, serta kurangnya pemeliharaan berkala. Kerusakan seperti lubang dan retakan dapat mengganggu keselamatan pengguna jalan sekaligus menurunkan efisiensi transportasi (Gumelar dkk., 2023).

Penurunan efisiensi transportasi sejalan dengan data jalan nasional non tol tahun 2024 yang menunjukkan bahwa 53,2% jalan berada dalam kondisi sedang, dengan rincian 1.735,72 km rusak ringan dan 541,77 km rusak berat (Kementerian PU, 2025). Kondisi ini mencerminkan masih rendahnya kualitas infrastruktur jalan di Indonesia akibat lemahnya sistem pemantauan dan pelaporan kerusakan yang masih dilakukan secara manual, memakan waktu, biaya besar, serta sering menghasilkan data yang tidak konsisten (Saparudin dkk., 2025).

Permasalahan ketidakkonsistenan hasil inspeksi visual membuka peluang penerapan teknologi *computer vision* yang memungkinkan sistem komputer untuk menafsirkan dan memahami konten visual layaknya penglihatan manusia (Dompeipen dkk., 2021). Berbagai penelitian telah mengembangkan pendekatan *computer vision* untuk mendeteksi kerusakan jalan secara otomatis menggunakan metode seperti *Convolutional Neural Network* (CNN) dan ResNet-50 yang digunakan untuk mengenali pola dan mengklasifikasikan jenis retakan dan lubang permukaan jalan dengan tingkat akurasi yang tinggi yaitu 99% (Ghofur & Ulum, 2025).

Pendekatan lain dalam *computer vision* adalah metode YOLO (*You Only Look Once*), yang dikenal karena kemampuannya mendeteksi objek secara

realtime dengan pemrosesan cepat dan efisien (Gabriel V & Irsyad, 2024). Namun, sebagian besar penelitian masih berfokus pada pengembangan model deteksi tanpa implementasi dalam bentuk aplikasi. Beberapa studi telah membuat aplikasi pelaporan kerusakan jalan berbasis Android dan web menggunakan metode *Object Oriented Analysis and Design (OOAD)* (Yuniar dkk., 2024) serta evaluasi *System Usability Scale (SUS)* (Wardani dkk., 2024), tetapi sebagian besar masih mengandalkan pelaporan manual tanpa integrasi deteksi objek otomatis. *Framework* Django sering digunakan sebagai *backend* untuk *deployment* model *object detection* karena arsitekturnya yang modular memudahkan integrasi model *deep learning* melalui Django REST *Framework*, sehingga komunikasi antara aplikasi klien dan model di sisi server dapat berlangsung secara terstruktur dan efisien (Saputra dkk., 2024).

Meninjau hasil dan keterbatasan penelitian sebelumnya, diperlukan integrasi antara kecerdasan buatan dan sistem pelaporan untuk menciptakan solusi yang lebih aplikatif. Oleh karena itu, penelitian ini mengembangkan sistem berbasis Android untuk pelaporan kerusakan jalan dengan penerapan metode ResNet-50 untuk klasifikasi jenis kerusakan dan YOLOv12 untuk deteksi objek secara otomatis menggunakan arsitektur *client-server* berbasis Django REST API. Hasil penelitian menunjukkan bahwa model ResNet-50 pada konfigurasi ukuran input 800×800 mencapai *precision* sebesar 0,97, *recall* sebesar 0,96, dan *F1-Score* sebesar 0,97, sedangkan model YOLOv12 pada konfigurasi yang sama menghasilkan mAP50 sebesar 0,939, mAP50-95 sebesar 0,711, *precision* sebesar 0,916, dan *recall* sebesar 0,894. Pengujian fungsional menggunakan metode *Black Box Testing* menunjukkan tingkat keberhasilan 100% dari seluruh skenario uji, dengan rata-rata total waktu inferensi sistem sebesar 3.150,66 ms yang mencakup inferensi ResNet-50 rata-rata 1.892,38 ms dan inferensi YOLOv12 rata-rata 859,73 ms. Sistem ini memungkinkan identifikasi kerusakan jalan secara otomatis dan terintegrasi dengan data spasial sehingga dapat mempercepat proses pemantauan serta membantu pengambil kebijakan dalam menentukan prioritas perbaikan jalan.

B. Identifikasi Masalah

Berdasarkan uraian pada latar belakang, didapatkan beberapa masalah sebagai berikut:

1. Proses survei jalan yang umumnya masih bersifat manual menyebabkan keterlambatan dalam memperoleh data dan penanganan kerusakan jalan, sehingga proses pemantauan kondisi jalan menjadi kurang efektif.
2. Data kerusakan jalan yang terkumpul sering kali tidak konsisten karena tingginya subjektivitas dalam penentuan jenis kerusakan jalan, sehingga menyulitkan proses pengolahan data dan pengambilan keputusan lanjutan
3. Kesenjangan dan kurangnya pengembangan aplikasi *implementasi Convolutional Neural Network* untuk klasifikasi kerusakan jalan. Sebagian besar penelitian sebelumnya hanya berfokus pada pembuatan model klasifikasi dan deteksi tanpa penerapan langsung dalam bentuk aplikasi.

C. Rumusan Masalah

Berdasarkan uraian identifikasi masalah yang terjadi di atas, maka dapat dirumuskan yaitu “Bagaimana merancang dan membangun sistem pelaporan kerusakan jalan yang mampu mendeteksi dan mengklasifikasikan jenis kerusakan jalan dengan *Convolutional Neural Network* untuk analisis citra otomatis?”.

D. Batasan Masalah

Berikut merupakan batasan masalah yang digunakan pada tahap penelitian ini untuk memperjelas ruang lingkup dan fokus pembahasan:

1. Penelitian menggunakan dataset campuran dari data primer dan data publik RDD2022, Roboflow, dan Mendeley Data sebagai sumber data pelatihan dan pengujian.
2. Kelas objek yang digunakan dibatasi pada empat kategori kerusakan jalan, yaitu lubang, retak buaya, melintang, dan memanjang.
3. Pelatihan model dilakukan menggunakan *framework* TensorFlow dan Keras untuk klasifikasi, serta Ultralytics untuk *object detection*.

4. Model klasifikasi menggunakan ResNet-50, dengan batasan evaluasi performa mencakup *precision*, *recall*, dan *F1-Score*.
5. Model deteksi objek menggunakan YOLOv12, dengan evaluasi performa dibatasi pada mAP50, mAP50-95, *precision* dan *recall*.
6. Sistem yang dikembangkan hanya diuji pada platform Android, menggunakan bahasa pemrograman Kotlin, dengan metode pengujian sistem berupa *Black Box Testing*.
7. *Deployment* model menggunakan Django, namun hanya difungsikan sebagai *model serving* untuk menerima citra, melakukan inferensi, dan mengembalikan hasil deteksi. Fitur-fitur lain di luar itu tidak dibahas dalam penelitian ini.
8. Penelitian tidak mencakup perbaikan fisik jalan, analisis kebijakan, atau proses penganggaran. Fokus terbatas pada proses deteksi, klasifikasi, pelaporan visual, dan tingkat prioritas perbaikan jalan.
9. Tingkat prioritas perbaikan jalan yang dihasilkan sistem hanya didasarkan pada jenis kerusakan yang teridentifikasi melalui hasil klasifikasi ResNet-50 dan deteksi YOLOv12, tanpa mempertimbangkan kedalaman lubang, lebar retakan, maupun tingkat keparahan fisik kerusakan jalan secara aktual di lapangan.

E. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah untuk merancang dan membangun sistem pelaporan kerusakan jalan yang mampu mendeteksi dan mengklasifikasikan jenis kerusakan jalan dengan *Convolutional Neural Network* untuk analisis citra otomatis.

F. Manfaat Penelitian

Manfaat penelitian secara teoritis adalah sebagai berikut:

1. Memberikan kontribusi ilmiah terhadap pengembangan sistem pelaporan kerusakan jalan berbasis *Convolutional Neural Network*.
2. Menjadi referensi bagi penelitian selanjutnya dalam bidang penerapan *computer vision* dan *artificial intelligence* pada infrastruktur transportasi.



Manfaat penelitian secara praktis adalah sebagai berikut:

1. Menyediakan saluran pelaporan yang membantu instansi pemerintah dalam melakukan pemantauan kondisi jalan dan koordinasi penanganan kerusakan jalan.
2. Mempercepat proses identifikasi kerusakan, karena sistem dapat menganalisis citra secara otomatis tanpa perlu inspeksi visual manual yang memakan waktu.

BAB II

LANDASAN TEORI

A. Teori dan Penelitian Terdahulu

1. Landasan Teori

Landasan teori berisi kajian konsep dan teori yang relevan sebagai dasar penelitian. Bagian ini bertujuan memberikan pemahaman konseptual terkait objek dan metode yang digunakan, khususnya mengenai jalan rusak serta penerapan *Convolutional Neural Network* (CNN) dalam klasifikasi dan deteksi kerusakan jalan.

a. Jalan rusak

Berdasarkan Peraturan Pemerintah Nomor 34 Tahun 2006, jalan didefinisikan sebagai sarana transportasi darat yang mencakup seluruh unsur jalan, beserta bangunan pelengkap dan perlengkapannya yang digunakan untuk menunjang arus lalu lintas. Jalan tersebut dapat terletak di permukaan tanah, di atas atau di bawah permukaan tanah, di dalam air, maupun di atas permukaan air. Definisi ini tidak mencakup jalan kereta api, jalan lori, dan jalan kabel.

1) Jenis kerusakan jalan

Secara umum, kerusakan jalan dapat dibagi menjadi dua jenis utama, yaitu kerusakan struktural dan kerusakan fungsional. Kerusakan struktural berkaitan dengan kegagalan lapisan perkerasan atau kerusakan pada satu atau lebih komponen perkerasan sehingga jalan tidak lagi mampu menahan beban lalu lintas. Sementara itu, kerusakan fungsional adalah kerusakan yang mengganggu tingkat keamanan dan kenyamanan pengguna jalan, yang pada akhirnya dapat menyebabkan meningkatnya biaya operasi kendaraan.

2) Rincian kerusakan jalan

Berbagai jenis kerusakan pada perkerasan jalan dapat muncul sebagai akibat beban lalu lintas, kondisi tanah dasar, pengaruh

cuaca, maupun umur konstruksi. Beberapa bentuk kerusakan yang umum dijumpai antara lain lubang, retak melintang, retak memanjang, serta retak buaya. Masing-masing kerusakan tersebut memiliki karakteristik visual dan tingkat keparahan yang berbeda.



Melintang

Memanjang

Retak Buaya

Lubang

Gambar 2.1 Bentuk Kerusakan Jalan Secara Umum

Berikut adalah pembagian jenis kerusakan jalan sesuai dengan manual pemeliharaan jalan No: 03/MN/B/1983 yang dikeluarkan oleh Direktorat Jenderal Bina Marga.

a) Retak melintang (*transverse crack*)

Retak refleksi melintang adalah retak yang arah utamanya tegak lurus atau hampir tegak lurus terhadap sumbu jalan sehingga memotong arah lalu lintas, biasanya dari satu sisi lajur ke sisi lainnya atau hanya sebagian lebar lajur. Terbentuknya retak refleksi melintang dapat disebabkan oleh perubahan suhu yang menimbulkan pemuaian dan penyusutan berulang pada lapisan perkerasan di bawah.

b) Retak memanjang (*longitudinal crack*)

Retak refleksi memanjang adalah retak yang arah utamanya sejajar dengan sumbu jalan atau mengikuti arah perkerasan utama. Terbentuknya retak refleksi memanjang dapat disebabkan oleh pergerakan vertikal maupun horizontal pada lapisan di bawah akibat beban lalu lintas berulang, serta perubahan kadar air pada tanah dasar

ekspansif yang menimbulkan proses pengembangan dan penyusutan.

c) Retak kulit buaya (*alligator crack*)

Lebar celah ≥ 3 mm dengan pola saling terhubung membentuk kotak-kotak kecil menyerupai kulit buaya. Penyebabnya antara lain bahan perkerasan yang kurang baik, pelapukan permukaan, tanah dasar atau lapisan di bawahnya yang jenuh air, serta repetisi beban lalu lintas yang melampaui kapasitas perkerasan.

d) Lubang (*potholes*)

Kerusakan berbentuk mangkuk dengan ukuran bervariasi yang menampung air dan mempercepat kerusakan. Penyebabnya termasuk campuran permukaan yang buruk (kadar aspal rendah, agregat kotor, temperatur tidak sesuai), lapisan permukaan yang terlalu tipis, sistem drainase yang buruk, serta retak yang tidak segera diperbaiki.

3) Prioritas Perbaikan Kerusakan Jalan Berdasarkan Metode PCI

Dalam praktik pemeliharaan jalan, identifikasi jenis kerusakan perlu diikuti dengan penentuan prioritas perbaikan. Salah satu metode yang digunakan secara luas adalah *Pavement Condition Index* (PCI) yang dikembangkan oleh U.S. Army Corp of Engineer dan telah banyak diadaptasi dalam sistem manajemen perkerasan jalan di Indonesia (Marfenita dkk., 2025). Metode PCI menilai kondisi perkerasan berdasarkan jenis, tingkat keparahan (*severity*), dan luas kerusakan untuk menghasilkan skor kondisi jalan antara 0 hingga 100.

a) Lubang (*potholes*)

Dalam metode PCI, lubang merupakan indikator kerusakan yang paling kritis karena berdampak langsung terhadap keselamatan pengguna jalan. Kerusakan ini bersifat

progresif, mudah menampung air, dan dapat mempercepat degradasi lapisan perkerasan di sekitarnya.

Oleh karena itu, keberadaan lubang pada perkerasan jalan memiliki prioritas perbaikan sangat tinggi, karena dapat menurunkan skor PCI secara signifikan serta menimbulkan risiko kecelakaan secara langsung bagi pengguna jalan.

b) Retak kulit buaya (*alligator crack*)

Retak kulit buaya menunjukkan adanya kegagalan struktural pada lapisan perkerasan akibat beban lalu lintas berulang.

Pada tingkat keparahan rendah, retak masih halus dan belum terjadi pelepasan material. Pada tingkat sedang hingga tinggi, pola retak semakin jelas, terjadi pengelupasan (*spalling*), dan dapat disertai ambles.

Kerusakan ini memiliki prioritas perbaikan tinggi, karena penanganannya tidak cukup hanya pada lapisan permukaan, melainkan perlu melibatkan perbaikan struktur perkerasan.

c) Retak Memanjang

Retak memanjang dinilai berdasarkan lebar celah dan kondisi tepi retakan. Tingkat keparahan kerusakan diklasifikasikan mulai dari retak dengan celah sempit dan tepi yang masih utuh hingga retak dengan celah lebar yang disertai kerusakan di sekitar tepi retakan.

Kerusakan ini memiliki prioritas perbaikan sedang, karena apabila tidak segera ditangani, air dapat masuk ke lapisan perkerasan di bawahnya dan memicu kerusakan yang lebih berat seperti retak kulit buaya atau terbentuknya lubang.

d) **Retak Melintang**

Retak melintang umumnya disebabkan oleh faktor lingkungan seperti perubahan suhu dan siklus cuaca. Penilaian tingkat keparahan dilakukan dengan kriteria yang sama seperti retak memanjang.

Kerusakan ini memiliki prioritas perbaikan rendah hingga sedang, terutama pada tahap awal, namun tetap memerlukan tindakan pemeliharaan seperti pengisian celah (*crack sealing*) untuk mencegah perkembangan kerusakan.

b. **Convolutional Neural Network**

Convolutional Neural Network (CNN) adalah jenis jaringan saraf tiruan berlapis dalam (*deep learning*) yang memiliki *convolutional layer* untuk melakukan operasi konvolusi (Zhang dkk., 2021). CNN merupakan jaringan saraf tiruan *feed forward* yang dalam di mana jaringan saraf mempertahankan struktur hierarkis dengan mempelajari representasi fitur internal dan menggeneralisasi fitur dalam masalah gambar umum seperti pengenalan objek dan masalah penglihatan komputer lainnya (Raharjo, 2022).

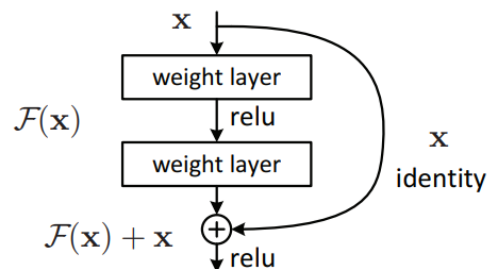
Perkembangan CNN mengalami lompatan besar sejak hadirnya model seperti LeNet-5 untuk klasifikasi digit tulisan tangan dan AlexNet, yang pada tahun 2012 secara signifikan meningkatkan akurasi kompetisi ImageNet dan memicu revolusi *deep learning* untuk *computer vision*. Setelah itu, berbagai inovasi muncul seperti *group convolution*, *deformable convolution*, dan *steerable convolution*, yang memperluas kemampuan CNN dalam menangani transformasi geometris dan meningkatkan efisiensi komputasi. Model-model baru seperti VGG, ResNet, MobileNet, dan GhostNet juga turut memperkaya desain arsitektur CNN dengan fokus pada kedalaman jaringan, *residual learning*, dan efisiensi komputasi.

c. ResNet-50

ResNet-50 adalah salah satu varian dari keluarga arsitektur *Residual Network* (ResNet) yang diperkenalkan oleh He et al. pada tahun 2016. Inovasi utama arsitektur ini terletak pada *residual connection* atau *skip connection* yang memungkinkan aliran informasi melewati satu atau beberapa lapisan secara langsung. Dengan mekanisme tersebut, ResNet-50 mampu membangun jaringan yang sangat dalam yaitu 50 lapisan (Berliani dkk., 2023), tanpa mengalami degradasi kinerja atau masalah *vanishing gradient* yang sering terjadi pada jaringan konvensional yang dalam.

1) *Residual Block*

Jantung dari ResNet-50 adalah blok residual. Berbeda dengan lapisan jaringan saraf tradisional yang mempelajari fungsi transformasi $H(x)$ secara langsung, blok residual memperkenalkan *skip connection* (koneksi pintas) yang memungkinkan input asli x untuk ditambahkan ke output dari lapisan-lapisan di dalamnya.

Gambar 2.2 *Skip Connection*

Sumber : (Jahani & Safaei, 2024)

Persamaan matematis untuk blok residual adalah:

$$y = F(x, \{W_i\}) + x \quad \dots\dots\dots (2.1)$$

Keterangan:

- x adalah input ke blok.
- $F(x, \{W_i\})$ adalah fungsi pemetaan residual yang dipelajari oleh dua atau tiga lapisan konvolusional di dalam blok.
- y adalah output akhir blok.

Skip connection (+ x) ini berfungsi sebagai jalur alternatif yang memudahkan aliran *gradien* saat proses *backpropagation*, memastikan informasi penting tidak hilang dan *gradien* tetap kuat bahkan di lapisan terdalam.

2) Penggunaan *Bottleneck Block*

ResNet-50 menggunakan desain yang disebut *bottleneck block* untuk efisiensi komputasi. Setiap blok *bottleneck* terdiri dari tiga lapisan konvolusional:

a) Konvolusi 1×1

Berfungsi untuk mengurangi dimensi (jumlah filter atau saluran) input, menciptakan "leher botol" yang efisien secara komputasi. Selain menekan jumlah parameter dan beban perhitungan, konvolusi ini juga membantu mempertahankan informasi penting sebelum diproses lebih lanjut oleh lapisan berikutnya.

b) Konvolusi 3×3

Lapisan konvolusi 3×3 berperan sebagai inti pemrosesan spasial, di mana fitur-fitur lokal pada citra diekstraksi secara lebih mendalam. Dengan bekerja pada dimensi yang telah diperkecil oleh konvolusi 1×1 sebelumnya, proses ini menjadi lebih efisien namun tetap mampu menangkap pola visual seperti tepi, tekstur, dan bentuk kerusakan.

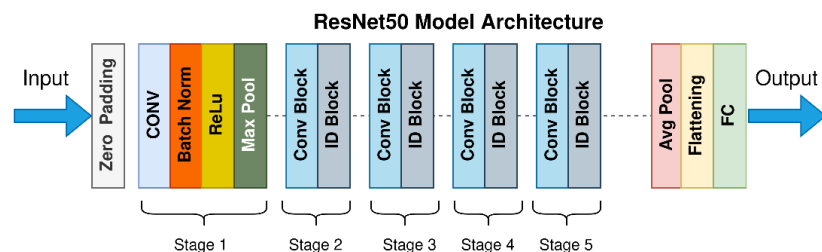
c) Konvolusi 1×1

Lapisan konvolusi 1×1 terakhir berfungsi untuk mengembalikan dimensi ke ukuran aslinya, sehingga hasil pemrosesan siap digabungkan dengan input awal x melalui mekanisme *skip connection*.

Desain ini memungkinkan ResNet-50 mencapai kedalaman yang signifikan dengan jumlah parameter yang relatif efisien dibandingkan dengan arsitektur lain pada masanya.

4) Arsitektur ResNet50

Arsitektur ResNet-50 secara spesifik terdiri dari 50 lapisan yang dapat dilatih dan disusun ke dalam beberapa tahap (*stages*) yang saling berurutan. Setiap tahap terdiri dari beberapa *residual block* dengan struktur *bottleneck block* (1×1 , 3×3 , 1×1) yang berfungsi mengekstraksi fitur secara bertahap dari tingkat rendah hingga tinggi. Pembagian ke dalam *stages* ini memungkinkan jaringan mempelajari representasi fitur yang semakin kompleks seiring bertambahnya kedalaman jaringan, sekaligus menjaga efisiensi komputasi dan stabilitas proses pelatihan.



Gambar 2.3 Arsitektur ResNet-50

Sumber : (Kanimozhi dkk., 2024)

Struktur umum ResNet-50 adalah sebagai berikut:

Tabel 2.1 Struktur Lapisan ResNet-50

Lapisan /Tahap	Ukuran Filter	Output Spasial	Jumlah Blok Bottleneck	Total Lapisan Konvolusi
Input	-	224 x 224 x 3	-	-
Awal	7x7, 64 filter	112 x 112	-	1
Stage 1	Max Pooling	56 x 56	-	-
Stage 2	3x3 (64, 64, 256)	56 x 56	3 blok	9
Stage 3	3x3 (128, 128, 512)	28 x 28	4 blok	12
Stage 4	3x3 (256, 256, 1024)	14 x 14	6 blok	18
Stage 5	3x3 (512, 512, 2048)	7 x 7	3 blok	9
Output	Avg Pooling + FC	4 kelas	-	1 + 1

Total 50 lapisan konvolusional dan satu lapisan *fully connected* dihitung dalam jumlah lapisan ResNet-50.

d. YOLOv12

YOLOv12 merupakan pengembangan terbaru dari keluarga *You Only Look Once* (YOLO) yang digunakan untuk deteksi objek secara *realtime*, dengan menggabungkan keunggulan arsitektur *Convolutional Neural Network* (CNN) dan Transformer melalui pendekatan *attention-centric*. Berbeda dari generasi sebelumnya, model ini menggunakan *area attention* untuk menyoroti bagian-bagian penting pada citra, memanfaatkan R-ELAN sebagai *backbone* guna meningkatkan aliran dan agregasi fitur, serta menerapkan *FlashAttention* untuk menekan beban akses memori saat perhitungan *attention*. Sejumlah penyesuaian tambahan, seperti penghapusan *positional encoding* eksplisit dan pengaturan ulang rasio ekspansi pada *feed-forward network*, membuat YOLOv12 mampu mempertahankan kecepatan pemrosesan tinggi sekaligus akurasi yang baik (Al dkk., 2025).

Dari sisi kemampuan, YOLOv12 tidak hanya berfungsi untuk deteksi objek, tetapi juga dapat digunakan untuk tugas visi komputer lain seperti segmentasi instan, klasifikasi citra, dan estimasi pose. Model ini disediakan dalam beberapa ukuran (N, S, M, L, X) sehingga fleksibel untuk diimplementasikan pada berbagai platform, mulai dari *edge device* hingga server berbasis *cloud*. Hasil pengujian pada dataset COCO menunjukkan bahwa YOLOv12 mampu memberikan nilai mAP yang lebih tinggi dengan kebutuhan komputasi (FLOPs) yang lebih rendah dibandingkan versi sebelumnya seperti YOLOv10 dan YOLOv11, sehingga menjadikannya salah satu model deteksi objek yang paling efisien, sekaligus tetap dilengkapi fitur keamanan untuk menjaga keandalan deteksi pada kondisi dunia nyata.

e. Django

Django merupakan kerangka kerja (*web framework*) tingkat tinggi berbasis bahasa pemrograman Python yang bersifat *open-source* dan bebas digunakan (Sabita dkk., 2022). Framework ini

17 dirancang untuk mempercepat pengembangan aplikasi web yang aman, terstruktur, dan mudah dipelihara. Django telah menyediakan berbagai komponen dasar seperti otentikasi pengguna, pengelolaan koneksi basis data, dan antarmuka admin (Saputra dkk., 2024), sehingga pengembang tidak perlu membangun semuanya dari awal dan dapat berfokus pada pengembangan fitur inti aplikasi. Django mendukung pengembangan yang cepat namun tetap menjaga kebersihan dan kepraktisan desain.

148 Secara arsitektur, Django memiliki pola *Model-View-Template* (MVT), yang merupakan variasi dari *Model-View-Controller* (MVC). Pada pola ini, *Model* berfungsi untuk mendefinisikan struktur data aplikasi dan mengelola interaksi dengan basis data melalui ORM bawaan, *View* bertugas memproses permintaan (*request*), berinteraksi dengan *Model* untuk mengambil atau memanipulasi data, serta menentukan data yang akan dikirim ke tampilan. Sedangkan *Template* mengatur antarmuka pengguna dan cara data disajikan kepada pengguna dengan tetap memisahkan logika presentasi dari logika bisnis. Django dibekali berbagai fitur terintegrasi seperti ORM yang mendukung berbagai jenis basis data (SQLite, PostgreSQL, MySQL), antarmuka admin otomatis yang siap pakai, sistem otentikasi dan pengelolaan perizinan pengguna, mekanisme pengaturan URL (*URL dispatcher*), serta fitur keamanan untuk melindungi aplikasi dari serangan umum seperti SQL injection, XSS, dan CSRF.

36 f. **Android**

81 Android merupakan sistem operasi sumber terbuka (*open source*) berbasis kernel Linux yang dirancang khusus untuk perangkat bergerak dengan antarmuka layar sentuh, seperti telepon pintar (*smartphone*) dan komputer tablet (Nurhidayati & Nur, 2021). Android dirancang untuk mempermudah pengembangan aplikasi yang portabel, skalabel, dan mampu berjalan di berbagai jenis perangkat dengan spesifikasi yang berbeda. Dengan menyediakan

API yang konsisten serta *tools* pengembangan seperti Android Studio, platform ini memungkinkan pengembang untuk fokus pada logika aplikasi tanpa harus menangani detail teknis rendah seperti manajemen memori dan komunikasi langsung dengan perangkat keras.

Android menyediakan platform standar untuk membangun aplikasi yang dapat memanfaatkan berbagai fitur perangkat, seperti kamera, GPS, sensor, jaringan, serta antarmuka pengguna yang interaktif. Dalam pengembangannya, Google merekomendasikan Kotlin sebagai bahasa utama untuk pengembangan aplikasi Android modern, menggantikan dominasi Java. Kotlin merupakan bahasa pemrograman yang berjalan di atas *Java Virtual Machine (JVM)*, bersifat statis (*statically typed*), *open-source*, dan dirancang agar *interoperable* dengan kode Java.

g. **Confusion Matrix**

Confusion matrix merupakan alat evaluasi untuk menilai kinerja model klasifikasi dengan membandingkan hasil prediksi dan data sebenarnya melalui komponen TP, TN, FP, dan FN (Sakinah & Riyadi, 2025). Berdasarkan **confusion matrix**, dapat dihitung metrik *accuracy*, *precision*, *recall*, dan *F1-Score* yang digunakan untuk menilai ketepatan, kemampuan deteksi, serta keseimbangan kinerja model, sehingga menjadi dasar penting dalam evaluasi performa klasifikasi. Rincian komponen **confusion matrix** dapat dilihat pada Tabel 2.2.

Tabel 2.2 Confusion Matrix

Class	Aktual Positif	Aktual Negatif
Prediksi Positif	TP (<i>True Positive</i>)	TN (<i>True Negative</i>)
Prediksi Negatif	FP (<i>False Postive</i>)	FN (<i>False Negative</i>)

True Positive (TP) merupakan kondisi ketika model berhasil memprediksi data positif dengan benar, sedangkan *True Negative*

(TN) terjadi ketika model dengan tepat memprediksi data negatif sebagai negatif. *False Positive* (FP) adalah keadaan ketika model memprediksi data sebagai positif padahal sebenarnya negatif, sementara *False Negative* (FN) merupakan kondisi ketika model memprediksi data sebagai negatif padahal sebenarnya positif. Keempat komponen ini menjadi dasar dalam penyusunan *confusion matrix* dan digunakan untuk mengevaluasi kinerja model klasifikasi.

Adapun rumus akurasi digunakan untuk mengukur tingkat ketepatan model dalam mengklasifikasikan data secara keseluruhan, yang dirumuskan sebagai berikut:

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \dots\dots\dots (2.2)$$

Selanjutnya, rumus presisi digunakan untuk mengetahui tingkat ketepatan prediksi positif yang dihasilkan oleh model, dengan rumus:

$$Precision = \frac{TP}{(TP + FP)} \dots\dots\dots (2.3)$$

Kemudian, rumus *recall* digunakan untuk mengukur kemampuan model dalam menemukan kembali seluruh data yang termasuk ke dalam kelas positif, yang dirumuskan sebagai berikut:

$$Recall = \frac{TP}{(TP + FN)} \dots\dots\dots (2.4)$$

Adapun rumus *F1-Score* digunakan untuk mengukur keseimbangan antara presisi dan *recall*, yang dirumuskan sebagai berikut:

$$F1\ Score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \dots\dots\dots (2.5)$$

h. *mean Average Precision* (mAP)

mean Average Precision (mAP) merupakan metrik evaluasi yang umum digunakan untuk menilai kinerja model *object detection* dengan mengukur tingkat ketepatan dan kelengkapan deteksi objek (Alif dkk., 2024). mAP50 mengacu pada nilai rata-rata *Average*

Precision (AP) yang dihitung pada ambang *Intersection over Union* (IoU) sebesar 0.5, yang menunjukkan seberapa baik model mendeteksi dan melokalisasi objek dengan toleransi kesesuaian yang relatif longgar. Sementara itu, mAP50–95 merupakan rata-rata AP pada beberapa ambang IoU mulai dari 0.5 hingga 0.95 dengan interval 0.05, sehingga memberikan penilaian yang lebih ketat dan komprehensif terhadap kemampuan model dalam mendeteksi objek secara presisi dan konsisten. Metrik mAP50–95 sering digunakan sebagai standar evaluasi utama karena mampu menggambarkan performa deteksi secara lebih menyeluruh dibandingkan mAP50 saja.

Adapun rumus mAP digunakan untuk mengukur kinerja model secara keseluruhan berdasarkan nilai *Average Precision* pada setiap kelas, yang dirumuskan sebagai berikut:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \dots\dots\dots (2.6)$$

Keterangan:

- 1) AP_i adalah nilai *Average Precision* pada kelas ke- i
- 2) N adalah jumlah kelas

Adapun rumus *Average Precision* (AP) digunakan untuk mengukur kinerja model pada satu kelas berdasarkan hubungan antara presisi dan *recall*, yang dirumuskan sebagai berikut:

$$AP = \int_0^1 p(r)dr \dots\dots\dots (2.7)$$

Keterangan:

- 1) AP adalah nilai *Average Precision*
- 2) $p(r)$ adalah presisi sebagai fungsi dari *recall*

Selanjutnya, perhitungan mAP menggunakan *threshold Intersection over Union (IoU)*, yang dirumuskan sebagai berikut:

$$IoU = \frac{Area(Box_{pred} \cap Box_{gt})}{Area(Box_{pred} \cup Box_{gt})} \dots\dots\dots (2.8)$$

Keterangan:

- 1) *IoU* adalah nilai *Intersection over Union*
- 2) Box_{pred} adalah *bounding box* hasil prediksi model
- 3) Box_{gt} adalah *bounding box* ground truth
- 4) $Area(Box_{pred} \cap Box_{gt})$ adalah luas area irisan antara *bounding box* prediksi dan *ground truth*
- 5) $Area(Box_{pred} \cup Box_{gt})$ adalah luas area gabungan antara *bounding box* prediksi dan *ground truth*

Dengan ketentuan:

- 1) $IoU \geq 0.5$ digunakan untuk perhitungan mAP50
- 2) $0.5 \leq IoU \leq 0.95$ digunakan untuk perhitungan mAP50–95

i. **Black Box Testing**

Black Box Testing adalah metode pengujian yang berfokus pada fungsionalitas sistem aplikasi tanpa melihat struktur internal atau kode program di dalamnya. Pada pendekatan ini, aplikasi diperlakukan seperti sebuah kotak hitam (*black box*), di mana penguji hanya memberikan input dan mengamati output yang dihasilkan untuk memastikan setiap fungsi berjalan sesuai dengan kebutuhan yang telah ditentukan. **Black Box Testing** digunakan untuk menemukan berbagai kesalahan pada sistem aplikasi, seperti fungsi yang tidak berjalan sesuai spesifikasi, *output* yang tidak sesuai harapan, hingga menu atau fitur yang hilang atau tidak dapat diakses (Nur Ichsanudin dkk., 2022). Dengan demikian, metode ini termasuk ke dalam pengujian fungsional (*functional testing*), karena yang dinilai adalah perilaku sistem dari sudut pandang pengguna, bukan cara kerja internalnya.

Dalam pengembangan perangkat lunak, pendekatan *Black Box* dinyatakan sebagai berikut:

$$f: I \rightarrow O \quad \dots\dots\dots (2.9)$$

Keterangan:

- 1) I adalah input dari pengguna
- 2) f adalah fungsi sistem yang tidak diobservasi struktur internalnya
- 3) O adalah output

Berdasarkan konsep tersebut, prinsip pengujian *Black Box* dapat dirumuskan sebagai:

$$Test = (I, O, f) \quad \dots\dots\dots (2.10)$$

Sistem dinyatakan berjalan dengan benar apabila output yang dihasilkan sesuai dengan output yang diharapkan, yang dirumuskan sebagai:

$$f(I) = O_{expected} \quad \dots\dots\dots (2.11)$$

2. Kajian Pustaka

Jalan raya adalah infrastruktur vital yang menunjang aktivitas ekonomi, sosial, serta mobilitas masyarakat, sehingga kualitasnya perlu dijaga. Namun, di banyak daerah, kondisi jalan mengalami penurunan akibat faktor lingkungan, cuaca, beban kendaraan berlebih, dan kurangnya pemeliharaan, yang menyebabkan kerusakan seperti lubang dan retakan yang dapat menurunkan keselamatan serta efektivitas transportasi (Gumelar dkk., 2023). Hal ini menunjukkan bahwa kualitas infrastruktur jalan di Indonesia masih perlu ditingkatkan, terutama karena proses pemantauan dan pelaporan kerusakan masih dilakukan secara manual.

Inspeksi visual manual dalam mendeteksi kerusakan jalan sering menimbulkan hasil yang tidak konsisten karena bergantung pada subjektivitas dan ketelitian petugas di lapangan. Kondisi ini menegaskan perlunya teknologi yang mampu melakukan identifikasi kerusakan secara otomatis dan objektif. Permasalahan ini membuka peluang pemanfaatan *computer vision* untuk menjawab kebutuhan tersebut. Dalam beberapa

tahun terakhir, berbagai studi telah mengusulkan model deteksi otomatis berbasis *computer vision* untuk meningkatkan efektivitas proses identifikasi kerusakan jalan. Akbar Pradana dkk., (2024) melakukan pendekatan CNN dan *hybrid* CNN-LSTM untuk mendeteksi keretakan jalan. Berdasarkan hasil pengujian, pendekatan CNN mendapatkan akurasi sebesar 92.76% sedangkan dengan algoritma *hybrid* CNN-LSTM mendapatkan Akurasi 97,5%. Penelitian lainnya oleh Gaho dkk., (2024) menggunakan arsitektur CNN berbasis Xception menunjukkan performa cukup baik dengan tingkat akurasi 90% dalam mendeteksi kerusakan jalan.

Pendekatan lain menggunakan CNN dengan arsitektur ResNet-50 seperti yang telah dilakukan oleh Ghofur & Ulum, (2025) menggunakan ukuran input *default* sebesar 224x224 piksel menunjukkan bahwa pendekatan ini dapat menghasilkan tingkat akurasi hingga 99% dalam klasifikasi jenis kerusakan permukaan jalan. Sementara itu, pendekatan dengan arsitektur ResNet-18 yang dilakukan oleh Saparudin dkk., (2025) juga menggunakan ukuran input *default* 224x224 piksel dan menunjukkan performa yang kompetitif dengan akurasi sebesar 92%. Temuan ini menegaskan potensi besar CNN dengan arsitektur ResNet sebagai alternatif yang lebih konsisten dan akurat.

Selain pendekatan berbasis CNN, metode lain dalam *computer vision* yang banyak digunakan untuk deteksi kerusakan jalan adalah YOLO (*You Only Look Once*). Algoritma ini dikenal karena kemampuannya melakukan deteksi objek secara *realtime* dengan kecepatan pemrosesan tinggi dan efisiensi komputasi yang baik, sehingga sangat sesuai untuk kebutuhan aplikasi lapangan berbasis citra maupun video. Penelitian oleh Surahmanto dkk., (2024) menerapkan deteksi jalan berlubang dengan data yang dikumpulkan sendiri sebanyak 476 menggunakan YOLOv5 diperoleh nilai *precision* sebesar 93% dan nilai *recall* 91%. Selanjutnya penelitian oleh Sakinah & Riyadi, (2025) menggunakan YOLOv8 untuk mengidentifikasi kerusakan jalan raya dengan dataset primer dan sekunder menghasilkan *F1-Score* tertinggi pada 88,7%, nilai *precision* sebesar

86.7% dan *recall* 90.8%. Sementara itu, penelitian oleh Takyudin dkk., (2025) melakukan penelitian serupa dengan pendekatan YOLOv11 menggunakan dataset yang diambil dari universal dataset roboflow sebanyak 2952 memperoleh hasil evaluasi dengan *precision* sebesar 75,1%, *recall* sebesar 76,6%, dan *F1-Score* sebesar 75,8%. Selain itu, penelitian terbaru oleh Deng & Song, (2025) memanfaatkan YOLOv12 dengan modifikasi pada mekanisme upsampling untuk meningkatkan performa deteksi. Menggunakan dataset UAV-PDD2023, penelitian tersebut berhasil mencapai hasil evaluasi yang cukup baik, yakni mAP50 sebesar 81,8% dan mAP50–95 sebesar 54,7%. Variasi pada hasil evaluasi tersebut menunjukkan bahwa kinerja YOLO tidak bersifat konstan dan sangat dipengaruhi oleh karakteristik serta kualitas dataset, maupun oleh konfigurasi arsitektur yang digunakan.

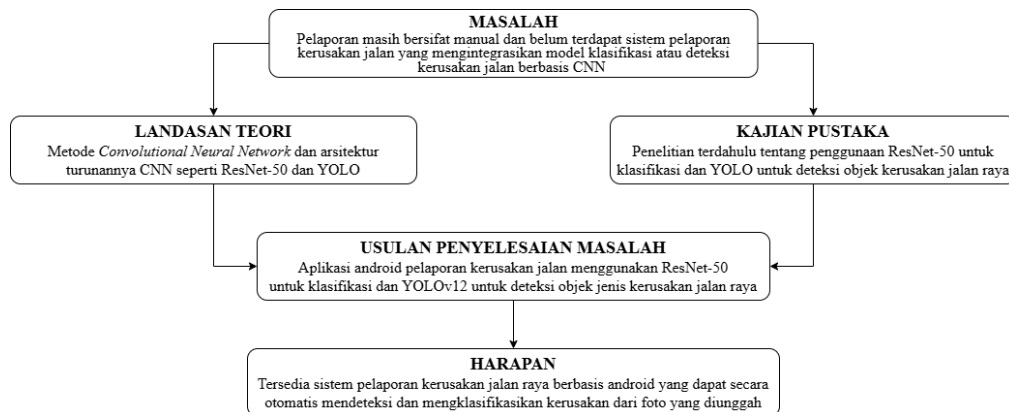
Meskipun hasil penelitian terkait pengembangan model deteksi berbasis CNN maupun YOLO menunjukkan kinerja yang menjanjikan, sebagian besar studi masih terfokus pada peningkatan akurasi model tanpa melanjutkan hingga tahap implementasi sistem dalam bentuk aplikasi yang dapat digunakan langsung oleh pengguna. Beberapa penelitian memang telah mengembangkan aplikasi pelaporan kerusakan jalan berbasis Android maupun web dengan pendekatan *Object Oriented Analysis and Design* (OOAD) seperti yang dilakukan oleh Yuniar dkk., (2024). Penelitian lain yang dilakukan oleh Wardani dkk., (2024) mengembangkan sistem informasi pemetaan dan pelaporan jalan rusak di Kota Tarakan menggunakan metode *waterfall* serta melakukan evaluasi kenyamanan pengguna menggunakan metode *System Usability Scale* (SUS) menghasilkan persentase sebesar 84,5%. Namun, aplikasi tersebut pada umumnya masih mengandalkan input manual berupa pelaporan atau unggahan foto dari masyarakat, sehingga belum mengintegrasikan kemampuan deteksi atau klasifikasi objek otomatis untuk mengidentifikasi kerusakan jalan secara langsung dari citra.

Di sisi lain, penelitian mengenai *deployment* model *machine learning* dan *deep learning* mulai berkembang. Framework Django banyak digunakan dalam penerapan model *object detection* berbasis web karena mempunyai kemampuan untuk memproses hasil deteksi secara *realtime*, mendukung *routing* yang fleksibel, serta kompatibel dengan berbagai pustaka *machine learning* dan *deep learning*. Seperti penelitian yang dilakukan oleh Zanevych, (2024) dengan membandingkan Django, Flask, Spring Boot untuk proyek *object detection* menunjukkan bahwa Django memiliki fitur yang komprehensif, termasuk ORM, modul autentikasi, dan panel admin, menjadikannya pilihan yang ideal untuk mengembangkan aplikasi kompleks yang memerlukan manajemen pengguna, penyimpanan data, serta penyusunan laporan yang mendetail. Penelitian yang dilakukan oleh Karthiga dkk., (2025) juga mengembangkan sistem keamanan CCTV secara *realtime* yang dilengkapi dengan deteksi objek untuk senjata api menggunakan Django sebagai *backend*. Selain itu, penelitian yang dilakukan oleh Chandra dkk., (2024) juga menggunakan Django sebagai *backend* untuk aplikasi android untuk menjalankan algoritma *Naïve Bayes*.

Dengan mengintegrasikan kemampuan ResNet-50 untuk mengklasifikasikan jenis kerusakan jalan dan YOLOv12 untuk mendeteksi lokasi kerusakan secara cepat dari citra yang diunggah, sistem mampu menghasilkan analisis visual yang lebih akurat. Framework Django berperan sebagai *backend* yang memproses hasil deteksi, mengelola seluruh data laporan, serta menyediakan API bagi aplikasi. Sementara itu, aplikasi Android menjadi antarmuka utama tempat pengguna mengambil gambar, mengirim laporan, dan menerima hasil deteksi maupun klasifikasi secara langsung. Kolaborasi teknologi tersebut membentuk solusi terpadu yang tidak hanya memfasilitasi pelaporan, tetapi juga mampu mengidentifikasi kerusakan secara otomatis, mempercepat proses verifikasi, dan menghadirkan informasi yang lebih objektif bagi pihak terkait.

B. Kerangka Berpikir

Kerangka berpikir pada Gambar 2.4 menggambarkan alur penelitian dari permasalahan hingga solusi yang diusulkan.



Gambar 2.4 Kerangka Berpikir

Kerangka berpikir pada penelitian ini diawali dari identifikasi masalah, yaitu belum tersedianya sistem pelaporan kerusakan jalan yang mengintegrasikan model klasifikasi atau deteksi kerusakan jalan berbasis CNN, khususnya arsitektur ResNet-50. Pelaporan kerusakan jalan masih banyak dilakukan secara manual sehingga berpotensi menimbulkan ketidakkonsistenan data, keterlambatan informasi, dan kurang efektif dalam mendukung pengambilan keputusan. Permasalahan ini kemudian dihubungkan dengan landasan teori mengenai metode CNN dan arsitektur turunannya seperti ResNet-50 yang memiliki kemampuan kuat dalam mengenali pola visual pada citra. Selain itu, kajian pustaka menunjukkan bahwa ResNet-50 telah dimanfaatkan dalam berbagai penelitian untuk tugas klasifikasi kerusakan jalan, sehingga memberikan dasar ilmiah bahwa arsitektur ini relevan untuk diterapkan dalam konteks penelitian ini.

Berdasarkan hubungan antara masalah, landasan teori, dan kajian pustaka tersebut, dirumuskan usulan penyelesaian berupa pemanfaatan model ResNet-50 untuk klasifikasi jenis kerusakan jalan raya. Model ini diintegrasikan ke dalam sistem pelaporan kerusakan jalan untuk mengolah citra yang diunggah dan mengklasifikasikannya ke dalam kategori kerusakan tertentu secara otomatis. Harapan dari kerangka berpikir ini adalah tersedianya sistem

25

pelaporan kerusakan jalan yang mampu mendeteksi dan mengklasifikasikan kerusakan dari foto yang diunggah secara lebih cepat, objektif, dan akurat, sehingga dapat mendukung proses pemantauan kondisi jalan dan perencanaan pemeliharaan infrastruktur secara lebih efektif.

BAB III

METODE PENELITIAN

A. Desain Penelitian

Desain penelitian ini disusun sebagai kerangka kerja yang sistematis untuk mengarahkan seluruh tahapan penelitian, mulai dari perumusan masalah hingga pencapaian tujuan. Perancangan desain penelitian bertujuan memastikan proses pengembangan, pengujian, dan evaluasi sistem pelaporan kerusakan jalan dapat dilakukan secara terstruktur dan sesuai dengan permasalahan yang dikaji.

1. Jenis Penelitian

Penelitian ini merupakan penelitian eksperimental dan pengembangan yang bertujuan untuk merancang dan membangun sistem pelaporan kerusakan jalan yang mampu mendeteksi dan mengklasifikasikan jenis kerusakan jalan dengan memanfaatkan arsitektur *Convolutional Neural Network*. Sebelum diintegrasikan ke dalam sistem, kedua model diuji pada beberapa variasi ukuran input citra untuk menentukan konfigurasi yang menghasilkan performa paling optimal. Model ResNet-50 dimanfaatkan untuk klasifikasi jenis kerusakan jalan, sedangkan YOLOv12 digunakan untuk mendeteksi dan melokalisasi objek kerusakan dalam citra secara otomatis, dengan platform Android sebagai media implementasi aplikasi. Sistem yang dibangun diharapkan mampu mendukung pelaporan kerusakan jalan secara otomatis dan terstandarisasi, serta membantu pengambil kebijakan dalam menentukan prioritas perbaikan jalan secara lebih efisien dan berbasis data.

2. Variabel Penelitian

Variabel penelitian merupakan unsur penting yang digunakan untuk menjelaskan faktor-faktor yang memengaruhi dan dipengaruhi dalam suatu penelitian. Variabel ini berfungsi sebagai dasar dalam proses pengamatan, pengukuran, serta analisis guna mencapai tujuan penelitian. Berdasarkan peran dan fungsinya, variabel penelitian dapat dibedakan menjadi beberapa jenis, yaitu variabel independen sebagai variabel yang

memengaruhi, variabel dependen sebagai variabel yang dipengaruhi, serta variabel kontrol sebagai variabel yang dijaga tetap konstan agar hasil penelitian tidak dipengaruhi oleh faktor luar.

a. Variabel Independen

Variabel independen merupakan variabel bebas yang memengaruhi atau menjadi penyebab perubahan pada variabel dependen. Dalam penelitian, variabel ini sengaja ditentukan atau dimanipulasi untuk mengetahui pengaruhnya terhadap hasil yang diamati. Variabel independen meliputi:

1) Jumlah *Epoch*

Variabel ini menunjukkan jumlah iterasi pelatihan model terhadap seluruh dataset citra kerusakan jalan. Semakin besar jumlah *epoch*, model memiliki kesempatan lebih banyak untuk mempelajari pola kerusakan, namun berpotensi menyebabkan *overfitting* apabila tidak dikontrol dengan baik.

2) Ukuran Input Citra

Variabel ini merujuk pada dimensi piksel citra masukan yang digunakan selama proses pelatihan dan pengujian model, yang memengaruhi tingkat detail fitur visual yang diekstraksi sehingga berdampak pada performa deteksi dan klasifikasi.

3) *Learning Rate*

Learning rate merupakan *hyperparameter* yang mengatur besar langkah pembaruan bobot model selama proses pelatihan. Nilai *learning rate* yang tepat diperlukan agar proses konvergensi berjalan stabil dan menghasilkan performa klasifikasi yang optimal.

4) *Batch Size*

Batch size menentukan jumlah citra kerusakan jalan yang diproses oleh model dalam satu kali iterasi pelatihan. Pengaturan *batch size* berpengaruh terhadap kecepatan pelatihan serta stabilitas proses pembelajaran model. *Batch size* yang terlalu

kecil dapat menyebabkan proses pelatihan menjadi kurang stabil, sedangkan *batch size* yang terlalu besar membutuhkan sumber daya komputasi yang lebih tinggi dan berpotensi mengurangi kemampuan generalisasi model.

b. Variabel Dependen

Variabel dependen merupakan variabel terikat yang dipengaruhi oleh variabel independen. Variabel ini menjadi hasil atau keluaran yang diamati dan diukur dalam penelitian untuk mengetahui dampak dari perlakuan atau perubahan pada variabel independen. Variabel dependen meliputi:

1) Jenis Kerusakan Jalan

Variabel ini merupakan keluaran sistem berupa hasil klasifikasi model terhadap citra yang diunggah, apakah tergolong dalam kategori kerusakan tertentu sesuai label klasifikasi atau tidak.

2) Performa Model

Variabel ini merupakan ukuran keluaran yang digunakan untuk mengevaluasi kemampuan model ResNet-50 dan YOLOv12 dalam melakukan klasifikasi dan deteksi kerusakan jalan terhadap data uji, yang diukur melalui nilai *precision, recall, F1-Score*, mAP50, dan mAP50-95.

c. Variabel Kontrol

Variabel kontrol merupakan variabel yang dijaga tetap selama penelitian agar tidak memengaruhi hubungan antara variabel independen dan dependen, sehingga hasil penelitian benar-benar disebabkan oleh variabel independen dan bukan oleh faktor lain.

Variabel kontrol meliputi:

1) Dataset

Dataset yang digunakan, baik berasal dari dokumentasi pribadi maupun sumber publik seperti RDD2022, Mendeley Data, serta Roboflow, dijaga konsistensinya selama proses

pelatihan dan pengujian. Pengendalian ini dilakukan dengan memastikan distribusi kelas, kualitas citra, serta skema pembagian data tetap seragam, sehingga variasi data tidak berubah secara signifikan. Dengan demikian, performa model yang dihasilkan benar-benar mencerminkan kemampuan model dalam mengenali kondisi citra kerusakan jalan yang ditemui pada situasi nyata.

2) Perangkat Keras

Spesifikasi perangkat keras, seperti penggunaan prosesor Intel® Core i7-7820HQ CPU @ 2.90 GHz, dikendalikan agar proses pelatihan dan pengujian tidak dipengaruhi perbedaan kemampuan komputasi yang dapat menimbulkan variasi kinerja model. Dengan demikian, hasil evaluasi lebih dapat dibandingkan secara objektif.

3. Metode Pengumpulan Data

Metode pengumpulan data pada penelitian ini dilakukan dengan memadukan data primer berupa dokumentasi lapangan dengan data sekunder dari beberapa platform penyedia dataset citra kerusakan jalan, antara lain Mendeley Data, Roboflow, dan GitHub. Ketiga platform tersebut merupakan sumber data yang cukup luas dan sering dimanfaatkan dalam penelitian terkait *computer vision*. Seluruh citra kerusakan jalan kemudian diklasifikasikan ke dalam empat kategori utama, yaitu retak melintang, retak memanjang, retak buaya, dan lubang. Proses pelabelan dan pengelompokan ini bertujuan agar algoritma dapat belajar membedakan pola kerusakan secara lebih akurat serta mendukung kinerja sistem dalam melakukan klasifikasi pada aplikasi yang dikembangkan.

B. Instrumen Penelitian

Instrumen penelitian yang digunakan berfungsi sebagai alat untuk mengumpulkan, mengolah, dan mengevaluasi data dalam pengembangan sistem pelaporan dan deteksi kerusakan jalan. Instrumen ini mencakup

perangkat keras, perangkat lunak, serta data pendukung yang digunakan selama proses penelitian.

1. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan dalam proses pengembangan dan implementasi sistem meliputi komponen berikut:

- Kapasitas penyimpanan sebesar 256GB SSD
- CPU menggunakan Intel® Core i7-7820HQ CPU @ 2.90 GHz
- RAM berkapasitas 32GB
- GPU menggunakan Intel® HD Graphics 630

2. Perangkat Lunak (*Software*)

Perangkat lunak pendukung yang digunakan dalam pengembangan dan implementasi sistem yaitu:

- Sistem operasi perangkat menggunakan Windows 10 Pro 22H2
- Bahasa pemrograman menggunakan Python dan Kotlin
- Tools* yang digunakan yaitu Visual Studio Code, Android Studio, Visual Paradigm.

3. Dataset

Penelitian ini memanfaatkan dataset gabungan yang terdiri dari data primer hasil dokumentasi lapangan serta data sekunder yang berasal dari beberapa platform publik. Total citra yang digunakan berjumlah 2.000 gambar kerusakan jalan. Sumber dataset publik diperoleh dari Mendeley Data, Roboflow, dan GitHub RDD 2022, yang masing-masing menyediakan kumpulan citra kerusakan jalan dalam berbagai jenis dan kondisi pencahayaan. Penggunaan data dari beberapa sumber bertujuan untuk menambah variasi citra dan meningkatkan kemampuan generalisasi model. Adapun referensi sumber dataset publik akan dicantumkan dalam Tabel 3.1 berikut:

Tabel 3.1 Sumber Dataset Sekunder

RDD2022	https://github.com/sekilab/RoadDamageDetector
Mendeley Data	https://data.mendeley.com/datasets/27c8pwsd6v/3
Roboflow	https://universe.roboflow.com/tugas-xogo7/jalanrusak-zofhk

4. Analisis Hasil

Pada penelitian ini dilakukan evaluasi menggunakan *Confusion Matrix* untuk menggambarkan performa model klasifikasi berdasarkan perbandingan antara hasil prediksi dengan label sebenarnya pada data uji. Melalui *Confusion Matrix* dapat diketahui jumlah prediksi benar maupun salah pada masing-masing kelas sehingga kinerja model dapat dianalisis secara lebih komprehensif. Selain itu, digunakan beberapa metrik evaluasi, yakni *precision*, *recall* serta *F1-Score*. Metrik-metrik tersebut digunakan untuk memastikan bahwa model tidak hanya memiliki tingkat ketepatan yang tinggi, tetapi juga mampu mendeteksi setiap kelas secara konsisten.

Evaluasi selanjutnya menggunakan metrik *mean Average Precision* (mAP), yaitu mAP50 dan mAP50–95. Nilai mAP50 menunjukkan rata-rata presisi model pada threshold $\text{IoU} \geq 0.5$ dengan penilaian yang lebih longgar, sedangkan mAP50–95 menghitung rata-rata presisi pada rentang IoU 0.5 hingga 0.95 dengan interval 0.05, sehingga memberikan evaluasi yang lebih ketat dan komprehensif terhadap kemampuan deteksi objek model.

Penggunaan keduanya dimaksudkan untuk memastikan model tidak hanya tepat pada kasus tumpang tindih rendah, tetapi juga mampu mempertahankan performa ketika kondisi deteksi lebih kompleks. Selain evaluasi model, pengujian aplikasi Android dilakukan menggunakan pendekatan *Black-box Testing*, yaitu metode pengujian perangkat lunak yang berfokus pada fungsi dan keluaran sistem tanpa melihat struktur internal atau kode sumber. Proses pengujian dilakukan dengan mensimulasikan alur penggunaan aplikasi dari sisi pengguna, seperti membuka kamera, memotret kerusakan jalan, menjalankan proses klasifikasi, hingga memperoleh hasil identifikasi. Melalui pendekatan ini, performa antarmuka, keakuratan fungsi, serta pengalaman pengguna dapat dievaluasi sehingga aplikasi tidak hanya bekerja secara teknis tetapi juga layak digunakan pada lingkungan operasional.

C. Jadwal Penelitian

Jadwal penelitian ini direncanakan dan dilaksanakan dalam kurun waktu 6 bulan. Selama periode tersebut, seluruh rangkaian kegiatan penelitian disusun secara sistematis, terstruktur, dan berurutan guna memastikan setiap tahapan penelitian dapat berjalan secara optimal dan saling mendukung. Penyusunan jadwal penelitian bertujuan untuk mengatur alokasi waktu secara efektif, meminimalkan kendala dalam pelaksanaan, serta menjaga ketercapaian target penelitian sesuai dengan tujuan yang telah ditetapkan.

Tabel 3.2 Jadwal Penelitian

No.	Kegiatan	Bulan Ke-																							
		1				2				3				4				5				6			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Studi Literatur																								
2	Pengumpulan Data																								
3	Desain Sistem																								
4	Pengembangan Aplikasi atau Implementasi																								
5	Uji Coba																								
6	Evaluasi dan Analisis Hasil																								
7	Penulisan Laporan																								

Rincian kegiatan penelitian disajikan dalam Tabel 3.2. Kegiatan penelitian meliputi studi literatur, pengumpulan data, desain sistem, pengembangan aplikasi atau implementasi, uji coba, evaluasi dan analisis hasil, serta penulisan laporan.

D. Objek Penelitian / Subjek Penelitian

Objek dan subjek penelitian ditetapkan untuk menentukan fokus kajian serta ruang lingkup penelitian. Keduanya menjadi dasar dalam proses analisis dan pengembangan sistem yang dilakukan, sehingga penelitian dapat berjalan terarah dan sesuai dengan tujuan yang telah ditetapkan.

1. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem bertujuan mengidentifikasi kebutuhan fungsional dan nonfungsional sebagai dasar perancangan dan pengembangan sistem agar sesuai dengan tujuan yang diharapkan.

a. Kebutuhan fungsional

- 1) Sistem mampu menampilkan hasil pengolahan citra secara efektif
- 2) Sistem dapat menjalankan proses algoritma secara otomatis ketika melakukan klasifikasi dan deteksi kerusakan jalan.
- 3) Sistem mampu menentukan dan menampilkan tingkat prioritas perbaikan jalan berdasarkan jenis kerusakan hasil klasifikasi.

b. Kebutuhan nonfungsional

Sistem harus memiliki kinerja yang optimal dengan waktu respons yang cepat serta akurasi deteksi yang tinggi terhadap objek kerusakan jalan. Selain itu, aplikasi diharapkan dapat berjalan secara stabil pada perangkat pengguna, mampu menangani proses pemuatan data dan tampilan hasil deteksi dengan baik, serta memberikan pengalaman penggunaan yang nyaman dan mudah dipahami oleh pengguna.

c. Kebutuhan pengguna

Aplikasi harus dirancang agar mudah digunakan, menampilkan hasil klasifikasi dan deteksi secara jelas, serta menyediakan antarmuka yang intuitif. Pengguna harus dapat memahami hasil identifikasi kerusakan jalan tanpa memerlukan pengetahuan teknis yang mendalam.

d. Kebutuhan data

Aplikasi harus menyediakan kemudahan bagi pengguna untuk memotret citra kerusakan jalan dan menerima hasil analisis berdasarkan model ResNet-50 sebagai algoritma klasifikasi serta YOLOv12 sebagai deteksi objek. Informasi keluaran dari proses tersebut harus ditampilkan secara jelas dan mudah diinterpretasikan atau dipahami oleh pengguna.

2. Objek Penelitian

Objek pada penelitian ini berupa aplikasi pelaporan dan pendeteksi kerusakan jalan berbasis Android yang dirancang untuk mengidentifikasi jenis kerusakan secara otomatis. Aplikasi ini memungkinkan pengguna

berinteraksi langsung dengan sistem, mulai dari proses pengambilan atau pemilihan citra hingga memperoleh keluaran berupa hasil klasifikasi, nilai akurasi serta tingkat prioritas perbaikan jalan. Sistem memanfaatkan model ResNet-50 sebagai algoritma klasifikasi serta YOLOv12 sebagai metode deteksi objek, sehingga hasil keluaran tidak hanya menunjukkan jenis kerusakan jalan tetapi juga lokasi objek kerusakan pada citra yang dianalisis. Dengan adanya integrasi antara aplikasi *mobile*, model kecerdasan buatan, dan *backend* sistem, aplikasi ini diharapkan mampu mendukung proses pelaporan kerusakan jalan yang lebih cepat, akurat, dan terstruktur.

3. Subjek Penelitian

Subjek dalam penelitian ini adalah pengguna aplikasi pelaporan dan deteksi kerusakan jalan berbasis Android yang berinteraksi secara langsung dengan sistem untuk mengidentifikasi kondisi permukaan jalan melalui citra yang diambil secara langsung maupun diunggah dari perangkat pengguna. Peran pengguna menjadi sangat penting karena melalui interaksi tersebut dapat dievaluasi tingkat efektivitas aplikasi dalam memenuhi kebutuhan identifikasi kerusakan jalan secara praktis, mulai dari aspek kemudahan penggunaan, kenyamanan antarmuka, hingga kejelasan informasi yang ditampilkan oleh sistem.

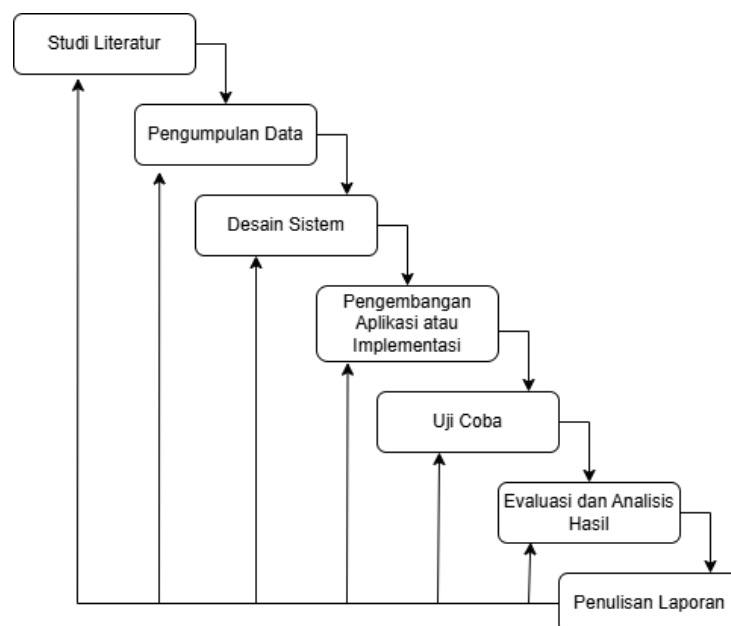
Aplikasi ini menghasilkan keluaran berupa klasifikasi jenis kerusakan jalan serta penandaan area kerusakan menggunakan model ResNet-50 dan YOLOv12, sehingga pengguna memperoleh umpan balik secara langsung terkait kondisi jalan yang dianalisis. Selain sebagai sarana pelaporan, data citra dan informasi kerusakan yang dihimpun dari aktivitas pengguna berpotensi dimanfaatkan sebagai tambahan dataset pada tahap pengembangan selanjutnya untuk meningkatkan performa dan kemampuan generalisasi model. Dengan demikian, penelitian ini tidak hanya berfokus pada evaluasi teknis algoritma, tetapi juga pada tingkat penerimaan pengguna dan kemanfaatan aplikasi dalam konteks

operasional di lapangan, serta membuka peluang pengembangan sistem berbasis data yang berkelanjutan.

E. Prosedur Penelitian

Pada penelitian ini, tahapan pengerjaan dilakukan secara bertahap dan terstruktur mulai dari pengumpulan teori hingga penyusunan laporan dengan mengacu pada metode *Waterfall*. Metode *Waterfall* merupakan metode pengembangan perangkat lunak yang mengikuti urutan tahapan yang jelas, berurutan, serta bersifat linier dan sistematis, di mana setiap tahap harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya. Pendekatan ini dipilih karena mampu memberikan alur kerja yang terencana dengan baik, memudahkan proses dokumentasi, serta meminimalkan kesalahan pada tahap implementasi.

Melalui penerapan metode *Waterfall*, proses penelitian diawali dengan tahap analisis kebutuhan dan studi literatur, kemudian dilanjutkan dengan perancangan sistem, implementasi model dan aplikasi, pengujian, serta evaluasi hasil. Tahapan akhir berupa penyusunan laporan penelitian yang dilakukan setelah seluruh proses pengembangan dan pengujian sistem selesai. Alur kegiatan penelitian ini secara keseluruhan ditunjukkan pada Gambar 3.1.



Gambar 3.1 Alur Kegiatan Penelitian

Secara lebih rinci, tahap studi literatur berfungsi untuk memperoleh landasan teoritis dan referensi yang relevan sebagai dasar pengembangan sistem. Selanjutnya, pengumpulan data dilakukan untuk menyediakan dataset yang sesuai dengan kebutuhan pelatihan dan pengujian model. Tahap perancangan sistem mencakup perancangan arsitektur aplikasi dan alur kerja sistem, yang kemudian direalisasikan pada tahap implementasi. Sistem yang telah dibangun selanjutnya diuji dan dievaluasi untuk menilai kinerja, akurasi, serta keandalan sistem, sebelum hasil akhirnya disajikan dalam bentuk laporan penelitian.

F. Teknik Analisis Data

Teknik analisis data dalam penelitian ini dilakukan untuk menjelaskan tahapan perancangan dan pengembangan sistem secara terstruktur. Pada bagian ini dipaparkan desain sistem yang digunakan sebagai acuan dalam membangun aplikasi pelaporan dan deteksi kerusakan jalan, mulai dari gambaran umum arsitektur sistem hingga interaksi antara pengguna dan sistem.

1. Desain Sistem

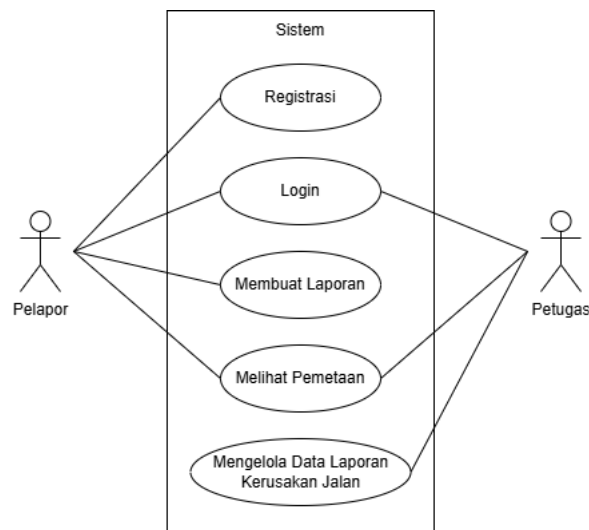
Desain sistem merupakan tahap awal yang bertujuan untuk menggambarkan struktur, komponen, serta alur kerja sistem sebelum tahap implementasi dilakukan. Perancangan ini disusun agar sistem yang dibangun sesuai dengan kebutuhan pengguna dan tujuan penelitian, sekaligus menjadi acuan dalam proses pengembangan, integrasi, dan pengujian sehingga sistem dapat berjalan secara efektif dan terstruktur.

a. Arsitektur Sistem (*System Architecture*)

Arsitektur sistem menjelaskan susunan komponen utama sistem serta hubungan antar komponen tersebut dalam mendukung proses pelaporan dan deteksi kerusakan jalan. Arsitektur ini menjadi dasar dalam memahami aliran data dan proses yang terjadi di dalam sistem. Dengan adanya arsitektur yang terstruktur, pengembangan, integrasi, serta pemeliharaan sistem dapat dilakukan secara lebih sistematis dan efisien.

1) Diagram *Use Case*

Diagram *Use Case* digunakan untuk menggambarkan interaksi antara pengguna dan sistem serta fungsi-fungsi yang dapat diakses oleh masing-masing aktor. Diagram ini memberikan gambaran umum mengenai kebutuhan fungsional sistem dan peran setiap pengguna dalam menjalankan sistem.



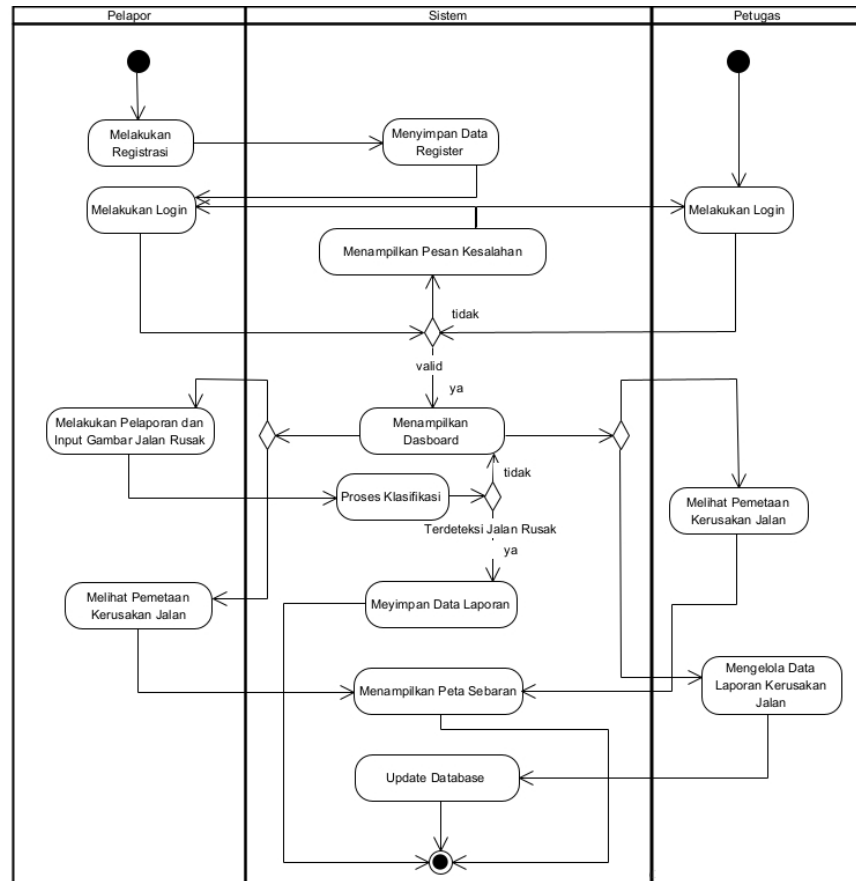
Gambar 3.2 Diagram *Use Case*

Gambar tersebut menunjukkan *Use Case* diagram aplikasi pelaporan dan deteksi kerusakan jalan yang melibatkan dua aktor, yaitu Pelapor dan Petugas. Pelapor berperan dalam melakukan registrasi, login, membuat laporan, dan melihat pemetaan kerusakan jalan, sedangkan Petugas memiliki akses untuk login, melihat pemetaan, serta mengelola data laporan. Diagram ini menegaskan pembagian peran dan hak akses masing-masing aktor dalam sistem.

2) *Activity Diagram*

Activity Diagram digunakan untuk menggambarkan alur aktivitas sistem secara rinci, mulai dari proses awal hingga akhir, serta menunjukkan urutan kegiatan yang terjadi dalam sistem. Diagram ini juga memperlihatkan interaksi antara pengguna dan sistem, termasuk keputusan, percabangan alur, dan proses yang

berjalan secara paralel, sehingga memudahkan pemahaman terhadap cara kerja sistem secara keseluruhan.



Gambar 3.3 Activity Diagram

Gambar 3.3 menunjukkan Activity Diagram yang menggambarkan alur aktivitas sistem pelaporan dan deteksi kerusakan jalan secara menyeluruh dengan melibatkan tiga aktor utama, yaitu Pelapor, Sistem, dan Petugas, yang masing-masing dipisahkan dalam swimlane untuk memperjelas peran dan tanggung jawab setiap pihak. Pemisahan ini bertujuan agar alur proses dapat dipahami dengan jelas, mulai dari interaksi pengguna hingga pemrosesan dan pengelolaan data oleh sistem.

Proses diawali dari sisi Pelapor, yang terlebih dahulu melakukan registrasi untuk membuat akun pada sistem. Data registrasi yang dimasukkan akan diproses dan disimpan oleh

sistem sebagai basis identitas pengguna. Setelah registrasi berhasil, Pelapor melakukan login ke dalam sistem. Pada tahap ini, sistem melakukan validasi terhadap data login yang dimasukkan. Apabila data tidak valid, sistem akan menampilkan pesan kesalahan sehingga pengguna harus mengulangi proses login. Jika data valid, sistem akan mengarahkan pengguna ke halaman *dashboard* sebagai pusat akses fitur sistem. Proses login yang sama juga dilakukan oleh Petugas, dengan hak akses yang disesuaikan dengan perannya.

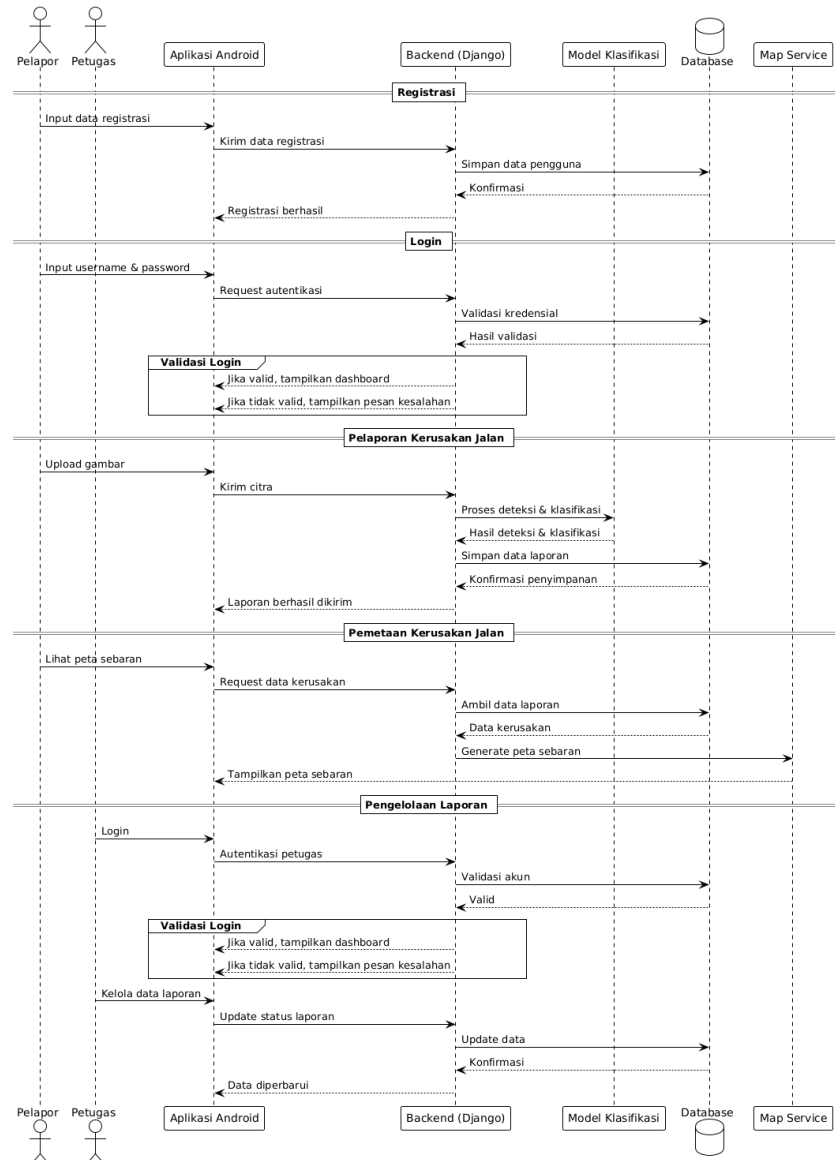
Setelah berhasil mengakses *dashboard*, Pelapor dapat melakukan pelaporan kerusakan jalan dengan mengunggah citra jalan rusak beserta informasi pendukung lainnya. Data yang dikirimkan kemudian diproses oleh sistem melalui tahap klasifikasi citra untuk mendeteksi keberadaan dan jenis kerusakan jalan. Jika hasil klasifikasi menunjukkan adanya kerusakan jalan, maka sistem akan menyimpan data laporan ke dalam basis data dan menampilkan hasilnya dalam bentuk peta sebaran kerusakan jalan yang dapat diakses oleh Pelapor sebagai umpan balik atas laporan yang telah diajukan.

Sementara itu, Petugas yang telah login juga dapat mengakses *dashboard* untuk melihat pemetaan kerusakan jalan secara keseluruhan serta mengelola data laporan kerusakan jalan, seperti melakukan verifikasi, validasi, dan pembaruan status laporan. Setiap aktivitas pengelolaan data akan memicu proses pembaruan basis data sehingga informasi yang ditampilkan pada sistem selalu bersifat terkini.

Secara keseluruhan, *Activity Diagram* ini menunjukkan alur kerja sistem yang terintegrasi antara Pelapor, Sistem, dan Petugas dalam mendukung proses pelaporan, klasifikasi, serta pengelolaan data kerusakan jalan, sehingga sistem dapat berjalan secara efektif dan sistematis.

3) Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan urutan interaksi antar aktor dan objek dalam sistem berdasarkan waktu, sehingga memperjelas alur komunikasi dan proses yang terjadi dari awal hingga akhir suatu skenario.



Gambar 3.4 Sequence Diagram

Sequence diagram pada Gambar 3.4 menggambarkan alur interaksi pada sistem yang melibatkan aktor Pelapor dan Petugas, serta komponen utama sistem yaitu Aplikasi Android, Backend

berbasis Django, Model Klasifikasi, Database, dan *Map Service*. Proses diawali dari tahap registrasi dan login, di mana pelapor memasukkan data akun melalui aplikasi. Data tersebut dikirim ke *backend* untuk disimpan atau divalidasi melalui database. Jika proses autentikasi berhasil, sistem akan menampilkan *dashboard* sesuai peran pengguna, sedangkan jika gagal, aplikasi menampilkan pesan kesalahan.

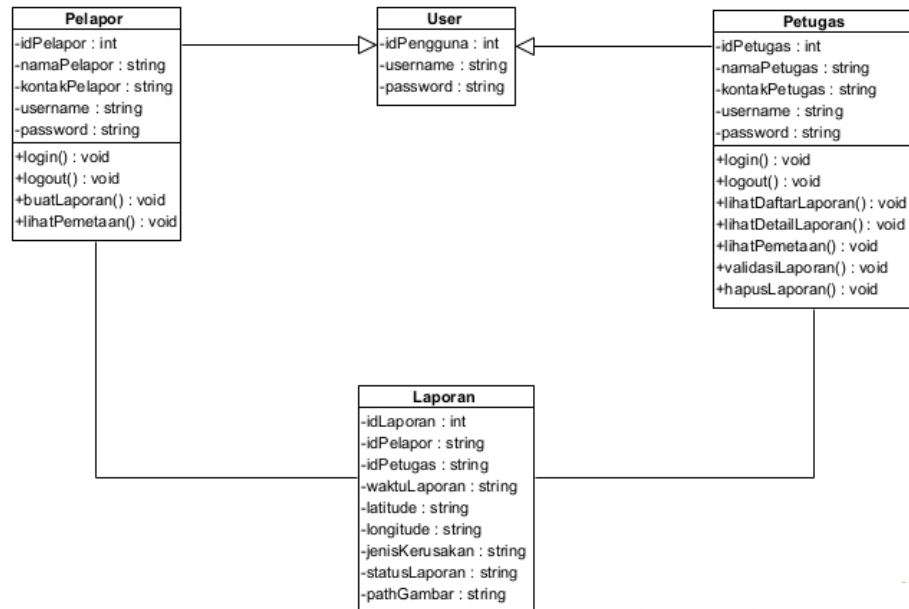
Pada tahap pelaporan kerusakan jalan, pelapor mengunggah gambar kondisi jalan beserta data pendukung melalui aplikasi Android. Data ini diteruskan ke *backend*, kemudian diproses oleh model YOLOv12 dan ResNet-50 untuk melakukan deteksi lokasi serta klasifikasi jenis kerusakan jalan. Hasil pemrosesan dikembalikan ke *backend* dan disimpan ke dalam database sebagai data laporan. Setelah proses penyimpanan berhasil, *backend* mengirimkan respons ke aplikasi bahwa laporan telah berhasil dikirim oleh pengguna.

Selanjutnya, *sequence diagram* juga menunjukkan proses pemetaan dan pengelolaan laporan. Pelapor dapat melihat peta sebaran kerusakan jalan dengan meminta data ke *backend*, yang kemudian mengambil informasi laporan dari database dan mengolahnya melalui *Map Service* untuk ditampilkan secara visual pada aplikasi. Di sisi lain, petugas dapat melakukan login dan mengelola data laporan, seperti memperbarui status laporan. Setiap perubahan yang dilakukan petugas akan disimpan kembali ke database dan dikonfirmasi ke aplikasi, sehingga sistem mendukung pemantauan dan pengelolaan kerusakan jalan secara terintegrasi dan terstruktur.

b. Desain Database

Class Diagram digunakan untuk menggambarkan struktur statis sistem yang meliputi kelas-kelas, atribut, metode, serta hubungan antar kelas yang membentuk aplikasi. Diagram ini memberikan

gambaran bagaimana data dan fungsi diorganisasikan dalam sistem pelaporan dan deteksi kerusakan jalan.



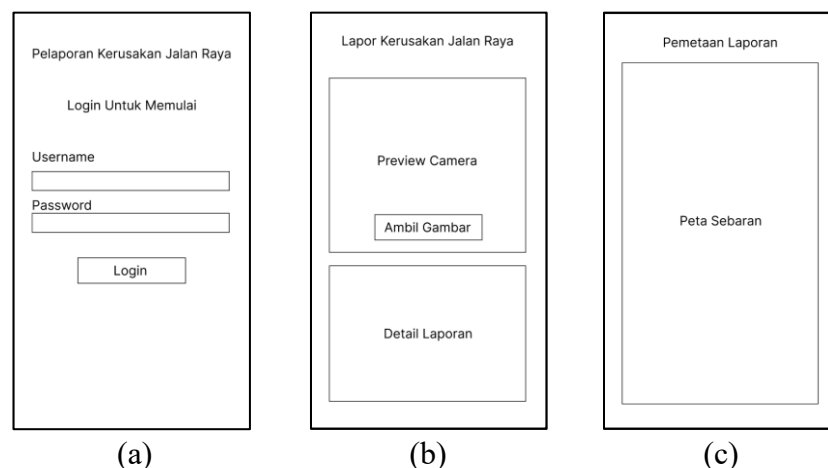
Gambar 3.5 Class Diagram

Class diagram pada Gambar 3.5 tersebut menggambarkan struktur kelas utama yang digunakan dalam sistem aplikasi pelaporan dan deteksi kerusakan jalan. Terdapat kelas User sebagai kelas induk (*superclass*) yang menyimpan atribut umum berupa *idPegguna*, *username*, dan *password*. Kelas Pelapor dan Petugas merupakan turunan dari kelas User yang mewarisi atribut dan operasi dasar, khususnya fungsi *login()* dan *logout()*. Kelas Pelapor memiliki atribut tambahan seperti *idPelapor*, *namaPelapor*, dan *kontakPelapor*, serta metode *buatLaporan()* dan *lihatPemetaan()* yang merepresentasikan aktivitas pengguna dalam membuat laporan kerusakan dan melihat sebaran lokasi jalan rusak. Sementara itu, kelas Petugas memiliki atribut *idPetugas*, *namaPetugas*, dan *kontakPetugas*, serta metode tambahan seperti *lihatDaftarLaporan()*, *lihatDetailLaporan()*, *validasiLaporan()*, *hapusLaporan()*, dan *lihatPemetaan()* yang mencerminkan peran petugas dalam mengelola dan memverifikasi laporan kerusakan jalan.

Selain itu, *class diagram* juga menampilkan kelas Laporan yang berfungsi sebagai representasi data laporan kerusakan jalan yang disimpan dalam sistem. Kelas Laporan memiliki atribut seperti *idLaporan*, *waktuLaporan*, *latitude*, *longitude*, *jenisKerusakan*, *statusLaporan*, dan *pathGambar*, yang menggambarkan informasi detail terkait laporan kerusakan jalan. Relasi asosiasi antara kelas Pelapor dan Laporan menunjukkan bahwa satu pelapor dapat membuat lebih dari satu laporan, sedangkan relasi antara kelas Petugas dan Laporan menunjukkan bahwa satu petugas dapat menangani atau memvalidasi banyak laporan. Struktur class diagram ini menunjukkan pembagian peran yang jelas antara pengguna dan petugas, serta mendukung pengelolaan data laporan kerusakan jalan secara terstruktur dan sistematis sesuai dengan kebutuhan aplikasi.

c. Desain Antarmuka Pengguna (UI/UX Design)

Desain antarmuka pengguna (UI/UX) dirancang agar aplikasi mudah digunakan, intuitif, dan nyaman bagi pengguna. Perancangan difokuskan pada kejelasan tampilan dan kemudahan navigasi sehingga proses pelaporan dan pemantauan kerusakan jalan dapat dilakukan secara efektif.



Gambar 3.6 Desain Antarmuka

Tampilan antarmuka pada Gambar 3.6(a) merupakan halaman login yang berfungsi sebagai gerbang awal akses sistem pelaporan

kerusakan jalan raya. Halaman ini menyediakan *field username* dan *password* untuk proses autentikasi pengguna. Setelah tombol Login ditekan, sistem memvalidasi data dengan basis data, kemudian mengarahkan pengguna ke halaman utama sesuai perannya jika berhasil, atau menampilkan pesan kesalahan jika data tidak valid.

Gambar 3.6(b) menampilkan halaman pelaporan kerusakan jalan yang menyediakan fitur *preview camera* untuk mengambil gambar kondisi jalan. Citra yang diunggah akan diproses menggunakan model deteksi dan klasifikasi untuk mengidentifikasi jenis kerusakan. Hasil analisis ditampilkan dalam bentuk visualisasi dan label prediksi, disertai detail laporan seperti lokasi dan waktu unggah sebagai pendukung sebelum laporan dikirim.

Gambar 3.6(c) menunjukkan halaman peta sebaran kerusakan jalan yang menyajikan lokasi jalan rusak secara visual melalui peta interaktif. *Marker* pada peta merepresentasikan titik kerusakan, sementara bagian detail sebaran menampilkan informasi rinci dari lokasi yang dipilih sehingga pengguna dapat memahami kondisi jalan secara lebih jelas dan terstruktur.

2. Simulasi Proses Penyelesaian Masalah

Pada bagian ini disajikan simulasi implementasi model ResNet-50 dalam mendeteksi kerusakan jalan untuk menjelaskan tahapan pemrosesan data dan mekanisme kerja model dalam mengklasifikasikan kondisi permukaan jalan berdasarkan citra digital. Simulasi ini menggambarkan alur pengolahan citra secara sistematis, mulai dari tahap awal hingga dihasilkan prediksi jenis kerusakan jalan.

a. Data Input

Pada tahap data input, pengguna memasukkan citra jalan berformat RGB dengan ukuran 224×224 piksel dan 3 channel warna (Red, Green, Blue) seperti contoh pada Gambar 3.7.



Gambar 3.7 Contoh Data Input

Citra ini kemudian diubah menjadi tensor agar dapat dihitung oleh jaringan CNN. Setiap elemen tensor mewakili intensitas warna pada piksel tertentu, yang menjadi dasar bagi proses ekstraksi fitur oleh jaringan ResNet-50. Tensor ini menjadi masukan awal bagi seluruh operasi konvolusi, *pooling*, dan blok residual dalam model.

b. **Proses Pengolahan Data**

Proses pengolahan data menggunakan metode ResNet-50 untuk mengekstraksi fitur citra dan menghasilkan klasifikasi melalui *resize*, normalisasi, tahapan konvolusi, *bottleneck block*, *Global Average Pooling*, dan Softmax. Proses perhitungan ResNet-50 dijelaskan sebagai berikut:

1) *Resize*

Dalam implementasi ResNet-50, citra umumnya diubah ukurannya sesuai standar masukan *pretrained model* menjadi:

$$224 \times 224 \times 3(RGB)$$

Namun, ukuran citra dapat disesuaikan berdasarkan hasil pengujian untuk memperoleh performa optimal.

2) Normalisasi

Mengubah rentang piksel menjadi 0 – 1.

Contoh satu piksel pada citra:

$$x_{(i,j)} = [R, G, B] = [120, 115, 110]$$

Sehingga didapatkan setelah normalisasi:

$$x' = \frac{x}{255} \Rightarrow [0.47, 0.45, 0.43]$$

3) Operasi Konvolusi Awal

Lapisan konvolusi awal pada ResNet-50 menggunakan kernel 7×7 dengan 64 filter, stride 2, dan padding untuk mengekstraksi fitur dasar dari citra input.

Contoh citra dengan single channel berukuran 4×4 :

$$X = \begin{bmatrix} 1 & 2 & 0 & 1 \\ 0 & 1 & 3 & 2 \\ 1 & 0 & 2 & 1 \\ 2 & 1 & 0 & 0 \end{bmatrix}$$

Dengan contoh kernel 3×3 :

$$K = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

Dengan operasi *same convolution* menambahkan 1 padding diluar citra awal maka perhitungan titik (0,0):

$$(1 \times 0) + (0 \times 0) + ((-1) \times 0) + (0 \times 0) + (1 \times 1) + (0 \times 2) + ((-1) \times 0) + (0 \times 0) + (1 \times 1) = 2$$

Dengan menerapkan operasi *same convolution* pada seluruh piksel citra, diperoleh matriks hasil konvolusi yang memiliki ukuran 4×4 . Adapun hasil konvolusi secara keseluruhan ditunjukkan sebagai berikut:

$$Y = \begin{bmatrix} 2 & 5 & 1 & -2 \\ -2 & 3 & 5 & 0 \\ 1 & -5 & 0 & 4 \\ 2 & 0 & -1 & 2 \end{bmatrix}$$

4) Batch Normalization (BN)

Untuk mempermudah perhitungan diasumsikan *mean* = 0 dan *variance* = 1. Sehingga BN tidak mengubah nilai:

$$BN(Y) = Y$$

Hasil BN:

$$BN(Y) = \begin{bmatrix} 2 & 5 & 1 & -2 \\ -2 & 3 & 5 & 0 \\ 1 & -5 & 0 & 4 \\ 2 & 0 & -1 & 2 \end{bmatrix}$$

5) Fungsi Aktivasi ReLU

Aplikasi aktivasi ReLU pada $BN(Y)$ merubah nilai negatif menjadi 0. Sehingga tekstur gelap pada lubang menjadi fitur dominan.

Hasil ReLU:

$$\text{ReLU}(BN(Y)) = \begin{bmatrix} 2 & 5 & 1 & 0 \\ 0 & 3 & 5 & 0 \\ 1 & 0 & 0 & 4 \\ 2 & 0 & 0 & 2 \end{bmatrix}$$

6) Max Pooling

Max pooling menggunakan kernel 2×2 , stride 2, dan padding valid untuk mengambil nilai maksimum pada setiap area lokal tanpa tumpang tindih. Proses ini mereduksi ukuran *feature map* secara signifikan, mempertahankan fitur dominan, serta meningkatkan efisiensi komputasi pada tahap selanjutnya.

Proses Pooling:

a) Region 1:

$$\begin{bmatrix} 2 & 5 \\ 0 & 3 \end{bmatrix} \Rightarrow \max = 5$$

b) Region 2:

$$\begin{bmatrix} 1 & 0 \\ 5 & 0 \end{bmatrix} \Rightarrow \max = 5$$

c) Region 3:

$$\begin{bmatrix} 1 & 0 \\ 2 & 0 \end{bmatrix} \Rightarrow \max = 2$$

d) Region 4:

$$\begin{bmatrix} 0 & 4 \\ 0 & 2 \end{bmatrix} \Rightarrow \max = 4$$

Maka didapatkan hasil Max Pooling:

$$P = \begin{bmatrix} 5 & 5 \\ 2 & 4 \end{bmatrix}$$

7) Bottleneck Block

Struktur *bottleneck* terdiri dari rangkaian konvolusi $1 \times 1 \rightarrow 3 \times 3 \rightarrow 1 \times 1$ yang dirancang untuk mengatur dimensi *channel* sekaligus mengekstraksi fitur spasial secara efektif.

a) Conv 1×1 (Reduksi *Channel*)

Karena simulasi *single channel*, konvolusi 1×1 dengan bobot = 1 tidak mengubah nilai:

$$F_1(P) = \begin{bmatrix} 5 & 5 \\ 2 & 4 \end{bmatrix}$$

b) Conv 3×3 (Ekstraksi Fitur)

Kernel 3×3 :

$$K = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

Karena ukuran input 2×2 , digunakan operasi *same convolution* dengan menambahkan 1 padding diluar citra.

Perhitungan Titik (0,0):

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 5 & 5 \\ 0 & 2 & 4 \end{bmatrix}$$

$$(1 \times 0) + (0 \times 0) + (-1 \times 0) + (0 \times 0) + (1 \times 5) + (0 \times 5) + (-1 \times 0) + (0 \times 2) + (1 \times 4) = 9$$

Hasil Konvolusi 3×3 :

$$F_2 = \begin{bmatrix} 9 & 7 \\ 3 & 7 \end{bmatrix}$$

Kemudian aktivasi ReLU. karena semua positif, maka tidak mengubah citra:

$$ReLU(F_2) = \begin{bmatrix} 9 & 7 \\ 3 & 7 \end{bmatrix}$$

c) Conv 1×1 (Ekspansi *Channel*)

Dengan bobot 1, maka didapatkan:

$$F_3(P) = \begin{bmatrix} 9 & 7 \\ 3 & 7 \end{bmatrix}$$

8) *Residual (Shortcut) Connection*

Karena *shortcut identity* tersedia atau dimensi sama, maka hasil blok residual:

$$R = \begin{bmatrix} 5 & 5 \\ 2 & 4 \end{bmatrix} + \begin{bmatrix} 9 & 7 \\ 3 & 7 \end{bmatrix} = \begin{bmatrix} 14 & 12 \\ 5 & 11 \end{bmatrix}$$

Kemudian dilakukan aktivasi ReLU, karena semua positif maka tidak mengubah citra:

$$\text{ReLU}(R) = \begin{bmatrix} 14 & 12 \\ 5 & 11 \end{bmatrix}$$

9) *Global Average Pooling (GAP)*

Global Average Pooling (GAP) digunakan untuk mereduksi *feature map* hasil ekstraksi fitur menjadi satu nilai representatif dengan cara menghitung rata-rata seluruh elemen pada setiap channel, sehingga diperoleh nilai sebagai berikut.

$$s = \frac{14 + 12 + 5 + 11}{4} = \frac{42}{4} = 10.5$$

10) *Fully Connected (FC) / Dense Layer*

Lapisan *Fully Connected* (FC) digunakan untuk mengonversi vektor fitur hasil *Global Average Pooling* menjadi empat kelas kerusakan jalan, yaitu lubang, retak buaya, retak memanjang, dan retak melintang. Proses ini dilakukan melalui perkalian bobot dan penambahan bias untuk menghasilkan nilai logit sebagai representasi awal tingkat keyakinan model sebelum diproses menggunakan fungsi Softmax.

Misal nilai bobot:

$$W = [0.6, 0.2, -0.1, 0.1]$$

Maka, perhitungan nilai logits:

$$z_1 = 0.6 \times 10.5 = 6.3$$

$$z_2 = 0.2 \times 10.5 = 2.1$$

$$z_3 = -0.1 \times 10.5 = -1.05$$

$$z_4 = 0.1 \times 10.5 = 1.05$$

Didapatkan nilai logits:

$$z = [6.3, 2.1, -1.05, 1.05]$$

11) Softmax

Setelah dioperasikan dengan logits, lalu mengurangi dengan nilai maksimum yaitu 6.3 untuk stabilitas:

$$\tilde{z} = [0, -4.2, -7.35, -5.25]$$

Kemudian melakukan perhitungan eksponensial dengan hasil berikut:

$$e^{\tilde{z}} \approx [1, 0.0149, 0.00064, 0.00525]$$

Lalu menjumlahkan hasil perhitungan eksponensial:

$$1 + 0.0149 + 0.00064 + 0.00525 = 1.02079$$

Diperoleh probabilitas untuk masing-masing kelas:

$$p_1 = \frac{1}{1.02079} \approx 0.9796$$

$$p_2 = \frac{0.0149}{1.02079} \approx 0.0146$$

$$p_3 = \frac{0.00064}{1.02079} \approx 0.0006$$

$$p_4 = \frac{0.00525}{1.02079} \approx 0.0052$$

Sehingga hasil akhir:

$$p \approx [0.9796, 0.0146, 0.0006, 0.0052]$$

c. Data Output dan Evaluasi

Pada tahap akhir perhitungan ResNet-50, model menghasilkan nilai probabilitas tiap kelas kerusakan jalan melalui fungsi Softmax, dan kelas dengan probabilitas tertinggi ditetapkan sebagai hasil klasifikasi. Berdasarkan hasil perhitungan, diperoleh nilai probabilitas sebagai berikut:

$$p \approx [0.9796, 0.0146, 0.0006, 0.0052]$$

Kelas dengan probabilitas tertinggi adalah kelas ke-1, yaitu “Lubang”, dengan nilai sekitar 97,96% sehingga ditetapkan sebagai hasil prediksi akhir. Probabilitas ini dibandingkan dengan label

sebenarnya untuk mengevaluasi kinerja model menggunakan *confusion matrix* dan menghitung metrik seperti *precision*, *recall*, dan *F1-Score*.

Selain digunakan untuk evaluasi performa model, hasil klasifikasi kerusakan jalan juga dimanfaatkan sebagai dasar rekomendasi dan penentuan prioritas penanganan jalan. Berdasarkan pendekatan *Pavement Condition Index* (PCI), kerusakan berupa “Lubang” termasuk dalam kategori kerusakan dengan prioritas perbaikan sangat tinggi, karena berdampak langsung terhadap keselamatan pengguna jalan dan dapat mempercepat penurunan kondisi perkerasan secara signifikan. Oleh karena itu, ketika model mendeteksi kerusakan dengan kelas “Lubang” dan probabilitas tinggi, sistem dapat merekomendasikan tindakan penanganan segera sebagai prioritas utama.

Dengan demikian, output model tidak hanya berfungsi sebagai hasil klasifikasi visual berbasis *deep learning*, tetapi juga sebagai informasi pendukung pengambilan keputusan dalam sistem manajemen pemeliharaan jalan. Integrasi antara hasil prediksi ResNet-50 dan konsep prioritas perbaikan memungkinkan sistem memberikan rekomendasi yang lebih objektif, efektif, dan sesuai dengan kondisi kerusakan aktual di lapangan.

BAB IV

HASIL DAN PEMBAHASAN

A. Hasil Penelitian

Pada bagian ini disajikan hasil penelitian yang meliputi implementasi, pengujian, serta alat ukur terhadap sistem yang telah dikembangkan.

1. Implementasi Desain Sistem

Implementasi desain sistem dilakukan untuk menerapkan rancangan sebelumnya ke dalam bentuk aplikasi yang dapat dijalankan, meliputi antarmuka, fitur sistem, dan integrasi model deteksi.

a. Implementasi Lembar Kerja

Implementasi lembar kerja menyajikan hasil rancangan sistem yang telah dikembangkan dalam bentuk modul-modul fungsional. Sistem yang dibangun terdiri dari dua modul utama, yaitu Modul Laporan dan Modul Sistem Cerdas. Masing-masing modul memiliki fungsi dan tujuan yang berbeda namun saling mendukung satu sama lain dalam membentuk sistem pelaporan kerusakan jalan yang terintegrasi.

1) Lembar Kerja/Modul Pelaporan

Lembar kerja pelaporan merupakan modul utama yang digunakan pengguna untuk berinteraksi dengan sistem secara langsung. Modul ini mencakup seluruh alur pelaporan kerusakan jalan, mulai dari proses registrasi akun hingga pemantauan sebaran laporan pada peta digital. Setiap tahapan pada modul ini dirancang agar dapat dijalankan secara mudah dan terstruktur oleh pengguna.

a) Register atau Login

Register berfungsi sebagai gerbang masuk pengguna ke dalam sistem. Tujuan dibuatnya adalah untuk memastikan bahwa setiap laporan yang masuk ke sistem dapat diidentifikasi dan dikaitkan dengan akun pengguna yang

terdaftar, sehingga data laporan lebih terorganisasi dan dapat dipertanggungjawabkan.



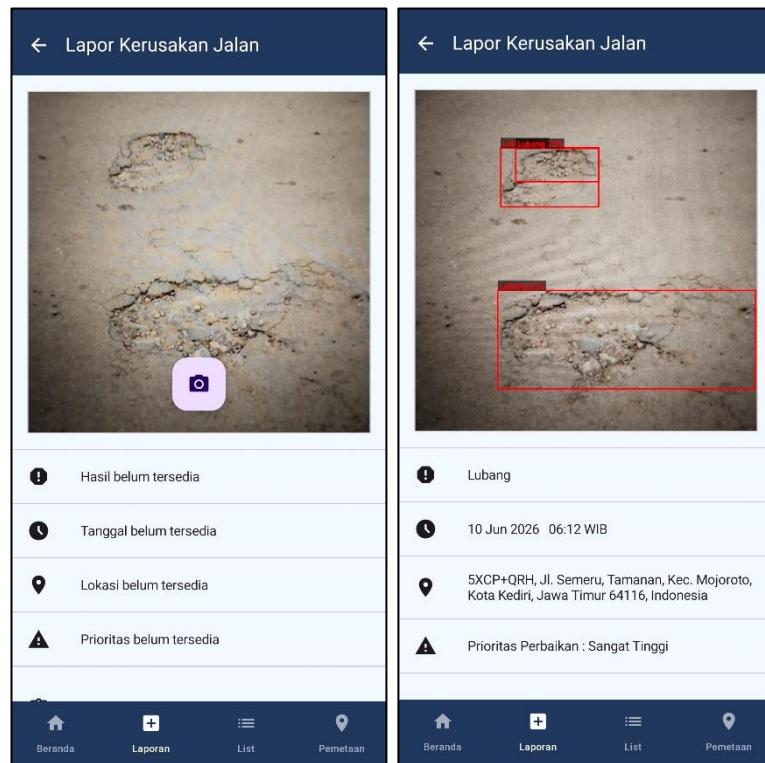
The image shows a mobile application interface for SIRA (Sistem Inspeksi dan Pelaporan Kerusakan Jalan Raya). The top section has a blue background with a white logo featuring a magnifying glass over a road. Below the logo, the text 'SIRA' is displayed in white, followed by 'Sistem Inspeksi dan Pelaporan Kerusakan Jalan Raya' in smaller white text. The main content area is white and contains the word 'REGISTER' in bold blue letters. Below this, there are two input fields: 'Username' and 'Password'. A blue button with the text 'DAFTAR' is positioned below the password field. At the bottom of the white area, there is a link that says 'Sudah punya akun? Login'.

Gambar 4.1 Tampilan Register

Melalui register ini, pengguna mengisi *username* dan *password* untuk membuat akun yang akan digunakan pada seluruh sesi penggunaan aplikasi berikutnya.

b) Lapor Kerusakan

Lapor kerusakan merupakan inti dari lembar kerja pelaporan yang berfungsi sebagai media bagi pengguna untuk mendokumentasikan kondisi jalan rusak melalui kamera perangkat dan mengirimkan laporan lengkap ke server. Lapor kerusakan ini dirancang untuk mengintegrasikan proses pengambilan gambar, deteksi otomatis oleh model kecerdasan buatan, pengambilan lokasi GPS, serta pengiriman data laporan ke dalam satu alur yang mudah digunakan tanpa memerlukan keahlian teknis.



(a)

(b)

Gambar 4.2 Tampilan Laporan Kerusakan: (a) Kondisi Awal, (b) Kondisi Akhir

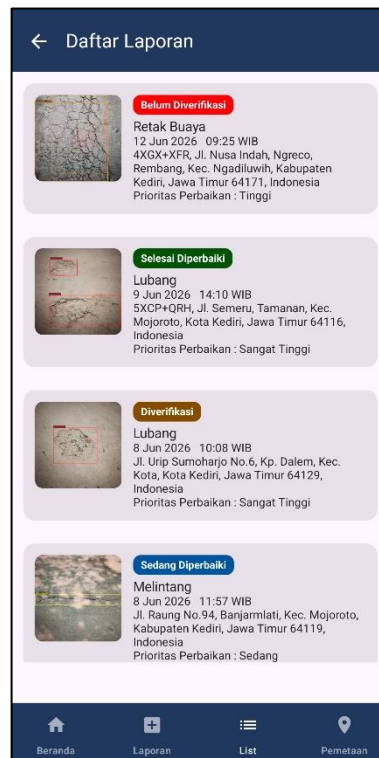
Pada kondisi awal pada Gambar 4.2(a), sistem menampilkan area kamera untuk mengambil gambar kerusakan jalan. Informasi hasil deteksi, waktu, lokasi, dan prioritas perbaikan belum tersedia, sedangkan tombol Kirim Laporan masih belum aktif hingga data yang diperlukan terpenuhi.

Setelah foto diambil pada Gambar 4.2(b), sistem memproses citra menggunakan model ResNet-50 untuk klasifikasi jenis kerusakan dan YOLOv12 untuk menampilkan lokasi objek kerusakan dalam bentuk *bounding box* beserta jenis kerusakan yang teridentifikasi. Pada proses pelaporan, sistem juga secara otomatis memperoleh koordinat posisi pengguna melalui GPS perangkat untuk memastikan setiap laporan memiliki informasi spasial yang

akurat tanpa perlu diisi secara manual. Koordinat tersebut kemudian dikonversi menjadi alamat yang dapat dibaca dan disertakan bersama data laporan sehingga lokasi kerusakan jalan dapat diidentifikasi secara tepat oleh pihak pengelola infrastruktur. Selain menampilkan hasil deteksi berupa *bounding box* dan jenis kerusakan, sistem juga menampilkan waktu pelaporan, alamat lokasi, tingkat prioritas perbaikan, serta kolom catatan tambahan. Setelah seluruh data tersedia, laporan dapat dikirim ke server dan disimpan pada basis data terpusat.

c) Daftar Laporan

Daftar laporan berfungsi menampilkan seluruh riwayat laporan kerusakan jalan yang telah dikirimkan dan tersimpan pada basis data sistem.

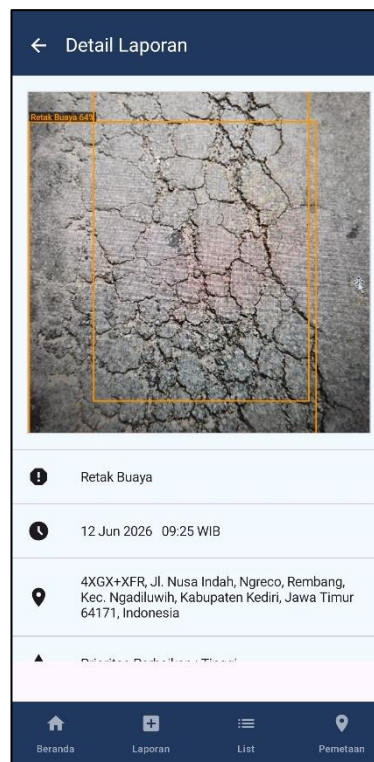


Gambar 4.3 Tampilan Daftar Laporan

Tujuan pembuatannya adalah memberikan kemudahan bagi pengguna maupun petugas dalam memantau dan meninjau kembali laporan yang telah masuk secara terstruktur. Setiap laporan ditampilkan dalam bentuk kartu informasi yang memuat data ringkas seperti gambar, jenis kerusakan, lokasi, waktu, dan tingkat prioritas perbaikan sehingga informasi dapat dipahami dengan cepat tanpa perlu membuka detail masing-masing laporan.

d) **Detail Laporan**

Detail laporan berfungsi menampilkan informasi lengkap dari satu laporan kerusakan jalan yang dipilih pengguna.



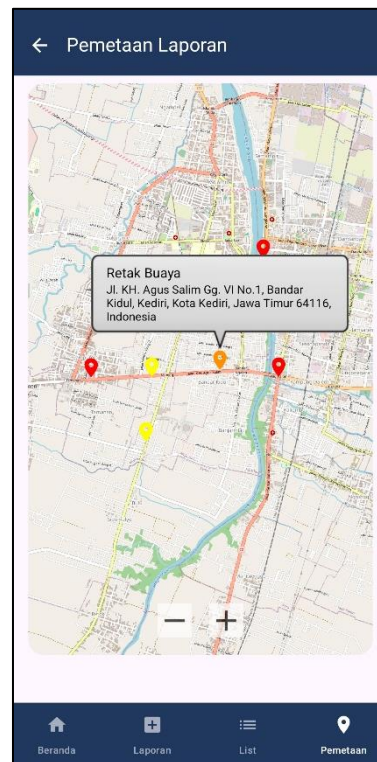
Gambar 4.4 Tampilan Detail Laporan

Tujuan pembuatannya adalah memberikan akses terhadap seluruh data laporan secara menyeluruh, termasuk gambar hasil deteksi dengan *bounding box*, jenis kerusakan,

koordinat GPS, waktu pelaporan, tingkat prioritas perbaikan, dan catatan tambahan.

e) Pemetaan Laporan

Pemetaan laporan berfungsi menampilkan visualisasi sebaran titik lokasi kerusakan jalan pada peta digital berdasarkan seluruh laporan yang tersimpan di sistem.



Gambar 4.5 Tampilan Pemetaan Sebaran Laporan

Pemetaan laporan dibuat dengan tujuan untuk memberikan gambaran menyeluruh mengenai kondisi infrastruktur jalan di suatu wilayah secara visual dan mudah dipahami. Melalui ini, instansi terkait dapat memantau pola sebaran kerusakan, mengidentifikasi wilayah dengan tingkat kerusakan tinggi, dan menentukan prioritas penanganan perbaikan jalan secara lebih efektif dan berbasis data.

2) Lembar Kerja/Modul Sistem Cerdas

Lembar kerja sistem cerdas merupakan modul yang berjalan di sisi server Django dan berfungsi sebagai mesin pemrosesan kecerdasan buatan dalam sistem. Tujuan utama modul ini adalah melakukan identifikasi kerusakan jalan secara otomatis dari gambar yang dikirimkan oleh aplikasi Android, sehingga pengguna tidak perlu menentukan jenis dan lokasi kerusakan secara manual. Modul ini terdiri dari tiga fungsi yang bekerja secara berurutan dalam satu siklus inferensi

a) Input Gambar

Input gambar berfungsi sebagai pintu masuk data ke dalam lembar kerja sistem cerdas. Tujuan pembuatan input gambar adalah memastikan gambar yang dikirimkan oleh aplikasi Android diterima dengan baik dan dipersiapkan secara tepat sebelum diproses oleh model kecerdasan buatan. Proses pada komponen ini mencakup penerimaan data gambar melalui REST API dan penyimpanan sementara pada direktori server.

Setelah gambar diterima, sistem melakukan proses *resize* atau pengubahan ukuran citra sebelum diteruskan ke model kecerdasan buatan. Proses ini diperlukan karena setiap model memiliki ukuran input yang telah ditetapkan selama pelatihan, sementara gambar dari perangkat Android memiliki resolusi yang bervariasi tergantung spesifikasi kamera pengguna. *Resize* dilakukan dengan metode interpolasi untuk mempertahankan proporsi objek kerusakan agar tidak terdistorsi, dilanjutkan dengan normalisasi nilai piksel ke rentang yang sesuai kebutuhan model sebelum proses inferensi dijalankan.

b) Inferensi Model (ResNet-50 dan YOLOv12)

Inferensi model berfungsi menjalankan proses deteksi dan klasifikasi kerusakan jalan menggunakan dua model *deep learning* secara berurutan. Model ResNet-50 bertugas mengklasifikasikan jenis kerusakan jalan dari gambar yang telah dipersiapkan dan menghasilkan prediksi berupa label kelas beserta nilai kepercayaan dari empat kategori kerusakan, yaitu lubang, retak melintang, retak memanjang, dan retak buaya. Selanjutnya model YOLOv12 melokalisasi setiap objek kerusakan dalam gambar dan menghasilkan koordinat *bounding box* beserta label kelas untuk setiap objek yang terdeteksi. Tujuannya adalah untuk menghasilkan informasi kerusakan jalan yang komprehensif dari satu gambar, yaitu jenis kerusakan dari ResNet-50 sekaligus lokasi objek kerusakan dalam gambar dari YOLOv12.

c) Pengembalian Hasil

Pengembalian hasil berfungsi mengemas dan mengirimkan seluruh output inferensi dari kedua model kembali ke aplikasi Android. Tujuan pembuatannya adalah memastikan hasil deteksi dan klasifikasi dapat dikomunikasikan secara efisien ke sisi klien dalam format yang dapat langsung ditampilkan pada antarmuka pengguna. Hasil yang dikembalikan mencakup label jenis kerusakan, nilai kepercayaan klasifikasi, koordinat *bounding box* setiap objek yang terdeteksi, serta gambar yang telah dianotasi dalam format siap tampil, seluruhnya dikemas dalam format JSON dan dikirimkan melalui REST API.

b. Keterkaitan Antar Lembar Kerja

Lembar kerja pelaporan dan lembar kerja sistem cerdas dirancang saling terhubung dan bekerja secara terintegrasi dalam satu alur sistem pelaporan kerusakan jalan yang utuh. Lembar kerja pelaporan

berperan sebagai antarmuka pengguna yang mengelola seluruh proses interaksi di sisi Android, sedangkan lembar kerja sistem cerdas berperan sebagai pemroses kecerdasan buatan yang bekerja di sisi server Django. Hubungan antar kedua lembar kerja berlangsung secara dua arah melalui komunikasi REST API, di mana komponen lapor kerusakan pada lembar kerja pelaporan menjadi titik penghubung utama yang mengirimkan data gambar ke lembar kerja sistem cerdas dan menerima kembali hasil identifikasi untuk ditampilkan kepada pengguna. Gambaran alur kerja yang menunjukkan keterkaitan antar kedua lembar kerja beserta seluruh komponennya adalah sebagai berikut.

- 1) Register pada lembar kerja pelaporan menjadi titik awal sistem. Pengguna membuat akun untuk dapat mengakses seluruh komponen aplikasi berikutnya.
- 2) Lapor kerusakan diaktifkan oleh pengguna melalui menu Lapor Kerusakan. Pada tahap ini, kamera perangkat aktif dan komponen ambil lokasi GPS pada lembar kerja pelaporan bekerja secara bersamaan di latar belakang untuk mendapatkan koordinat posisi pengguna secara otomatis.
- 3) Gambar yang diambil pengguna dikirimkan oleh komponen lapor kerusakan ke lembar kerja sistem cerdas melalui REST API. Penerimaan gambar pada lembar kerja sistem cerdas menerima dan mempersiapkan gambar sebelum diidentifikasi.
- 4) Inferensi model pada lembar kerja sistem cerdas menjalankan ResNet-50 untuk mengklasifikasikan jenis kerusakan jalan, dilanjutkan dengan YOLOv12 untuk mendeteksi dan melokalisasi objek kerusakan dalam gambar secara berurutan.
- 5) Pengembalian hasil pada lembar kerja sistem cerdas mengemas seluruh output kedua model ke dalam format JSON dan mengirimkannya kembali ke komponen lapor kerusakan pada lembar kerja pelaporan melalui REST API.

- 6) Lapor kerusakan menampilkan hasil yang diterima berupa gambar beranotasi *bounding box*, jenis kerusakan, dan tingkat prioritas perbaikan. Pengguna kemudian menekan tombol Kirim Laporan untuk menyimpan seluruh data ke basis data server.
- 7) Daftar laporan pada lembar kerja pelaporan mengambil data yang telah tersimpan dari basis data server dan menampilkannya dalam bentuk kartu informasi sehingga seluruh riwayat laporan dapat diakses kembali oleh pengguna.
- 8) Detail laporan terhubung dengan komponen daftar laporan. Ketika pengguna memilih salah satu laporan, komponen ini menampilkan informasi lengkap beserta opsi membuka lokasi kerusakan pada aplikasi peta eksternal berdasarkan koordinat yang tersimpan.
- 9) Pemetaan laporan secara otomatis membaca seluruh data koordinat laporan dari basis data dan memvisualisasikannya sebagai titik-titik lokasi pada peta digital, sehingga persebaran kerusakan jalan dapat dipantau secara menyeluruh oleh pengguna maupun petugas kapan saja.

Dengan keterkaitan yang terstruktur antara lembar kerja pelaporan dan lembar kerja sistem cerdas, sistem mampu memproses laporan kerusakan jalan secara otomatis dan efisien. Beban komputasi berat yang dibutuhkan oleh model *deep learning* ditangani sepenuhnya di sisi server sehingga aplikasi Android tetap ringan dan responsif pada berbagai spesifikasi perangkat. Alur kerja yang terintegrasi ini memastikan bahwa setiap gambar yang diambil pengguna dapat menghasilkan laporan kerusakan jalan yang lengkap, mulai dari identifikasi jenis dan lokasi kerusakan hingga visualisasi sebaran laporan pada peta digital.

2. Pengujian Fungsional

Pengujian fungsional dilakukan menggunakan metode *Black Box Testing* untuk memverifikasi bahwa setiap fitur dan fungsi pada aplikasi

berjalan sesuai dengan kebutuhan sistem yang telah dirancang. Pengujian ini berfokus pada input dan output sistem tanpa memperhatikan implementasi kode secara internal.

Pengujian lembar kerja/modul pelaporan dilakukan terhadap enam fitur yang berjalan di sisi aplikasi Android, yaitu fitur register, lapor kerusakan, ambil lokasi GPS, daftar laporan, detail laporan, dan pemetaan laporan. Hasil pengujian lembar kerja pelaporan disajikan pada Tabel 4.1 berikut.

Tabel 4.1 Hasil Pengujian Fungsional Modul Pelaporan

No	Nama Fitur	Skenario Uji	Input	Output yang Diharapkan	Output Aktual	Status
1	Register	Registrasi akun berhasil dengan data valid	Username dan password lebih 8 karakter	Akun tersimpan, pengguna diarahkan ke halaman utama	Akun tersimpan, diarahkan ke halaman utama	✓
2	Register	Registrasi gagal jika username sudah terdaftar	username yang sudah terdaftar di sistem	Muncul pesan error 'username sudah digunakan'	Muncul pesan error 'username sudah digunakan'	✓
3	Register	Registrasi gagal jika kolom wajib kosong	Salah satu kolom dibiarkan kosong	Muncul pesan validasi pada kolom yang kosong	Muncul pesan validasi pada kolom yang kosong	✓
4	Register	Registrasi gagal jika konfirmasi password tidak cocok	Password dan konfirmasi password berbeda	Muncul pesan 'Konfirmasi password tidak cocok'	Muncul pesan 'Konfirmasi password tidak cocok'	✓
5	Lapor Kerusakan	Kamera berhasil diaktifkan	Pengguna membuka menu Lapor Kerusakan	Area pratinjau kamera tampil pada layar	Area pratinjau kamera tampil	✓
6	Lapor Kerusakan	Gambar berhasil diambil dan ditampilkan	Pengguna menekan tombol ambil gambar	Gambar tampil pada area pratinjau laporan	Gambar tampil pada area pratinjau	✓

Lanjutan Tabel 4.1 Hasil Pengujian Fungsional Modul Pelaporan

No	Nama Fitur	Skenario Uji	Input	Output yang Diharapkan	Output Aktual	Status
7	Lapor Kerusakan	Gambar dikirim ke server setelah diambil	Gambar berhasil ditangkap kamera	Request API terkirim ke server dengan gambar terlampir	Request API terkirim ke server	✓
8	Lapor Kerusakan	Tombol Kirim Laporan tidak aktif sebelum deteksi selesai	Gambar baru diambil, hasil deteksi belum diterima	Tombol Kirim Laporan berstatus nonaktif (<i>disabled</i>)	Tombol Kirim Laporan berstatus nonaktif	✓
9	Lapor Kerusakan	Hasil deteksi tampil setelah respons server diterima	Respons JSON diterima dari server	Gambar beranotasi <i>bounding box</i> , jenis kerusakan, dan prioritas tampil	Gambar anotasi dan info kerusakan tampil	✓
10	Lapor Kerusakan	Laporan berhasil dikirim ke basis data	Pengguna menekan tombol Kirim Laporan	Data tersimpan di server, muncul notifikasi berhasil	Data tersimpan, notifikasi berhasil muncul	✓
11	Lapor Kerusakan	Kolom catatan tambahan bersifat opsional	Kolom catatan dibiarkan kosong	Laporan tetap dapat dikirim tanpa catatan tambahan	Laporan terkirim tanpa catatan tambahan	✓
12	Ambil Lokasi GPS	Koordinat GPS diperoleh secara otomatis	GPS perangkat aktif, sinyal tersedia	Koordinat latitude dan longitude tampil secara otomatis	Koordinat tampil otomatis	✓
13	Ambil Lokasi GPS	Alamat lokasi dikonversi dari koordinat	Koordinat GPS berhasil didapat	Alamat lokasi tampil menggantikan koordinat mentah.	Alamat lokasi tampil dengan benar	✓
14	Ambil Lokasi GPS	Meminta izin akses lokasi jika belum diberikan	Izin lokasi belum pernah diberikan pengguna	Muncul dialog permintaan izin akses lokasi	Dialog permintaan izin lokasi muncul	✓

Lanjutan Tabel 4.1 Hasil Pengujian Fungsional Modul Pelaporan

No	Nama Fitur	Skenario Uji	Input	Output yang Diharapkan	Output Aktual	Status
15	Daftar Laporan	Daftar laporan tampil dengan data terbaru di atas	Membuka menu Daftar Laporan	Seluruh laporan tampil dalam kartu informasi, diurutkan dari terbaru	Laporan tampil berurutan dari terbaru	✓
16	Daftar Laporan	Tampil pesan jika belum ada laporan	Membuka Daftar Laporan saat belum ada data	Muncul pesan 'Belum ada laporan yang tersimpan'	Pesan kosong tampil dengan benar	✓
17	Daftar Laporan	Setiap kartu laporan berisi informasi ringkas	Laporan tersimpan di basis data	Kartu menampilkan gambar, jenis kerusakan, lokasi, waktu, dan prioritas	Semua informasi ringkas tampil di kartu	✓
18	Daftar Laporan	Daftar laporan dapat digulir untuk melihat semua data	Terdapat lebih dari satu laporan tersimpan	Pengguna dapat menggulir daftar ke bawah tanpa error	Daftar dapat digulir dengan lancar	✓
19	Daftar Laporan	Laporan baru muncul setelah berhasil dikirim	Laporan baru dikirimkan lalu membuka Daftar Laporan	Laporan terbaru muncul pada urutan paling atas daftar	Laporan baru tampil paling atas	✓
20	Detail Laporan	Halaman detail terbuka saat kartu laporan dipilih	Pengguna mengetuk salah satu kartu pada Daftar Laporan	Halaman detail laporan terbuka dengan data lengkap	Halaman detail terbuka dengan benar	✓
21	Detail Laporan	Gambar beranotasi <i>bounding box</i> tampil dengan benar	Membuka halaman detail laporan	Foto kondisi jalan dengan <i>bounding box</i> dan label kerusakan tampil	Gambar beranotasi tampil dengan benar	✓

Lanjutan **Tabel 4.1 Hasil Pengujian Fungsional Modul Pelaporan**

No	Nama Fitur	Skenario Uji	Input	Output yang Diharapkan	Output Aktual	Status
22	Detail Laporan	Informasi lengkap laporan tampil seluruhnya	Membuka halaman detail laporan	Jenis kerusakan, koordinat, waktu, prioritas, dan catatan tampil lengkap	Semua informasi tampil lengkap	✓
23	Detail Laporan	Tombol Lihat Lokasi di Map membuka peta eksternal	Pengguna menekan tombol Lihat Lokasi di Map	Aplikasi peta terbuka pada koordinat lokasi laporan	Google map terbuka pada koordinat yang tepat	✓
24	Pemetaan Laporan	Peta digital berhasil dimuat	Membuka menu Pemetaan Laporan	Peta digital tampil dengan benar pada layar	Peta digital tampil tanpa error	✓
25	Pemetaan Laporan	Titik lokasi laporan tampil pada peta	Terdapat laporan tersimpan di basis data	Titik-titik lokasi laporan tampil pada peta sesuai koordinat	Titik lokasi tampil pada posisi yang tepat	✓
26	Pemetaan Laporan	Informasi ringkas laporan muncul saat titik dipilih	Pengguna mengetuk salah satu titik pada peta	Popup informasi laporan tampil berisi waktu, jenis kerusakan, dan kode laporan	Popup informasi tampil dengan data yang benar	✓

Berdasarkan Tabel 4.1, seluruh 26 skenario pengujian pada lembar kerja pelaporan menunjukkan hasil yang sesuai dengan output yang diharapkan.

3. **Pengujian Non Fungsional**

Pengujian non fungsional dilakukan untuk mengevaluasi kinerja model kecerdasan buatan yang diintegrasikan dalam sistem, yaitu model klasifikasi ResNet-50 dan model deteksi objek YOLOv12.

a. Pengujian Model ResNet-50

Proses pelatihan model ResNet-50 dilakukan menggunakan dataset yang sama dengan empat kelas kerusakan jalan. Model dilatih sebanyak 50 *epoch* dengan memanfaatkan arsitektur *residual connection* yang memungkinkan model mempelajari fitur visual secara mendalam tanpa mengalami masalah *vanishing gradient*. Pengujian dilakukan pada tiga variasi ukuran input citra, yaitu 512×512 , 640×640 , dan 800×800 , dengan parameter evaluasi berupa *precision*, *recall*, dan *F1-Score* untuk menentukan konfigurasi klasifikasi yang paling optimal.

Tabel 4.1 Hasil Pengujian Model ResNet-50

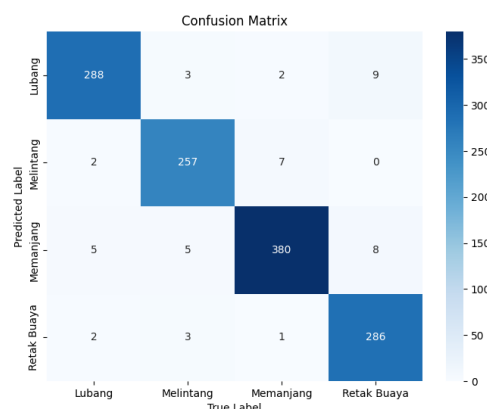
Resize	Class	Precision	Recall	F1-Score
512×512	All	0,96	0,96	0,96
	Lubang	0,95	0,97	0,96
	Melintang	0,97	0,96	0,96
	Memanjang	0,95	0,97	0,96
	Retak Buaya	0,98	0,94	0,96
640×640	All	0,96	0,96	0,96
	Lubang	0,94	0,99	0,96
	Melintang	0,96	0,95	0,95
	Memanjang	0,95	0,97	0,96
	Retak Buaya	0,98	0,92	0,95
800×800	All	0,97	0,96	0,97
	Lubang	0,94	0,97	0,96
	Melintang	0,98	0,96	0,97
	Memanjang	0,97	0,97	0,97
	Retak Buaya	0,97	0,96	0,96

Berdasarkan hasil pengujian menggunakan arsitektur ResNet-50, ukuran *resize* 512×512 menghasilkan performa klasifikasi yang cukup baik dengan nilai *precision*, *recall*, dan *F1-Score* keseluruhan masing-masing sebesar 0,96. Pada pengujian per kelas, kelas Lubang memperoleh *precision* sebesar 0,95, *recall* 0,97, dan *F1-Score* 0,96. Kelas Melintang memperoleh *precision* 0,97, *recall* 0,96, dan *F1-Score* 0,96. Kelas Memanjang menghasilkan *precision* 0,95, *recall* 0,97, dan *F1-Score* 0,96, sedangkan kelas Retak Buaya memperoleh *precision* tertinggi sebesar 0,98, *recall* 0,94, dan *F1-Score* 0,96.

Pada ukuran *resize* 640×640, performa model ResNet-50 menunjukkan hasil yang relatif stabil dengan nilai *precision*, *recall*, dan *F1-Score* keseluruhan tetap sebesar 0,96. Pada pengujian per kelas, kelas Lubang memperoleh *recall* tertinggi sebesar 0,99 dengan *precision* 0,94 dan *F1-Score* 0,96. Kelas Melintang memperoleh *precision* 0,96, *recall* 0,95, dan *F1-Score* 0,95. Kelas Memanjang menghasilkan *precision* 0,95, *recall* 0,97, dan *F1-Score* 0,96, sedangkan kelas Retak Buaya memperoleh *precision* 0,98, *recall* 0,92, dan *F1-Score* 0,95.

Sementara itu, ukuran *resize* 800×800 memberikan performa terbaik dibandingkan ukuran lainnya dengan nilai *precision* keseluruhan sebesar 0,97, *recall* 0,96, dan *F1-Score* 0,97. Pada pengujian per kelas, kelas Lubang memperoleh *precision* 0,94, *recall* 0,97, dan *F1-Score* 0,96. Kelas Melintang menunjukkan performa terbaik dengan *precision* sebesar 0,98, *recall* 0,96, dan *F1-Score* 0,97. Kelas Memanjang juga menghasilkan performa tinggi dengan *precision* dan *recall* masing-masing sebesar 0,97 serta *F1-Score* 0,97. Sementara itu, kelas Retak Buaya memperoleh *precision* 0,97, *recall* 0,96, dan *F1-Score* 0,96.

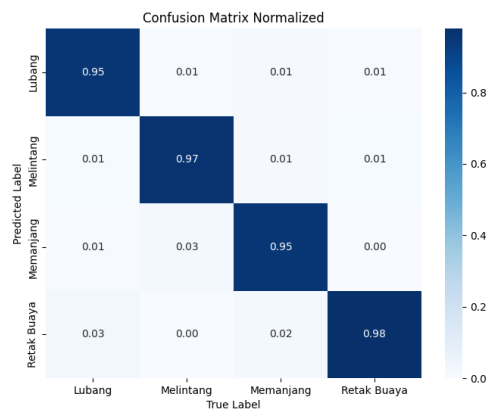
1) *Confusion Matrix* dan Grafik Pelatihan ResNet-50 *Resize* 512x512



Gambar 4.6 *Confusion Matrix* ResNet-50 *Resize* 512x512

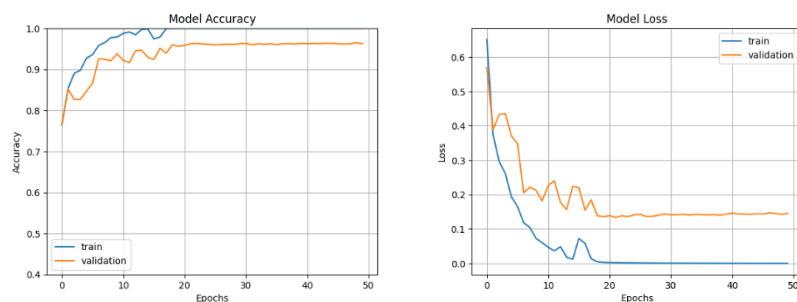
68

Berdasarkan *confusion matrix* ukuran 512×512 , sebagian besar data berhasil diklasifikasikan dengan benar pada setiap kelas. Kelas Lubang memiliki prediksi benar sebanyak 288 data, Melintang 257 data, Memanjang 380 data, dan Retak Buaya 286 data. Namun, masih terdapat beberapa kesalahan klasifikasi antar kelas, terutama pada kelas Lubang yang diprediksi sebagai Retak Buaya sebanyak 9 data serta kelas Memanjang yang diprediksi sebagai Retak Buaya sebanyak 8 data.



Gambar 4.7 Confusion Matrix Normalisasi ResNet-50 Resize 512x512

Berdasarkan *confusion matrix* normalisasi, kelas Lubang memperoleh nilai sekitar 0,95, Melintang 0,97, Memanjang 0,95, dan Retak Buaya 0,98. Meskipun demikian, masih terdapat beberapa kesalahan prediksi antar kelas, khususnya pada kelas Lubang dan Memanjang yang sesekali diprediksi sebagai Retak Buaya.

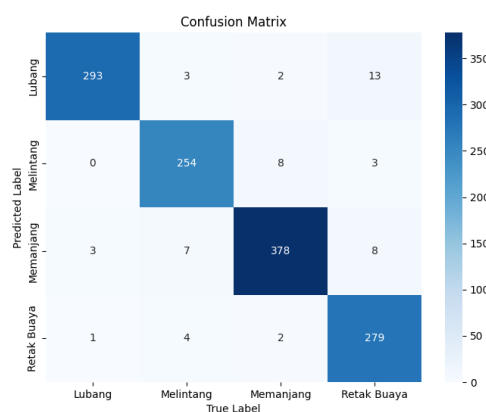


Gambar 4.8 Grafik Pelatihan ResNet-50 Resize 512x512

69

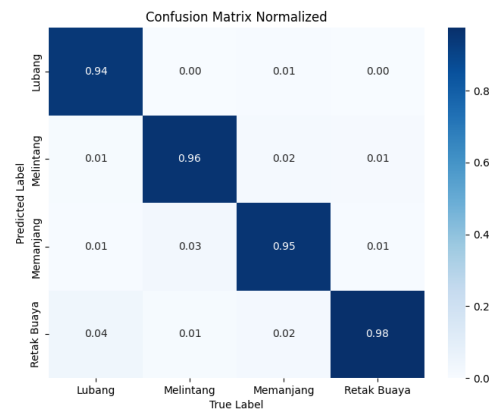
Berdasarkan grafik hasil pelatihan model ResNet50 menggunakan ukuran input gambar 512×512 , terlihat bahwa nilai *train accuracy* dan *validation accuracy* mengalami peningkatan secara bertahap selama proses training berlangsung. Pada awal *epoch*, akurasi validasi berada di kisaran 77%, kemudian meningkat hingga mencapai sekitar 96% pada akhir pelatihan. Selain itu, nilai *train loss* dan *validation loss* menunjukkan tren penurunan yang stabil seiring bertambahnya *epoch*. Nilai *validation loss* pada akhir pelatihan berada di kisaran 0,14, yang menandakan bahwa kesalahan prediksi model semakin kecil.

2) Confusion Matrix dan Grafik Pelatihan ResNet-50 Resize 640x640



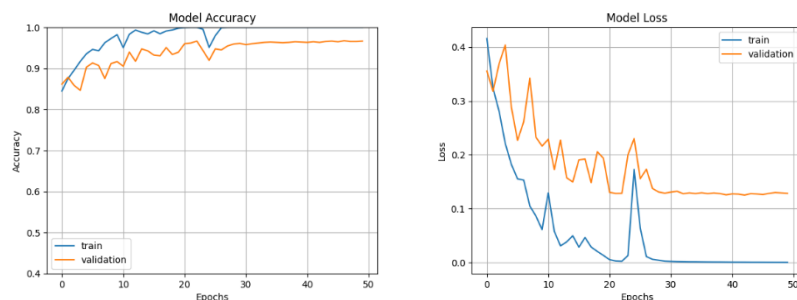
Gambar 4.9 Confusion Matrix ResNet-50 Resize 640x640

Hasil *confusion matrix* pada ukuran 640×640 menunjukkan performa klasifikasi yang tetap baik dengan jumlah prediksi benar yang tinggi pada setiap kelas. Kelas Lubang memiliki prediksi benar sebanyak 293 data, Melintang 254 data, Memanjang 378 data, dan Retak Buaya 279 data. Beberapa kesalahan klasifikasi masih ditemukan, terutama pada kelas Lubang yang diprediksi sebagai Retak Buaya sebanyak 13 data serta kelas Memanjang yang diprediksi sebagai Retak Buaya sebanyak 8 data.



**Gambar 4.10 Confusion Matrix Normalisasi ResNet-50
Resize 640x640**

Pada *confusion matrix* normalisasi, nilai akurasi tiap kelas juga menunjukkan hasil yang tinggi. Kelas Lubang memperoleh nilai sekitar 0,94, Melintang 0,96, Memanjang 0,95, dan Retak Buaya 0,98.



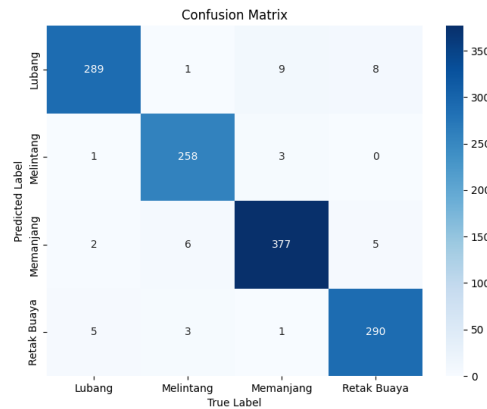
Gambar 4.11 Grafik Pelatihan ResNet-50 Resize 640x640

Pada proses pelatihan menggunakan ukuran input gambar 640×640, grafik menunjukkan peningkatan *train accuracy* dan *validation accuracy* yang cukup konsisten hingga akhir *epoch*. Nilai *validation accuracy* meningkat dari sekitar 85% pada awal pelatihan hingga mencapai kisaran 96–97% pada akhir *epoch*.

Nilai *train loss* dan *validation loss* juga mengalami penurunan secara bertahap selama proses training berlangsung. *Validation loss* pada akhir pelatihan berada di kisaran 0,13. Meskipun terdapat beberapa lonjakan pada *validation loss* di

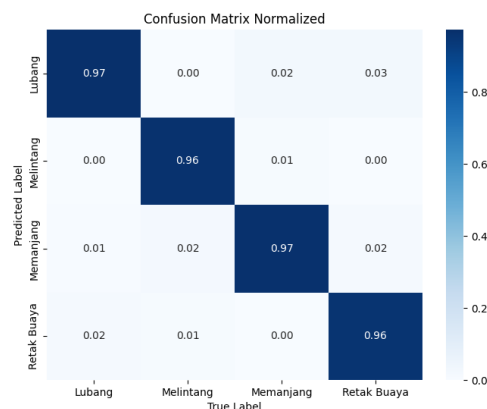
beberapa *epoch*, model tetap mampu kembali stabil pada *epoch* berikutnya.

3) Confusion Matrix dan Grafik Pelatihan ResNet-50 Resize 800x800



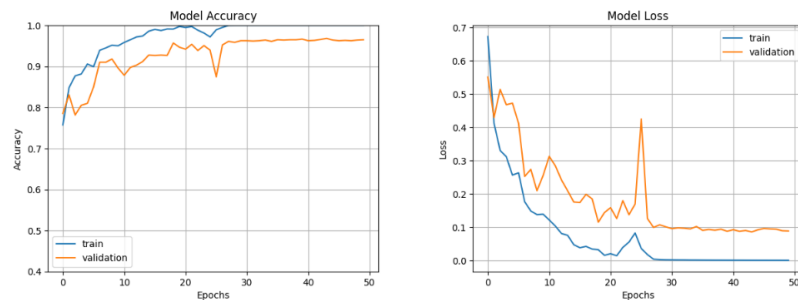
Gambar 4.12 Confusion Matrix ResNet-50 Resize 800x800

Pada ukuran input 800×800, *confusion matrix* menunjukkan hasil klasifikasi dengan jumlah prediksi benar pada masing-masing kelas yaitu Lubang sebanyak 289 data, Melintang 258 data, Memanjang 377 data, dan Retak Buaya 290 data. Kesalahan klasifikasi antar kelas juga relatif lebih sedikit, meskipun masih terdapat beberapa kesalahan pada kelas Memanjang yang diprediksi sebagai Lubang sebanyak 9 data.



**Gambar 4.13 Confusion Matrix Normalisasi ResNet-50
Resize 800x800**

Berdasarkan *confusion matrix* normalisasi, kelas Lubang memperoleh nilai sekitar 0,97, Melintang 0,96, Memanjang 0,97, dan Retak Buaya 0,96.



Gambar 4. 14 Grafik Pelatihan ResNet-50 *Resize* 800x800

Berdasarkan grafik pelatihan model ResNet50 dengan ukuran input gambar 800×800, terlihat bahwa nilai *train accuracy* dan *validation accuracy* meningkat secara signifikan selama proses training. *Validation accuracy* mencapai kisaran 96–97% pada akhir *epoch*. Selain itu, nilai *train loss* mengalami penurunan hingga mendekati nol pada akhir pelatihan, sedangkan *validation loss* turun hingga sekitar 0,09.

b. Pengujian Model YOLOv12

Proses pelatihan model YOLOv12 dilakukan menggunakan dataset kerusakan jalan yang terdiri dari empat kelas kerusakan jalan raya. Model dilatih sebanyak 250 *epoch* untuk memastikan proses pembelajaran yang cukup hingga model mencapai konvergensi. Pengujian model YOLOv12 dilakukan untuk mengetahui performa deteksi kerusakan jalan berdasarkan variasi ukuran *input image*. Pengujian dilakukan menggunakan ukuran citra 512×512, 640×640, dan 800×800 dengan parameter evaluasi berupa mAP50, mAP50-95, *precision*, dan *recall*. Hasil pengujian digunakan untuk menentukan konfigurasi model dengan performa terbaik dalam mendeteksi kerusakan jalan.

Tabel 4.2 Hasil Pengujian Model YOLOv12

Resize	Class	mAP50	mAP50-95	Precision	Recall
512x512	All	0.931	0.684	0.909	0.890
	Lubang	0.938	0.690	0.905	0.882
	Melintang	0.942	0.647	0.920	0.904
	Memanjang	0.932	0.677	0.896	0.883
	Retak Buaya	0.911	0.721	0.914	0.891
640x640	All	0.937	0.704	0.898	0.905
	Lubang	0.942	0.716	0.915	0.877
	Melintang	0.942	0.680	0.884	0.934
	Memanjang	0.935	0.693	0.887	0.895
	Retak Buaya	0.930	0.729	0.908	0.915
800x800	All	0.939	0.711	0.916	0.894
	Lubang	0.942	0.710	0.921	0.867
	Melintang	0.955	0.702	0.921	0.928
	Memanjang	0.942	0.689	0.908	0.890
	Retak Buaya	0.917	0.741	0.915	0.891

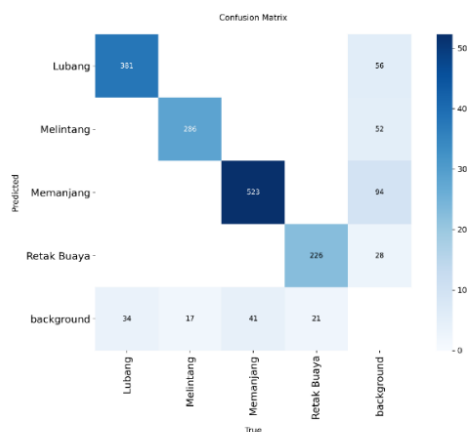
Berdasarkan hasil pengujian, ukuran *resize* 512×512 menghasilkan nilai mAP50 sebesar 0,931, mAP50-95 sebesar 0,684, *precision* sebesar 0,909, dan *recall* sebesar 0,890. Pada pengujian per kelas, nilai mAP50 tertinggi diperoleh kelas Melintang sebesar 0,942, sedangkan nilai mAP50-95 tertinggi diperoleh kelas Retak Buaya sebesar 0,721.

Pada ukuran *resize* 640×640, performa model mengalami peningkatan dengan nilai mAP50 sebesar 0,937, mAP50-95 sebesar 0,704, *precision* sebesar 0,898, dan *recall* sebesar 0,905. Pada pengujian per kelas, nilai mAP50 tertinggi diperoleh kelas Lubang dan Melintang sebesar 0,942, sedangkan nilai mAP50-95 tertinggi diperoleh kelas Retak Buaya sebesar 0,729.

Sementara itu, ukuran *resize* 800×800 memberikan hasil terbaik dibandingkan ukuran lainnya dengan nilai mAP50 sebesar 0,939, mAP50-95 sebesar 0,711, *precision* sebesar 0,916, dan *recall* sebesar 0,894. Pada pengujian per kelas, nilai mAP50 tertinggi diperoleh kelas Melintang sebesar 0,955 dan nilai mAP50-95 tertinggi diperoleh kelas Retak Buaya sebesar 0,741.

Selain menggunakan parameter evaluasi kuantitatif, performa model juga dianalisis menggunakan *confusion matrix* dan grafik hasil pelatihan untuk melihat tingkat ketepatan prediksi serta kestabilan proses training pada setiap ukuran *resize image*. *Confusion matrix* digunakan untuk menunjukkan hubungan antara label aktual dan hasil prediksi model sehingga dapat diketahui pola kesalahan klasifikasi pada setiap kelas kerusakan jalan. Sementara itu, grafik pelatihan digunakan untuk melihat perkembangan *nilai loss*, *precision*, *recall*, serta *mAP* selama proses training berlangsung.

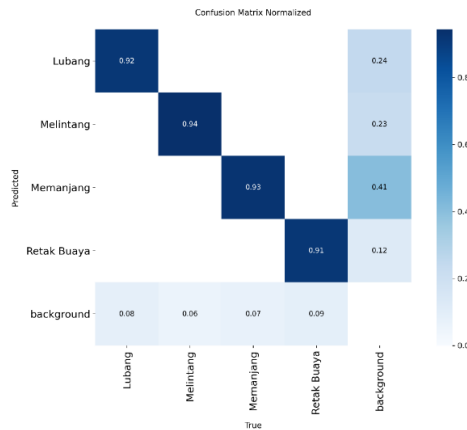
1) *Confusion Matrix* dan Grafik Pelatihan YOLOv12 *Resize* 512×512



Gambar 4.15 *Confusion Matrix* YOLOv12 *Resize* 512×512

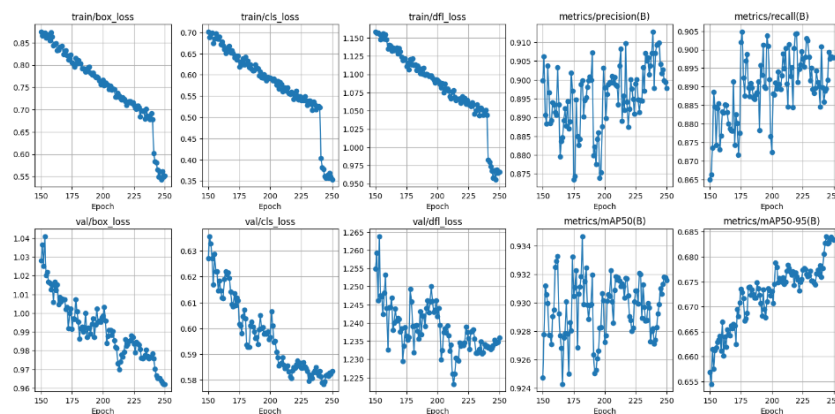
Pada ukuran input 512×512, hasil *confusion matrix* menunjukkan bahwa sebagian besar data berhasil diprediksi sesuai dengan kelas sebenarnya. Hal ini terlihat dari dominasi nilai pada diagonal matriks. Kelas Lubang memiliki nilai prediksi benar sebesar 381 data, Melintang sebesar 286 data, Memanjang sebesar 523 data, dan Retak Buaya sebesar 226 data. Meskipun demikian, masih terdapat beberapa kesalahan klasifikasi antar kelas serta deteksi *background* pada beberapa objek.

75



**Gambar 4.16 Confusion Matrix Normalisasi YOLOv12
Resize 512×512**

Berdasarkan *confusion matrix* normalisasi, tingkat akurasi prediksi tiap kelas juga tergolong baik. Kelas Lubang memperoleh nilai sekitar 0,92, Melintang 0,94, Memanjang 0,93, dan Retak Buaya 0,91. Namun, beberapa objek masih terdeteksi sebagai *background* terutama pada kelas Memanjang dan Lubang.

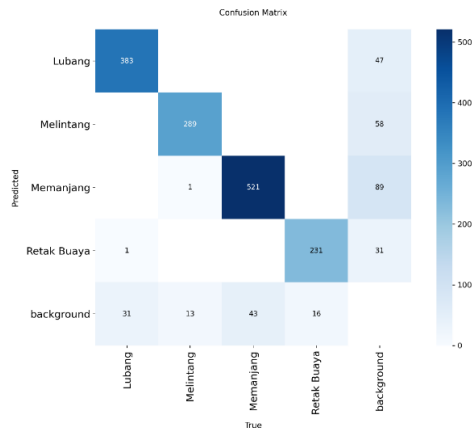


**Gambar 4.17 Grafik Pelatihan YOLOv12 Resize 512×512
epoch 150-250**

Pada grafik pelatihan, nilai *train loss* dan *validation loss* menunjukkan tren penurunan yang cukup stabil hingga akhir *epoch*. Selain itu, nilai *precision* dan *recall* juga mengalami peningkatan secara bertahap meskipun masih terlihat fluktuasi pada beberapa *epoch*. Nilai *mAP50* dan *mAP50-95* juga

mengalami peningkatan selama proses pelatihan. Pada akhir *epoch*, nilai mAP50 mencapai sekitar 0,93, sedangkan mAP50-95 berada pada kisaran 0,68.

2) *Confusion Matrix* dan Grafik Pelatihan YOLOv12 *Resize* 640x640



Gambar 4.18 *Confusion Matrix* YOLOv12 *Resize* 640x640

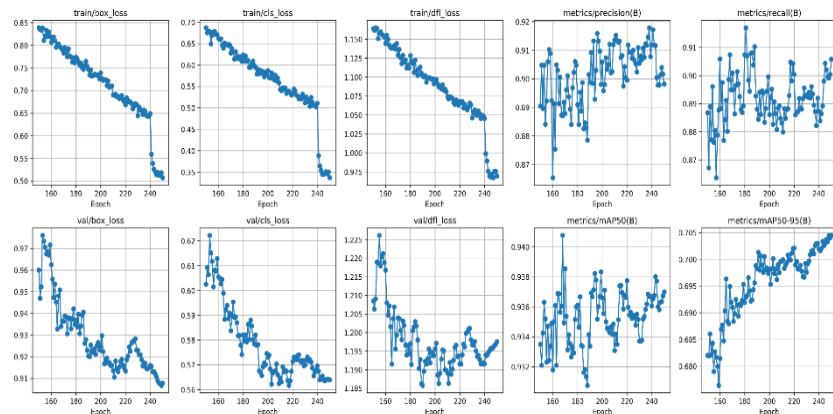
Pada ukuran input 640×640, hasil *confusion matrix* menunjukkan dominasi nilai pada diagonal matriks yang menunjukkan prediksi benar pada masing-masing kelas. Kelas Lubang memiliki nilai prediksi benar sebesar 383 data, Melintang sebesar 289 data, Memanjang sebesar 521 data, dan Retak Buaya sebesar 231 data.



Gambar 4.19 *Confusion Matrix* Normalisasi YOLOv12 *Resize* 640x640

77

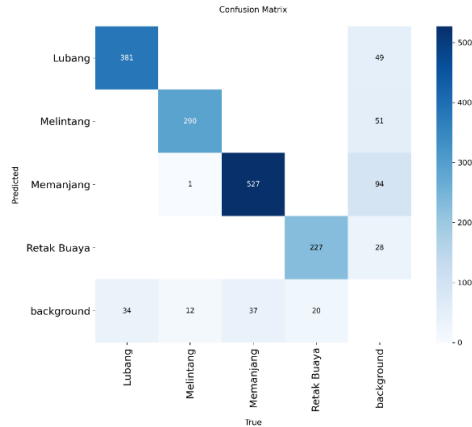
Berdasarkan *confusion matrix* normalisasi, tingkat akurasi prediksi untuk setiap kelas juga tergolong tinggi. Kelas Lubang memperoleh nilai sekitar 0,92, Melintang 0,95, Memanjang 0,92, dan Retak Buaya 0,94. Namun demikian, masih terdapat beberapa prediksi yang masuk ke kategori *background*, terutama pada kelas Memanjang dan Melintang, meskipun jumlahnya relatif kecil.



**Gambar 4.20 Grafik Pelatihan YOLOv12 Resize 640×640
epoch 150-250**

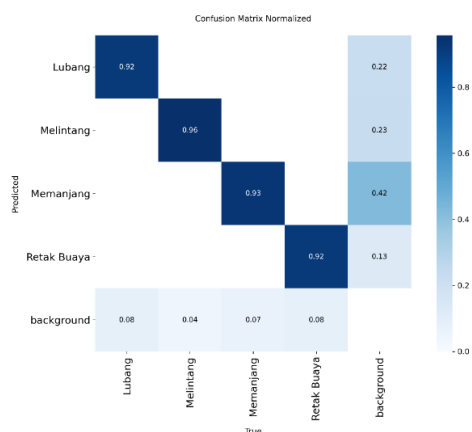
Berdasarkan grafik pelatihan, nilai *train loss* dan *validation loss* menunjukkan tren penurunan yang stabil hingga akhir *epoch*. Selain itu, nilai *precision* dan *recall* mengalami peningkatan secara bertahap selama proses pelatihan, meskipun masih terdapat beberapa fluktuasi pada beberapa *epoch*. Nilai *mAP50* dan *mAP50-95* juga terus meningkat dari awal hingga akhir pelatihan. Pada akhir *epoch*, nilai *mAP50* mencapai sekitar 0,94, sedangkan nilai *mAP50-95* mencapai sekitar 0,70. Perubahan nilai pada setiap metrik dapat diamati secara bertahap seiring bertambahnya jumlah *epoch* selama proses training berlangsung.

3) *Confusion Matrix* dan Grafik Pelatihan YOLOv12 *Resize* 800x800



Gambar 4.21 *Confusion Matrix* YOLOv12 *Resize* 800x800

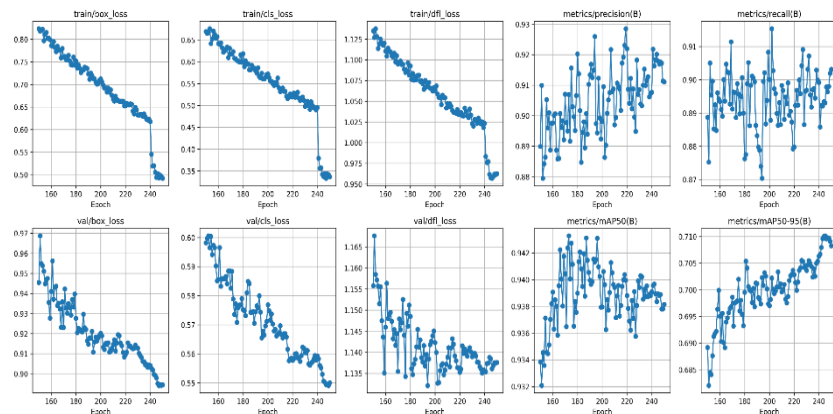
Pada ukuran input 800×800, hasil *confusion matrix* menunjukkan performa model yang baik. Hal ini terlihat dari dominasi nilai pada diagonal matriks yang menandakan sebagian besar data berhasil diprediksi sesuai dengan kelas sebenarnya. Kelas Lubang memiliki nilai prediksi benar sebesar 381 data, Melintang sebesar 290 data, Memanjang sebesar 527 data, dan Retak Buaya sebesar 227 data. Selain itu, tingkat kesalahan klasifikasi antar kelas relatif kecil.



Gambar 4.22 *Confusion Matrix* Normalisasi YOLOv12 *Resize* 800x800

79

Berdasarkan *confusion matrix* normalisasi, tingkat akurasi prediksi pada setiap kelas juga tergolong tinggi. Kelas Lubang memiliki nilai sekitar 0,92, Melintang 0,96, Memanjang 0,93, dan Retak Buaya 0,92. Meskipun demikian, masih terdapat beberapa kesalahan deteksi pada *background* terutama pada kelas Memanjang dan Lubang, yang menunjukkan adanya objek yang belum terdeteksi secara sempurna.



**Gambar 4.23 Grafik Pelatihan YOLOv12 *Resize* 800×800
epoch 150-250**

Pada grafik pelatihan, nilai *train loss* dan *validation loss* mengalami penurunan secara stabil hingga akhir *epoch*. Nilai *precision* dan *recall* juga menunjukkan performa yang stabil dengan rata-rata berada di atas 0,89, meskipun masih terdapat sedikit fluktuasi pada beberapa *epoch* selama proses training berlangsung. Selain itu, nilai mAP50 dan mAP50-95 mengalami peningkatan secara bertahap hingga akhir pelatihan. Pada akhir *epoch*, nilai mAP50 mencapai sekitar 0,94, sedangkan mAP50-95 mencapai sekitar 0,71.

B. Pembahasan

Bagian pembahasan ini menyajikan temuan-temuan utama yang diperoleh dari hasil penelitian, meliputi pengaruh ukuran input citra terhadap performa model, pengaruh jumlah *epoch* pelatihan, serta pengaruh waktu respons sistem. Setiap temuan dianalisis secara rinci disertai posisi penelitian terhadap kajian-

kajian terdahulu untuk memberikan pemahaman terhadap faktor-faktor yang memengaruhi kinerja sistem pelaporan kerusakan jalan berbasis *Convolutional Neural Network* yang dikembangkan.

1. Pengaruh Ukuran Input Citra terhadap Performa Model

Hasil pengujian menunjukkan bahwa secara umum semakin besar ukuran input citra yang digunakan, semakin baik performa deteksi dan klasifikasi yang dihasilkan oleh model YOLOv12 maupun ResNet-50.

a. Pengaruh pada Model ResNet-50

Pada model ResNet-50, ukuran input 512x512 menghasilkan *F1-Score* keseluruhan sebesar 0,96, ukuran 640x640 menghasilkan *F1-Score* yang sama sebesar 0,96, sedangkan ukuran 800x800 menghasilkan *F1-Score* tertinggi sebesar 0,97. Meskipun perbedaan angka terlihat kecil, peningkatan ini konsisten terjadi pada hampir seluruh kelas kerusakan, terutama pada kelas Melintang dan Memanjang yang masing-masing meningkat dari 0,96 menjadi 0,97 ketika ukuran input ditingkatkan ke 800x800.

Fenomena ini terjadi karena mekanisme kerja CNN pada arsitektur ResNet-50 sangat bergantung pada kualitas representasi spasial yang dihasilkan oleh lapisan konvolusi pertama. Pada ukuran input yang lebih kecil seperti 512x512, gambar kerusakan jalan mengalami *downsampling* yang lebih agresif sejak awal proses ekstraksi fitur. Akibatnya, detail tekstur permukaan jalan seperti pola retakan halus, lebar celah, dan batas tepi kerusakan menjadi kurang terwakili dalam peta fitur yang dihasilkan. Sebaliknya, pada ukuran 800x800, lapisan konvolusi awal dapat memproses gambar dengan informasi spasial yang jauh lebih kaya sebelum operasi *max-pooling* dilakukan, sehingga fitur-fitur diskriminatif yang membedakan antarkelas kerusakan dapat diekstraksi dengan lebih presisi. Mekanisme *residual connection* pada ResNet-50 semakin menguatkan keunggulan ini karena memungkinkan informasi detail dari lapisan awal tetap mengalir ke lapisan-lapisan yang lebih dalam

melalui *shortcut connection*, sehingga detail fitur dari gambar beresolusi tinggi dapat dimanfaatkan secara optimal di seluruh kedalaman jaringan.

Temuan ini mendukung penelitian Saparudin dkk., (2025) yang menggunakan CNN berbasis ResNet untuk klasifikasi kerusakan jalan dengan ukuran input default 224×224 piksel dan memperoleh akurasi sebesar 92%. Penelitian ini memperkuat sekaligus melengkapi temuan tersebut dengan membuktikan bahwa peningkatan ukuran input citra diatas resolusi *default*, dari 512×512 hingga 800×800 , secara konsisten meningkatkan *F1-Score* pada seluruh kelas kerusakan jalan, sehingga ukuran input yang lebih besar dari 224×224 yang digunakan Saparudin dkk. terbukti menghasilkan performa klasifikasi yang lebih tinggi, yaitu *F1-Score* sebesar 0,97 pada konfigurasi optimal 800×800 .

b. Pengaruh pada Model YOLOv12

Pada model YOLOv12, pola yang sama teramati pada metrik mAP50 dan mAP50-95. Ukuran input 512×512 menghasilkan mAP50 sebesar 0,931 dan mAP50-95 sebesar 0,684. Ukuran 640×640 meningkatkan mAP50 menjadi 0,937 dan mAP50-95 menjadi 0,704. Ukuran 800×800 menghasilkan mAP50 tertinggi sebesar 0,939 dan mAP50-95 tertinggi sebesar 0,711. Peningkatan yang paling signifikan terlihat pada metrik mAP50-95, yaitu dari 0,684 pada 512×512 menjadi 0,711 pada 800×800 atau meningkat sebesar 3,95%. Peningkatan mAP50-95 yang lebih besar dibandingkan mAP50 mengindikasikan bahwa ukuran input yang lebih besar terutama memberikan manfaat pada presisi lokalisasi bounding box, bukan hanya pada ketepatan klasifikasi kelas.

Hal ini terjadi karena arsitektur YOLOv12 menggunakan *feature pyramid network* yang menghasilkan prediksi pada tiga skala berbeda. Pada ukuran input yang lebih kecil, setiap piksel pada *feature map* merepresentasikan area yang lebih luas pada gambar asli sehingga

prediksi koordinat *bounding box* menjadi kurang presisi. Pada ukuran 800x800, setiap piksel pada feature map merepresentasikan area yang lebih kecil sehingga model dapat memprediksi batas-batas objek kerusakan dengan granularitas yang lebih tinggi, yang secara langsung meningkatkan nilai mAP50-95. Satu temuan menarik adalah nilai *recall* pada ukuran 640x640 sebesar 0,905 yang justru lebih tinggi dibandingkan ukuran 800x800 sebesar 0,894, menunjukkan adanya *trade-off* antara presisi lokalisasi dan sensitivitas deteksi. Pada ukuran 800x800, model cenderung lebih konservatif dalam memberikan prediksi positif karena ekspektasi presisi *bounding box* yang lebih tinggi, sehingga beberapa objek kerusakan berukuran sangat kecil dapat terlewat.

Temuan ini mendukung dan memperluas penelitian Hananta & Cahyono, (2026) dengan konfigurasi *resize* 640x640 yang mengemukakan bahwa model berbasis YOLO mampu mendeteksi kerusakan jalan multi-kelas dengan akurasi tinggi. Penelitian ini tidak hanya mengonfirmasi kemampuan tersebut melalui model YOLOv12 yang lebih baru dengan mAP50 sebesar 0,939, tetapi juga memberikan bukti empiris mengenai hubungan antara ukuran input citra dan presisi lokalisasi *bounding box* yang belum dibahas secara mendalam pada penelitian-penelitian sebelumnya. Selain itu, temuan mengenai *trade-off* antara *recall* dan presisi antar ukuran input memberikan wawasan baru bagi peneliti yang ingin mengoptimalkan konfigurasi model YOLO untuk kasus penggunaan spesifik, baik yang mengutamakan sensitivitas deteksi maupun presisi lokalisasi.

2. Pengaruh Jumlah *Epoch* Pelatihan terhadap Konvergensi Model

Kedua model menggunakan jumlah *epoch* yang berbeda, yaitu 50 *epoch* untuk ResNet-50 dan 250 *epoch* untuk YOLOv12, yang mencerminkan perbedaan kompleksitas tugas yang diemban masing-masing model.

a. *Epoch* pada Model ResNet-50

Model ResNet-50 dilatih sebanyak 50 *epoch* dan menunjukkan konvergensi yang baik berdasarkan grafik pelatihan pada seluruh variasi ukuran input. Untuk melihat pengaruh jumlah *epoch* secara lebih jelas, performa model pada ukuran input 800×800 dibandingkan pada beberapa titik *epoch*, yaitu *epoch* 10, 20, 30, dan 50. Pada *epoch* 10, *validation accuracy* baru mencapai 91,81% dengan *validation loss* sebesar 0,209, menunjukkan bahwa model masih dalam tahap awal pembelajaran dan belum mampu mengenali seluruh pola kerusakan secara optimal. Pada *epoch* 20, *validation accuracy* meningkat signifikan menjadi 95,71% dengan *validation loss* turun menjadi 0,115, mengindikasikan bahwa model mulai mengekstraksi fitur kerusakan jalan dengan lebih baik. Pada *epoch* 30, *validation accuracy* mencapai 95,87% dengan *validation loss* 0,107, menunjukkan model hampir mencapai konvergensi namun masih terdapat ruang peningkatan. Pada *epoch* 50, model mencapai performa terbaik dengan *validation accuracy* sebesar 96,42% dan *validation loss* terendah sebesar 0,089, membuktikan bahwa penambahan *epoch* dari 30 ke 50 masih memberikan peningkatan yang berarti pada kualitas klasifikasi.

Jumlah 50 *epoch* terbukti memadai untuk ResNet-50 karena arsitektur *residual connection* secara inheren mempercepat proses konvergensi. Mekanisme *shortcut connection* memungkinkan *gradient* mengalir langsung dari lapisan output ke lapisan input tanpa melewati seluruh transformasi nonlinear, sehingga masalah *vanishing gradient* yang umum terjadi pada jaringan sangat dalam dapat diminimalkan. Model dapat mempelajari representasi fitur yang bermakna secara lebih efisien dibandingkan arsitektur CNN konvensional, sehingga 50 *epoch* sudah cukup untuk menghasilkan *F1-Score* rata-rata 0,97 tanpa indikasi *overfitting* yang signifikan. Temuan ini mendukung klaim He dkk. dalam penelitian asli ResNet

yang menyatakan bahwa arsitektur *residual connection* secara fundamental mempermudah proses optimasi dan memungkinkan model yang lebih dalam dilatih dengan lebih efisien. Penelitian ini mengonfirmasi bahwa keunggulan tersebut berlaku secara konsisten pada kasus klasifikasi kerusakan jalan dengan empat kelas menggunakan dataset foto lapangan yang memiliki variasi kondisi pencahayaan dan sudut pengambilan gambar.

b. *Epoch* pada Model YOLOv12

Model YOLOv12 memerlukan jumlah *epoch* yang jauh lebih besar, yaitu 250 *epoch*, karena tugas deteksi objek jauh lebih kompleks dibandingkan klasifikasi citra. Kompleksitas ini disebabkan oleh kenyataan bahwa model harus mengoptimalkan fungsi loss gabungan yang mencakup *classification loss*, *objectness loss*, dan *localization loss* secara simultan, sehingga proses optimasi membutuhkan lebih banyak iterasi dibandingkan optimasi fungsi *loss* klasifikasi tunggal pada ResNet-50.

Untuk melihat pengaruh jumlah *epoch* secara lebih jelas, performa model pada ukuran input 800×800 dibandingkan pada beberapa titik *epoch*, yaitu *epoch* 100, 150, 200, dan 250. Pada *epoch* 100, model menghasilkan mAP50 sebesar 0,9221 dan mAP50-95 sebesar 0,6520 dengan *val_box_loss* sebesar 1,0262, menunjukkan bahwa model masih dalam proses pembelajaran aktif dan belum mencapai presisi lokalisasi yang optimal. Pada *epoch* 150, mAP50 meningkat menjadi 0,9338 dan mAP50-95 menjadi 0,6892 dengan *val_box_loss* turun menjadi 0,9455. Pada *epoch* 200, mAP50 sebesar 0,9379 dan mAP50-95 sebesar 0,6970 dengan *val_box_loss* 0,9201, menandakan model semakin mendekati konvergensi meskipun masih terdapat ruang peningkatan. Pada *epoch* 250, model mencapai performa terbaik dengan mAP50 sebesar 0,9382, mAP50-95 sebesar 0,7082, *precision* sebesar 0,9111, *recall* sebesar 0,9031, dan *val_box_loss* terendah sebesar 0,8945. Peningkatan yang konsisten

dari *epoch* 100 hingga 250 membuktikan bahwa penambahan jumlah *epoch* memberikan kontribusi nyata terutama pada presisi lokalisasi *bounding box* yang tercermin dari kenaikan mAP50-95 sebesar 0,0562 poin. Temuan ini mendukung praktik umum dalam penelitian deteksi objek berbasis YOLO yang merekomendasikan jumlah *epoch* pelatihan yang signifikan lebih besar dibandingkan model klasifikasi, sekaligus memberikan data empiris konkret bahwa 250 *epoch* merupakan jumlah yang memadai untuk mencapai nilai mAP50 di atas 0,93 dan mAP50-95 di atas 0,70 secara stabil pada dataset kerusakan jalan empat kelas.

3. Pengaruh Waktu Respons terhadap Pengalaman Pengguna

Sistem yang dikembangkan menempatkan seluruh beban inferensi model di sisi server Django yang berjalan pada laptop, sementara aplikasi Android hanya bertugas mengirim gambar dan menampilkan hasil yang diterima. Pendekatan ini memiliki dampak langsung terhadap kecepatan dan pengalaman pengguna dalam proses pelaporan kerusakan jalan.

Tabel 4.3 Hasil Pengujian Latensi Inferensi Model

No	Upload Gambar (ms)	Inferensi ResNet-50 (ms)	Inferensi YOLOv12 (ms)	Total (ms)
1	151,37	1.900,71	845,65	3.123,65
2	106,44	1.887,30	778,84	2.991,16
3	108,57	1.889,10	837,22	3.169,95
4	130,17	1.928,41	877,45	3.209,94
5	132,88	1.945,10	837,78	3.097,84
6	107,94	1.869,16	955,46	3.181,32
7	115,82	1.874,19	903,78	3.311,53
8	110,06	1.840,75	859,92	3.151,33
9	145,47	1.927,10	863,23	3.229,87
10	127,46	1.861,99	838,01	3.039,97
Rata-Rata	123,62	1.892,38	859,73	3.150,66

Berdasarkan hasil pengujian latensi inferensi model pada Tabel 4.3, rata-rata total waktu respons dari 10 kali percobaan adalah 3.150,66 ms atau sekitar 3,15 detik dalam kondisi jaringan lokal. Waktu respons ini

terdiri dari tiga tahap, yaitu upload gambar dari Android ke server rata-rata 123,62 ms, inferensi model ResNet-50 rata-rata 1.892,38 ms, dan inferensi model YOLOv12 rata-rata 859,73 ms. Tahap inferensi ResNet-50 menjadi tahap yang paling memakan waktu dengan kontribusi rata-rata 1.892,38 ms atau sekitar 60% dari total waktu respons, diikuti inferensi YOLOv12 sebesar 859,73 ms atau sekitar 27%. Kondisi ini disebabkan oleh perbedaan mekanisme pemrosesan kedua model, di mana ResNet-50 dengan ukuran input 800×800 melakukan ekstraksi fitur pada resolusi yang lebih besar dibandingkan YOLOv12 sehingga membutuhkan waktu komputasi yang lebih panjang, terlebih tanpa akselerasi GPU yang optimal pada laptop yang digunakan sebagai server.

Waktu upload yang rata-rata 123,62 ms atau hanya sekitar 4% dari total waktu respons mencerminkan keuntungan penggunaan jaringan lokal yang meminimalkan latensi komunikasi jaringan. Hal ini menunjukkan bahwa *bottleneck* utama sistem bukan pada komunikasi data, melainkan pada kapasitas komputasi server dalam menjalankan inferensi kedua model secara berurutan. Total waktu respons sebesar 3,15 detik masih dapat diterima untuk konteks aplikasi pelaporan di lapangan, karena pengguna tidak perlu melakukan tindakan apa pun selama proses berlangsung dan hasil identifikasi tetap diperoleh secara otomatis dalam satu kali pengambilan foto. Meskipun demikian, waktu respons ini berpotensi untuk dipangkas secara signifikan melalui migrasi server ke infrastruktur berbasis GPU dedicated atau cloud computing, yang dapat menurunkan waktu inferensi ResNet-50 dan YOLOv12 secara drastis mengingat kedua model ini dirancang untuk diakselerasi menggunakan GPU.

Temuan ini mendukung penelitian Angkasa dkk., (2026) yang menyimpulkan bahwa responsivitas sistem merupakan faktor kritis dalam penerimaan pengguna terhadap aplikasi pelaporan berbasis mobile. Penelitian ini melangkah lebih jauh dengan mengintegrasikan inferensi dua model *deep learning* secara berurutan dalam satu alur respons API,

dan membuktikan bahwa pendekatan *client-server* dengan pemrosesan di sisi server merupakan strategi arsitektur yang layak secara fungsional meskipun masih memerlukan optimasi lebih lanjut pada sisi komputasi server untuk mencapai waktu respons yang lebih ideal dalam skenario penggunaan nyata.



BAB V

PENUTUP

A. Kesimpulan

Kerusakan jalan seperti lubang dan retakan mengancam keselamatan pengguna jalan dan menurunkan efisiensi transportasi. Metode pelaporan manual oleh petugas lapangan dinilai tidak efektif karena memakan waktu, menghasilkan data yang tidak konsisten, serta tidak memiliki sistem pemetaan visual yang memadai sehingga penanganan kerusakan menjadi lambat dan tidak terstandarisasi.

Untuk mengatasi permasalahan tersebut, penelitian ini merancang dan membangun sistem pelaporan kerusakan jalan berbasis Android yang mengintegrasikan model YOLOv12 dan ResNet-50 menggunakan arsitektur *client-server* berbasis Django REST API, dengan pengujian fungsional melalui *Black Box Testing* serta evaluasi performa model menggunakan metrik mAP, *precision*, *recall*, dan *F1-Score*.

Hasil penelitian menunjukkan bahwa sistem berhasil dibangun dan berjalan secara andal dengan tingkat keberhasilan pengujian fungsional sebesar 100% dari 26 skenario uji yang mencakup skenario normal, preventif, dan korektif. Model ResNet-50 pada konfigurasi ukuran input 800×800 dengan pelatihan selama 50 *epoch* mencapai performa klasifikasi terbaik dengan nilai *precision* sebesar 0,97, *recall* sebesar 0,96, dan *F1-Score* sebesar 0,97. Model YOLOv12 pada konfigurasi ukuran input yang sama dengan pelatihan selama 250 *epoch* menghasilkan performa deteksi terbaik dengan mAP50 sebesar 0,939 dan mAP50-95 sebesar 0,711. Rata-rata total waktu respons sistem dari 10 kali percobaan adalah 3.150,66 ms yang mencakup proses unggah gambar rata-rata 123,62 ms, inferensi ResNet-50 rata-rata 1.892,38 ms, dan inferensi YOLOv12 rata-rata 859,73 ms dalam kondisi jaringan lokal.

B. Saran

Berdasarkan hasil penelitian dan keterbatasan yang ditemukan, beberapa saran disampaikan untuk kesempurnaan penelitian serupa pada kesempatan selanjutnya. Penambahan dan diversifikasi dataset pelatihan sangat

direkomendasikan, terutama untuk kelas retak melintang dan memanjang yang konsisten menunjukkan kesalahan klasifikasi akibat kemiripan visual, disertai eksplorasi teknik augmentasi data yang lebih bervariasi seperti rotasi ekstrem dan transformasi perspektif. Selain itu, penambahan kategori jenis kerusakan yang lebih lengkap seperti kerusakan tepi perkerasan, amblesan, dan pelepasan butir disarankan agar sistem dapat mendeteksi kondisi kerusakan jalan yang lebih beragam sesuai standar klasifikasi yang berlaku di Indonesia.

Penelitian selanjutnya juga disarankan untuk mengintegrasikan kemampuan estimasi kedalaman kerusakan jalan, baik kedalaman lubang maupun lebar dan kedalaman retakan, melalui teknik *depth estimation* berbasis *monocular camera* atau sensor LiDAR pada perangkat *mobile*. Dengan informasi kedalaman tersebut, sistem dapat menghasilkan penentuan tingkat prioritas perbaikan yang lebih tepat dan objektif karena tidak hanya didasarkan pada jenis kerusakan, tetapi juga pada tingkat keparahan fisik kerusakan yang sesungguhnya di lapangan.