

IMPLEMENTASI SSD-MOBILENET DAN U-NET UNTUK DETEKSI DAN PENILAIAN TINGKAT KEPARAHAN PADA APLIKASI PELAPORAN JALAN BERLUBANG

Rahma Danu Sadewa¹, Yisti Vita Via^{2*}, Afina Lina Nurlaili³

^{1,2,3}Program Studi Informatika, Universitas Pembangunan Nasional Veteran Jawa Timur; Jl. Rungkut Madya No. 1, Gunung Anyar, Surabaya; Telp. (031) 8706369

Received: 6 September 2024

Accepted: 5 Oktober 2024

Published: 12 Oktober 2024

Keywords:

SSD-MobileNet;

U-Net;

Deep Learning;

Aplikasi Jalan Berlubang.

Correspondent Email:

yistivia.if@upnjatim.ac.id

Abstrak. Kondisi jalan rusak di Indonesia sangat memprihatinkan dan membutuhkan penanganan segera. Lubang-lubang di jalan yang menyerupai mangkuk dapat mengancam keselamatan pengguna jalan. Deteksi dan evaluasi kerusakan jalan berlubang bisa dilakukan dengan memanfaatkan teknologi kecerdasan buatan, seperti *deep learning*. Salah satu penerapan *deep learning* untuk tugas deteksi objek kompleks adalah Single Shot MultiBox Detector (SSD) yang memproses gambar dan menganalisis fiturnya menggunakan arsitektur MobileNet. Selain itu, tingkat keparahan kerusakan jalan berlubang dapat diidentifikasi menggunakan arsitektur U-Net yang berfungsi untuk segmentasi gambar dengan memprediksi pixel yang mewakili objek. Kombinasi metode SSD dan arsitektur MobileNet dengan U-Net menghasilkan analisis jumlah deteksi dan tingkat keparahan yang membantu pembuatan laporan kerusakan jalan berlubang dalam aplikasi. Pengujian beberapa gambar menunjukkan bahwa model SSD-MobileNet berhasil mendeteksi lubang di jalan dengan akurasi 93%, sementara model U-Net mencapai akurasi 80% dalam memprediksi tingkat keparahan pada jalan berlubang.

Abstract. The poor condition of roads in Indonesia is alarming and requires immediate attention. Potholes on roads which resemble bowls, pose a danger to road users. Detecting and evaluating road damage can be achieved using artificial intelligence technology, such as *deep learning*. One application of *deep learning* for complex object detection tasks is the Single Shot MultiBox Detector (SSD) which processes images and analyzes features using the MobileNet architecture. Additionally, the severity of road damage can be identified using the U-Net architecture which functions as an image segmentation model by predicting pixels representing the objects. The combination of SSD and MobileNet architecture with U-Net provides an analysis of the number of detections and severity levels, assisting in the creation of pothole damage reports in an application. Testing on several images demonstrated that the SSD-MobileNet model successfully detected road potholes with 93% accuracy, while the U-Net model achieved 80% accuracy in predicting the severity of the potholes.

1. PENDAHULUAN

Kondisi jalan yang rusak di Indonesia menjadi perhatian serius, terutama berdasarkan data Badan Pusat Statistik (BPS) tahun 2021 yang menunjukkan bahwa 31 persen atau

sekitar 175 ribu km dari total panjang jalan di Indonesia mengalami kerusakan, baik dalam kondisi rusak ringan maupun berat. Lubang-lubang pada jalan yang rusak, berbentuk seperti mangkuk dengan ukuran bervariasi, semakin

memperparah kondisi jalan karena air yang terkumpul dapat meresap ke dalam lapisan jalan dan mempercepat kerusakan [1]. Kerusakan ini tidak hanya merugikan secara ekonomi tetapi juga berpotensi membahayakan keselamatan pengguna jalan, sehingga memerlukan solusi yang cepat dan efektif.

Penilaian dan perbaikan jalan yang rusak membutuhkan sistem pelaporan yang memadai untuk mengidentifikasi kerusakan secara tepat. Sebelum perbaikan dapat dilakukan, tahap awal yang harus dilalui adalah melakukan penilaian kerusakan untuk mengetahui tingkat keparahannya. Selain itu, diperlukan survei lalu lintas untuk memahami karakteristik lalu lintas di area yang rusak [2]. Teknologi kecerdasan buatan, khususnya *deep learning*, dapat menjadi solusi inovatif untuk masalah ini. *Deep learning* adalah cabang dari *machine learning* yang memungkinkan komputer belajar dari data secara otomatis untuk mengenali pola-pola kompleks dalam citra atau gambar dan menggunakan jaringan saraf tiruan dengan lapisan yang banyak [3, 15]. Dengan menggunakan perangkat sehari-hari seperti kamera dan ponsel cerdas, *deep learning* dapat diterapkan dalam penilaian kerusakan jalan melalui aplikasi khusus.

Penelitian sebelumnya telah mengembangkan aplikasi pelaporan jalan rusak yang menghasilkan aplikasi Android untuk memudahkan masyarakat melaporkan kondisi jalan yang rusak kepada Dinas Pekerjaan Umum (PU) [4]. Penelitian serupa mengembangkan aplikasi untuk pendataan jalan rusak karena kurangnya sistem pendataan yang komprehensif, termasuk bukti gambar jalan rusak [5]. Namun, penelitian-penelitian ini belum memanfaatkan teknologi *deep learning* untuk mendeteksi dan menilai tingkat keparahan kerusakan jalan berlubang secara otomatis.

Berdasarkan penelitian sebelumnya, terlihat adanya kekurangan dalam pemanfaatan teknologi kecerdasan buatan untuk identifikasi dan penilaian keparahan kerusakan jalan berlubang. Oleh karena itu, penelitian ini bertujuan untuk mengatasi kekurangan tersebut dengan mengintegrasikan metode Single Shot MultiBox Detector (SSD) dan arsitektur MobileNet untuk mendeteksi objek pada citra jalan berlubang secara otomatis. MobileNet dipilih karena efisiensi dan kinerjanya yang

baik pada perangkat ponsel pintar [6]. Selain itu, segmentasi gambar dengan arsitektur U-Net akan digunakan untuk menilai tingkat keparahan kerusakan jalan. Penilaian tingkat keparahan mengikuti penelitian yang dilakukan di Korea Selatan dalam menilai keparahan dari jalan rusak menggunakan segmentasi dan deteksi objek *deep learning* [7]. Aplikasi ini diharapkan dapat memberikan laporan yang akurat dan cepat terkait kerusakan jalan berlubang dan tidak hanya membantu pengguna jalan untuk menghindari potensi kecelakaan, tetapi juga mempermudah pihak berwenang dalam memantau dan memperbaiki kerusakan jalan.

2. TINJAUAN PUSTAKA

2.1. Jalan Berlubang

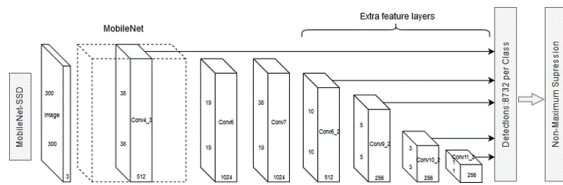
Jalan berlubang adalah kerusakan pada permukaan jalan yang terjadi akibat proses kimiawi dan mekanis pada lapisan perkerasan [1]. Bentuk cekungannya dapat membuat pengemudi kehilangan kendali, sehingga meningkatkan risiko kecelakaan. Penyebab kerusakan ini meliputi perkerasan yang lembab, siklus beku-leleh, kondisi lapisan dasar yang tidak memadai, beban lalu lintas yang berlebihan, atau kombinasi dari faktor-faktor tersebut [8].

2.2. Aplikasi Mobile Berbasis Android

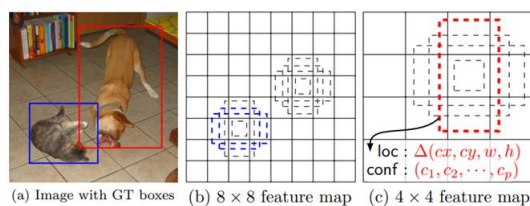
Android adalah sistem operasi berbasis Linux yang dikembangkan oleh Google untuk perangkat bergerak dengan layar sentuh, seperti ponsel pintar dan tablet, dan dirilis secara resmi pada tahun 2007 bersama dengan pembentukan Open Handset Alliance [9]. Android bersifat *open-source* dan menjadi platform komprehensif yang dirancang khusus untuk perangkat *mobile* [14]. Aplikasi Android merupakan perangkat lunak yang berjalan di sistem operasi ini, didukung oleh berbagai fitur bawaan. Dalam penelitian ini, fitur kamera dan GPS pada Android akan digunakan.

2.3. Single Shot MultiBox Detector

Single Shot MultiBox Detector (SSD) adalah metode *one-stage object detection* yang menggunakan *single deep neural network* untuk mendeteksi objek pada gambar. Jaringan SSD dimulai dengan lapisan ekstraksi fitur dari arsitektur klasifikasi citra seperti VGG, Inception, ResNet, atau MobileNet yang kemudian ditambahkan dengan lapisan fitur untuk deteksi multiskala [10].



SSD bekerja dengan memprediksi lokasi dan *confidence score* objek berdasarkan *default boxes* dengan berbagai rasio aspek pada *feature maps* [11].

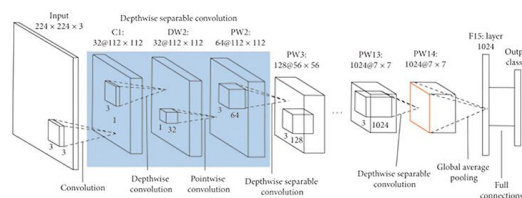


Gambar 2. *Default boxes* pada SSD

Dalam penelitian ini, SSD digabungkan dengan MobileNet untuk mendeteksi jalan berlubang.

2.4. MobileNet

MobileNet adalah model penglihatan komputer yang dirancang untuk efisiensi di perangkat mobile dengan sumber daya terbatas, menggunakan teknik seperti *depthwise separable convolution* untuk mengurangi jumlah parameter dan operasi komputasi [12]. *Depthwise separable convolution* terdiri dari *convolution filters* dan *point convolution filters*, di mana konvolusi dilakukan secara terpisah pada tiap input channel dan hasilnya digabungkan secara linear menggunakan konvolusi 1×1 .

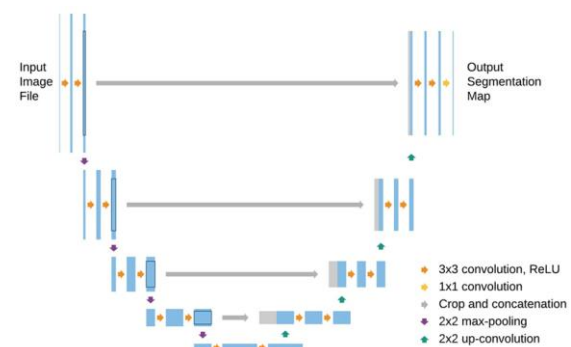


Gambar 3. Arsitektur MobileNet

Dalam penelitian ini, MobileNet digunakan untuk ekstraksi fitur pada gambar yang kemudian diproses lebih lanjut oleh metode Single Shot MultiBox Detector.

2.5. U-Net

U-Net adalah arsitektur *neural network* yang dirancang untuk segmentasi gambar, terdiri dari dua bagian: *contracting path* yang mirip dengan arsitektur CNN biasa dengan dua konvolusi 3×3 diikuti aktivasi ReLU dan *max-pooling*, serta *expansive path* yang melakukan *upsampling* pada *feature maps* menggunakan konvolusi 2×2 . *Feature maps* dari *contracting path* digabung dengan *feature map* yang sudah dinaikkan sample-nya, diikuti dua konvolusi 3×3 dan ReLU, serta konvolusi 1×1 untuk menghasilkan gambar tersegmentasi [13].

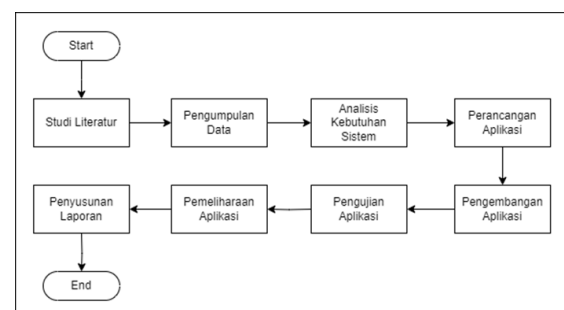


Gambar 4. Arsitektur U-Net

Dalam penelitian ini, U-Net digunakan untuk segmentasi gambar dan analisis tingkat keparahan jalan berlubang.

3. METODE PENELITIAN

Penelitian ini dimulai dengan studi literatur dan pengumpulan data yang dilakukan mengikuti panduan siklus hidup pengembangan sistem (SDLC) menggunakan metode waterfall. Langkah-langkah penelitian ini digambarkan pada Gambar 5.



Gambar 5. Metode Penelitian

3.1. Studi Literatur

Penelitian ini dimulai dengan studi literatur untuk menyelesaikan masalah yang dihadapi. Referensi mencakup deteksi jalan berlubang, metode deteksi objek, segmentasi gambar, pengukuran tingkat keparahan, serta pengembangan aplikasi pelaporan jalan rusak. Sumber yang digunakan berasal dari jurnal ilmiah, buku, dan artikel pendukung.

3.2. Pengumpulan Data

Data penelitian diperoleh dari dataset sekunder yang diakses melalui situs open-source Kaggle. Dataset pertama berisi 665 gambar dengan anotasi *bouding box* yang digunakan untuk melatih model deteksi objek jalan berlubang. Data ini terbagi menjadi 532 gambar untuk pelatihan dan 133 gambar untuk pengujian. Dataset kedua terdiri dari 780 gambar yang digunakan untuk segmentasi gambar, dengan pembagian 720 gambar untuk pelatihan dan 60 gambar untuk pengujian model segmentasi.

3.3. Analisis Kebutuhan Sistem

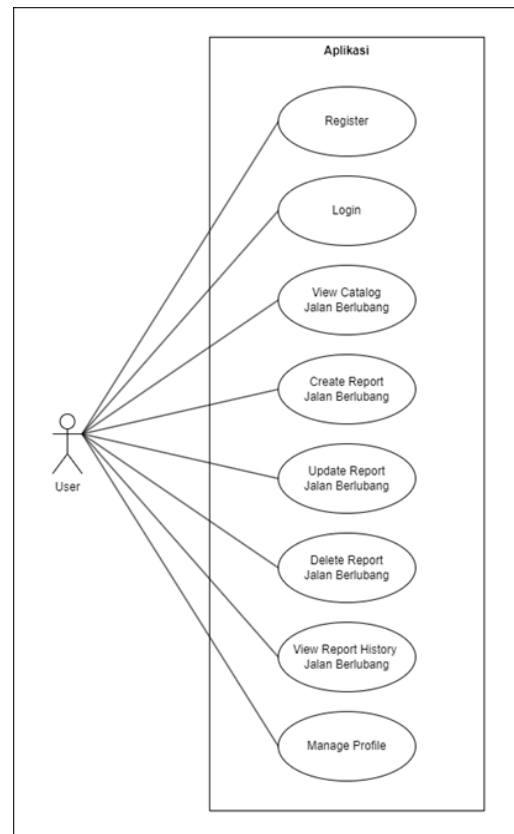
Analisis kebutuhan dilakukan untuk mendefinisikan persyaratan sistem dan perangkat yang diperlukan dalam pengembangan aplikasi. Tahap ini membantu mengklarifikasi kebutuhan untuk menyelesaikan penelitian sesuai dengan tujuan yang ditetapkan.

3.4. Perancangan Aplikasi

Tahap perancangan bertujuan untuk mengelola kompleksitas dalam pengembangan aplikasi, memastikan sistem yang aman dan mudah dipelihara. Desain aplikasi dibuat menggunakan alat bantu diagrams.net, termasuk pembuatan use case diagram dan class diagram.

3.4.1. Use Case Diagram

Use case diagram menggambarkan interaksi antara aktor dan fitur dalam sistem. Aktor dapat melakukan beberapa aksi, seperti: registrasi dan login untuk mengakses aplikasi, melihat katalog dan detail laporan jalan berlubang, membuat, memperbarui, dan menghapus laporan, melihat riwayat laporan yang telah dibuat, serta mengelola profil pengguna yang berisi informasi pribadi. Token authorization digunakan untuk keamanan saat login, dan laporan dapat mencakup judul, deskripsi, gambar dari kamera, serta lokasi yang diperoleh melalui GPS atau Google Maps.

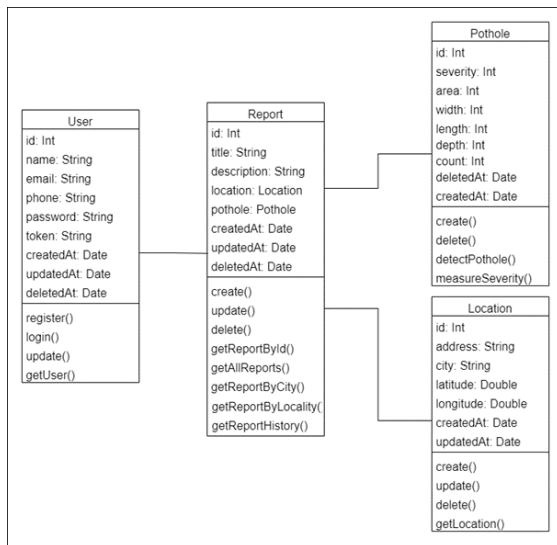


Gambar 6. Use Case Diagram

3.4.2. Class Diagram

Class diagram menunjukkan relasi antara objek yang terlibat dalam perancangan sistem aplikasi, di mana setiap objek memiliki atribut dan fungsi yang digunakan dalam fitur aplikasi. Dalam aplikasi pelaporan keparahan jalan berlubang, objek utama meliputi *user*, *report*, *pothole*, dan *location*.

Objek *user* digunakan untuk proses pendaftaran, login, dan pengelolaan profil. Objek ini memiliki relasi dengan objek *report* yang berfungsi untuk mencatat, memperbarui, dan menghapus laporan terkait jalan berlubang. *Report* berhubungan dengan *pothole* yang menyimpan informasi hasil analisis model *deep learning* dan *location* yang menyimpan data lokasi yang diambil dari Google Maps atau GPS. Atribut seperti *id* pada *report*, *city* pada *location*, dan *severity* pada *pothole* digunakan sebagai *identifier* untuk membantu proses pencarian dan pengurutan dalam operasi database.

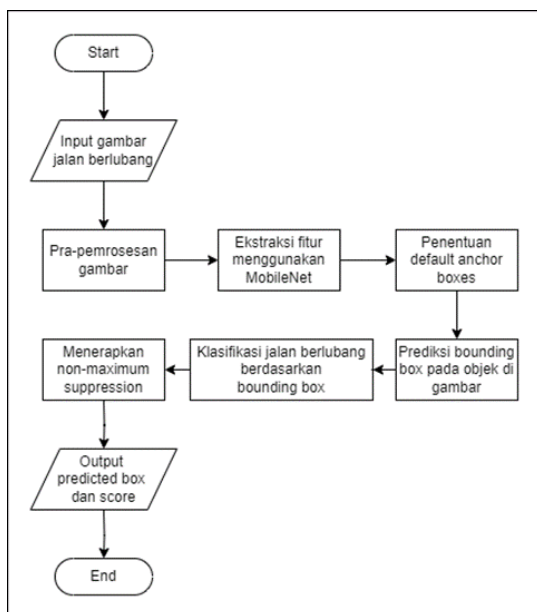


Gambar 7. Class Diagram

3.5. Pengembangan Aplikasi

Pengembangan aplikasi dilakukan menggunakan bahasa pemrograman Kotlin, JavaScript, dan Python. Android Studio digunakan untuk mengembangkan aplikasi di Android, sementara Visual Studio digunakan untuk pengembangan API menggunakan NodeJS dan Flask. Proyek dikelola menggunakan Git dan Google Drive untuk mengatur source code dan tugas.

3.5.1. Deteksi Objek SSD-MobileNet



Gambar 8. Alur Penerapan SSD-MobileNet

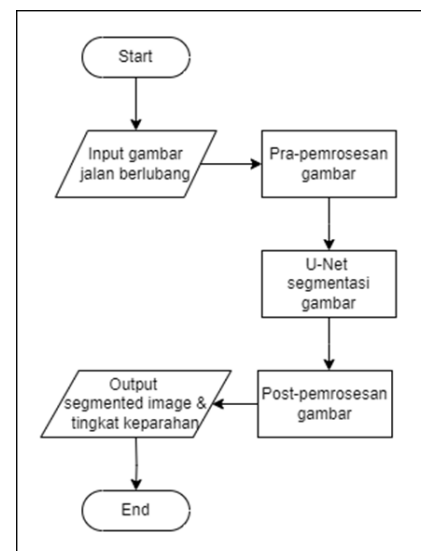
Aplikasi ini memanfaatkan algoritma deteksi objek yang menggabungkan metode Single Shot MultiBox Detector (SSD) dengan

arsitektur MobileNet. Proses dimulai dengan gambar yang diambil oleh pengguna untuk laporan jalan berlubang. Model kemudian melakukan pra-pemrosesan gambar, termasuk perubahan ukuran, rotasi, dan konversi ke bentuk tensor untuk meningkatkan kinerja dan variasi gambar.

MobileNet bertanggung jawab untuk ekstraksi fitur penting dari gambar, seperti *edges*, *texture*, dan *shape*. Hasil ekstraksi ini, berupa *feature maps*, kemudian digunakan oleh metode SSD. SSD menentukan *anchor boxes* atau *default boxes* pada setiap lokasi gambar berdasarkan perhitungan konvolusi dan memprediksi *bounding boxes* untuk mendeteksi potensi objek jalan berlubang.

Model selanjutnya mengklasifikasikan objek dalam gambar menjadi dua kategori: *background* atau jalan berlubang. Untuk mengurangi duplikasi dan *overlapping* pada *bounding boxes*, SSD melakukan *non-maximum suppression*. Hasil akhir, yaitu skor deteksi jalan berlubang yang ditampilkan kepada pengguna. Arsitektur MobileNet dipilih karena efisiensinya dan kemampuannya untuk bekerja dengan baik pada perangkat Android yang memiliki keterbatasan sumber daya komputasi.

3.5.2. Segmentasi Gambar U-Net



Gambar 9. Alur Penerapan U-Net

Model segmentasi gambar U-Net diterapkan untuk mengidentifikasi dan menilai keparahan objek jalan berlubang dari gambar yang diambil oleh pengguna. Segmentasi ini memungkinkan

penilaian tingkat keparahan jalan berlubang selama proses pengambilan gambar dan pembuatan laporan. Dalam aplikasi, model U-Net berjalan melalui Flask REST API untuk menerima *input* gambar dan mengembalikan hasil segmentasi.

Proses dimulai dengan aplikasi memperoleh gambar yang diambil oleh pengguna saat membuat laporan. Gambar tersebut kemudian di *pre-process*, termasuk pemotongan ukuran dan peningkatan kontras untuk meningkatkan kualitas dan variasi gambar. Setelah itu, model U-Net disimpan dalam format .h5 dan dijalankan dengan TensorFlow Keras di server Flask, melakukan segmentasi gambar.

Hasil segmentasi selanjutnya di *post-process*, di mana gambar yang tersegmentasi diubah menjadi format hitam-putih dan dikembalikan ke ukuran semula.

3.5.3. Penilaian Tingkat Keparahan

Luas objek tersegmentasi dihitung dengan memfilter hasil segmentasi menggunakan threshold nilai pixel lebih dari 0.5, lalu menghitung jumlah pixel yang bernilai 1. Rasio luas objek jalan berlubang dibandingkan dengan luas keseluruhan gambar dihitung untuk menentukan tingkat keparahan.

$$\text{Rasio (\%)} = \frac{\text{Luas Objek}}{\text{Luas Keseluruhan Gambar}} \times 100 \quad (1)$$

Tingkat keparahan ditentukan berdasarkan rasio luas yang dihitung, dengan kategori sebagai berikut: rendah jika rasio kurang dari 10%, sedang jika rasio antara 10% hingga 25%, dan tinggi jika rasio 25% atau lebih. Data mengenai luas objek dan tingkat keparahan ini kemudian ditampilkan kepada pengguna untuk laporan jalan berlubang setelah proses pengambilan gambar.

Rasio (%)	Tingkat Keparahan
$x < 10$	Rendah
$10 \leq x < 25$	Sedang
$25 \leq x$	Tinggi

Tabel 1. Kriteria Penilaian Tingkat Keparahan

3.6. Pengujian Aplikasi

Pengujian aplikasi menggunakan teknik black box untuk memastikan aplikasi berfungsi sesuai dengan kebutuhan yang dianalisis. Selain itu, model deteksi objek dan segmentasi gambar diuji untuk mengevaluasi performa sesuai ekspektasi.

3.7. Pemeliharaan Aplikasi

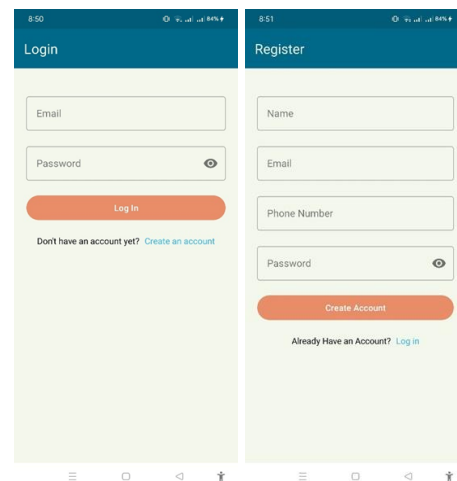
Pemeliharaan aplikasi meliputi perbaikan bug, optimasi performa, dan pemantauan *error* atau *crash*. Firebase Performance Monitoring digunakan untuk mengawasi performa aplikasi, sementara Firebase Crashlytics dipakai untuk memantau *error* atau *crash* secara *real-time*.

4. HASIL DAN PEMBAHASAN

4.1. Implementasi Aplikasi

4.1.1. Halaman Login

Saat aplikasi dibuka, halaman *login* akan ditampilkan seperti pada gambar 10. Pengguna dapat masuk dengan akun yang sudah ada atau memilih "Create an account" untuk mendaftar. Pendaftaran memerlukan nama, email, nomor telepon, dan password. Setelah login atau pendaftaran berhasil, pengguna akan diarahkan ke halaman *view catalog*.

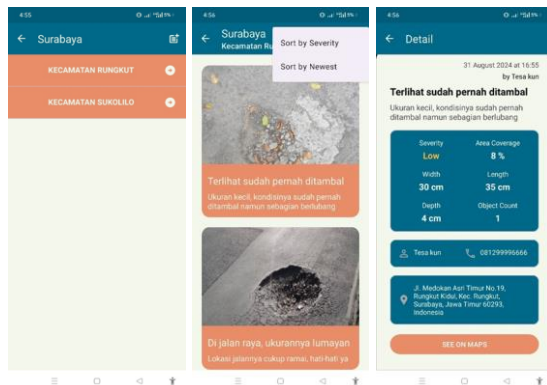


Gambar 10. Halaman Login & Register

4.1.2. Halaman View Catalog

Halaman *catalog* jalan berlubang pada gambar 11 menampilkan daftar laporan yang terfilter berdasarkan kecamatan dan kota. Pengguna dapat menyortir daftar berdasarkan tingkat keparahan atau tanggal laporan melalui tombol sortir di pojok kanan atas. Memilih salah satu laporan akan mengarahkan pengguna ke halaman detail yang mencakup foto, judul,

deskripsi, tanggal & waktu, pelapor, perkiraan luas, tingkat keparahan, dan lokasi. Tombol "See on maps" akan mengarahkan ke halaman peta yang menampilkan lokasi pin yang telah disematkan.

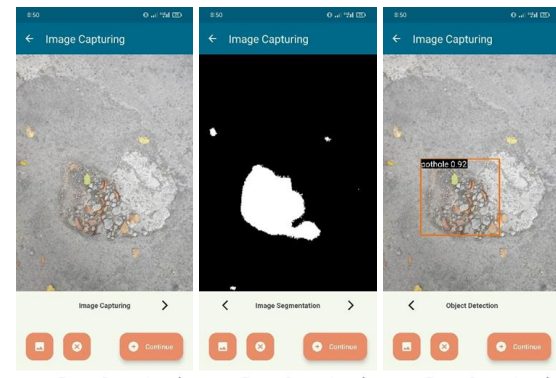


Gambar 11. Halaman *Catalog* Jalan Berlubang

4.1.3. Halaman *Create Report*

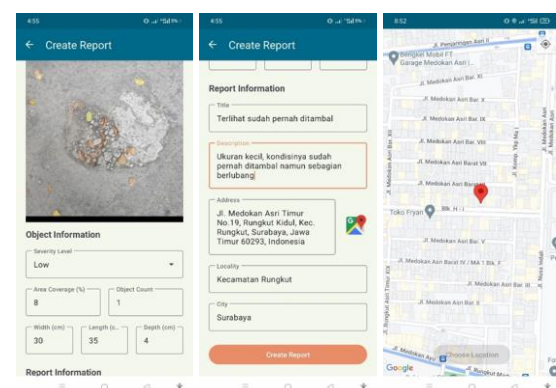
Halaman *create report* pada gambar 13 dapat diakses dari halaman *view catalog* atau *history report* melalui tombol "New Report" di pojok kanan atas. Pengguna akan diarahkan ke halaman pengambilan gambar, di mana mereka dapat memilih gambar dari kamera atau galeri. Tombol "Reset" memungkinkan pengguna mengganti gambar yang diambil. Setelah gambar diambil, aplikasi menampilkan indikator *loading* yang menunjukkan bahwa gambar sedang dianalisis. Gambar awal pada halaman ini menampilkan gambar asli, diikuti dengan gambar hasil segmentasi menggunakan U-Net, dan gambar hasil deteksi objek menggunakan SSD-MobileNet.

Setelah analisis selesai, hasil segmentasi dan deteksi objek ditampilkan dengan tombol panah kanan. Pengguna dapat memeriksa hasil analisis sebelum melanjutkan ke langkah berikutnya. Untuk melanjutkan pembuatan laporan, pengguna dapat menekan tombol "Continue". Proses ini memastikan bahwa data yang diambil akurat dan sesuai sebelum laporan final dikirimkan.



Gambar 12. Halaman Pengambilan Gambar

Pada halaman *create report*, aplikasi menunjukkan hasil analisis gambar, termasuk luas, tingkat keparahan, dan jumlah jalan berlubang yang terdeteksi. Pengguna dapat memperbarui informasi ini dan menambahkan data seperti lebar, panjang, kedalaman objek, serta judul, deskripsi, dan lokasi jalan berlubang. Untuk menambahkan lokasi, pengguna dapat menekan tombol "Google Maps" yang mengarahkan ke halaman maps dengan Google Maps API. Di halaman maps, pengguna dapat menyematkan pin lokasi dan menggunakan tombol "Current Location" untuk mencari lokasi saat ini. Setelah pin lokasi disematkan, pengguna menekan tombol "Choose Location" untuk mendapatkan informasi dari Google Maps API. Jika semua informasi lengkap, pengguna dapat menekan tombol "Create Report" untuk menyelesaikan pembuatan laporan.



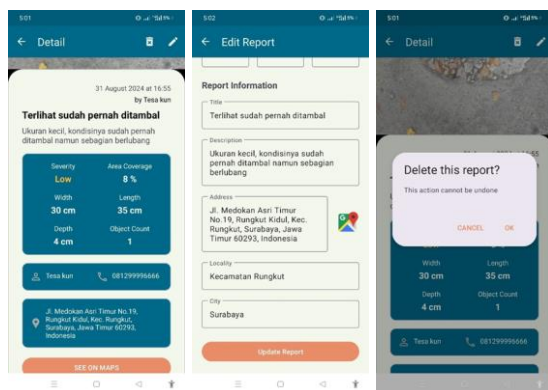
Gambar 13. Halaman *Create Report*

4.1.4. Halaman *Update Report*

Halaman detail *history report* pada gambar 14 merupakan langkah awal sebelum pengguna mengakses opsi untuk mengedit laporan dengan menekan tombol "Edit Report" di pojok kanan

atas. Halaman *update report* kemudian menampilkan kolom yang memungkinkan pengguna mengubah informasi seperti judul, deskripsi, alamat, kecamatan, dan kota. Setelah melakukan perubahan, pengguna dapat menekan tombol “Google Maps” untuk memperbarui lokasi dan tombol “Update Report” untuk menyimpan perubahan.

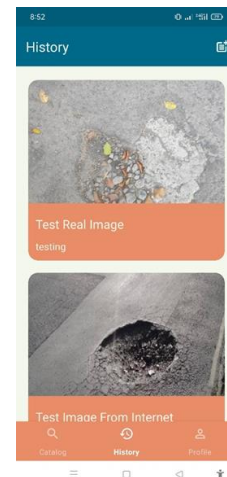
Pengguna juga memiliki opsi untuk menghapus laporan dengan menekan tombol “Delete Report” yang terletak di sebelah kiri tombol “Edit Report”. Aplikasi akan menampilkan dialog peringatan untuk mengonfirmasi penghapusan, dan pengguna harus menekan tombol “Ok” untuk menghapus laporan secara permanen.



Gambar 14. Halaman Detail *History* & *Update Report*

4.1.5. Halaman *History Report*

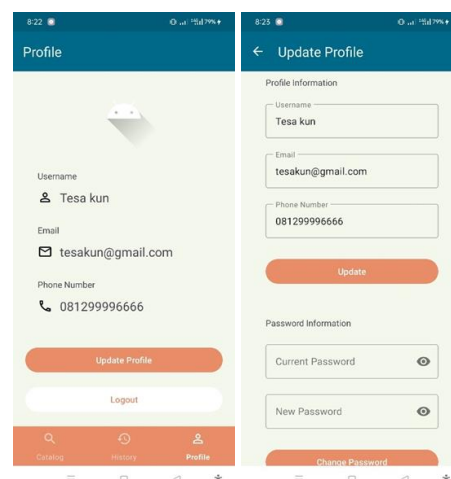
Riwayat laporan setiap pengguna disimpan di halaman *history report* pada gambar 15 dapat diakses melalui tombol navigasi di bagian bawah aplikasi. Pengguna dapat melihat rincian laporan dengan memilih salah satu dari daftar riwayat yang akan membawa mereka ke halaman detail *history report*. Di halaman ini, pengguna dapat melakukan tindakan lebih lanjut, seperti memperbarui informasi laporan atau menghapus laporan jika diperlukan. Halaman detail *history report* juga menyediakan akses mudah untuk melakukan perubahan atau penghapusan laporan.



Gambar 15. Halaman *History Report*

4.1.6. Halaman *Manage Profile*

Aplikasi menyediakan halaman untuk mengelola dan memperbarui informasi pengguna yang dapat diakses melalui tombol navigasi di bagian bawah utama seperti pada gambar 16. Halaman *manage profile* memungkinkan pengguna untuk melihat dan mengubah data akun. Pengguna dapat memperbarui informasi yang didaftarkan selama registrasi dengan menekan tombol “Update Profile,” yang juga memungkinkan perubahan kata sandi. Tombol “Logout” menghapus otorisasi pengguna dari aplikasi dan membuat pengguna untuk login kembali untuk melanjutkan akses.



Gambar 16. Halaman *Manage Profile*

4.2. Pengujian Model pada Aplikasi






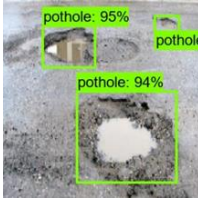
Pengujian model deteksi objek SSD-MobileNet menunjukkan akurasi 93%, sementara model segmentasi gambar U-Net

memperoleh akurasi 80% menggunakan data uji yang tidak digunakan selama pelatihan. Tabel 2 menyajikan hasil pengujian akurasi untuk model deteksi objek SSD-MobileNet dan segmentasi gambar U-Net.

Model	Jumlah Gambar	Benar	Akurasi
SSD-MobileNet	133	124	93%
U-Net	60	46	80%







Tabel 2. Pengujian Akurasi Model Pendukung Aplikasi

Tabel 3 menunjukkan hasil pengujian model SSD-MobileNet terhadap berbagai kondisi input gambar. Pengujian 1 hingga 4 mengevaluasi kemampuan model dalam menangani latar belakang dan pencahayaan yang bervariasi, latar belakang yang dekat dengan objek lain, dan kemampuan deteksi untuk lebih dari satu objek.

Ground Truth	Predicted	Score
		97%
		94%
		95%

Tabel 3. Pengujian Model SSD-MobileNet

Selain pengujian model SSD-MobileNet, model segmentasi gambar U-Net juga diuji seperti pada tabel 4. Pengujian ini menilai hasil prediksi segmentasi dan penilaian tingkat keparahan objek yang tersegmentasi.

Ground Truth	Predicted	Tingkat Keparahan
		Rendah
		Sedang
		Tinggi

Tabel 4. Pengujian Model U-Net

4.3. Pengujian Black Box

Pengujian dilakukan menggunakan metode black box seperti yang direncanakan. Tabel 5 mencakup tahapan pengujian yang termasuk skenario dan hasil pengujian.

No.	Skenario Uji	Hasil
1.	Melakukan login	Sesuai
2.	Melakukan registrasi	Sesuai
3.	Melihat daftar laporan jalan berlubang	Sesuai
4.	Melihat detail laporan jalan berlubang	Sesuai
5.	Mengambil gambar melalui kamera dan galeri	Sesuai
6.	Menjalankan deteksi	Sesuai

	objek menggunakan SSD-MobileNet pada gambar	
7.	Menjalankan segmentasi gambar menggunakan U-Net pada gambar	Sesuai
8.	Membuat laporan jalan berlubang	Sesuai
9.	Memperbarui laporan jalan berlubang	Sesuai
10.	Menghapus laporan jalan berlubang	Sesuai
11.	Melihat riwayat laporan jalan berlubang	Sesuai
12.	Melihat profil pengguna	Sesuai
13.	Memperbarui profil serta kata sandi pengguna	Sesuai

Tabel 5. Pengujian Black Box

5. KESIMPULAN

- Aplikasi pelaporan tingkat keparahan jalan berlubang yang dikembangkan menggunakan metode Single Shot MultiBox Detector dan arsitektur MobileNet efektif dalam melaporkan jalan berlubang dengan pemfilteran berdasarkan lokasi.
- Pengembangan aplikasi berhasil dilakukan menggunakan bahasa pemrograman Kotlin, JavaScript, dan Python, dengan Android Studio sebagai Integrated Development Environment (IDE) dan Visual Studio Code sebagai text editor.
- Penerapan metode Single Shot MultiBox Detector dengan arsitektur MobileNet serta U-Net memfasilitasi deteksi dan prediksi tingkat keparahan jalan berlubang,

dengan hasil pengujian model SSD-MobileNet mencapai akurasi 93% dan U-Net mencapai 80%.

UCAPAN TERIMA KASIH

Penulis ingin menyampaikan apresiasi yang mendalam kepada Dosen Pembimbing atas bimbingan dan arahan yang telah diberikan, sehingga penelitian ini dapat diselesaikan dengan baik. Terima kasih juga kepada orang tua dan teman-teman yang telah memberikan dukungan dan doa yang berharga sepanjang proses ini.

DAFTAR PUSTAKA

- [1] F. Yudaningrum Dan Ikhwanudin, "Identifikasi Jenis Kerusakan Jalan," *Teknika*, Vol. 12, No. 2, P. 8, 2017.
- [2] A. Z. Nashruddin Dan C. Buana, "Analisis Penilaian Kerusakan Jalan Dan Perbaikan Perkerasan Pada Jalan Raya Roomo, Kecamatan Manyar, Kabupaten Gresik," *Jurnal Teknik Its*, Vol. 10, No. 1, Pp. 27-34, 2021.
- [3] A. C. Jiantono, "Mengenal Deep Learning Beserta Contoh Penerapannya," 18 7 2023. [Online]. Available: <https://Sis.Binus.Ac.Id/2023/07/18/Mengenal-Deep-Learning-Beserta-Contoh-Penerapannya/>.
- [4] N. F. Iskandar, Z. Dan F. , "Pengembangan Aplikasi Pelaporan Jalan Rusak Di Kelurahan Bulupabbulu Kecamatan Tempe Kabupaten Wajo Berbasis Android," *Teknovokasi*, Vol. 1, No. 1, Pp. 25-33, 2023.
- [5] D. A. Kalengkongan, V. D. Kumenap Dan L. Sitanayah, "Aplikasi Online Pendataan Jalan Rusak Di Dinas Pu/Pr Bidang Bina Marga Minahasa Utara," *Jurnal Realtech*, Vol. 15, No. 1, Pp. 33-39, 2019.
- [6] M. F. Supriadi, E. Rachmawati Dan A. Arifianto, "Pembangunan Aplikasi Mobile Pengenalan Objek Untuk Pendidikan Anak Usia Dini," *Jurnal Teknologi Informasi Dan Ilmu Komputer (Jtiik)*, Vol. 8, No. 2, Pp. 357-364, 2021.
- [7] J. Ha, D. Kim Dan M. Kim, "Assessing Severity Of Road Cracks Using Deep Learning Based Segmentation And Detection," *The Journal Of Supercomputing*, Vol. 78, No. 16, Pp. 17721-17735, 2022.
- [8] T. Sita, "Penggunaan Material Cold Mix Asphalt Untuk Penanganan Penambalan Lubang," *Journal Of Airport Engineering Technology (Jaet)*, Vol. 1, No. 1, Pp. 24-29, 2020.
- [9] Y. Yudhanto Dan A. Wijayanto, Mudah Membuat Dan Berbisnis Aplikasi Android

- Dengan Android Studio, Jakarta: Pt Elex Media Komputindo, 2017.
- [10] Z. Munawar, M. Hasnawi, I. S. Beno, I. W. A. S. Darma, C. K. Sasatradipraja, N. P. Sutramiani Dan A. Amir, *Visi Komputer: Konsep, Metode, Dan Aplikasi*, Bandung: Kaizen Media Publishing, 2023.
- [11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Yang Fu Dan A. C. Berg, "Ssd: Single Shot Multibox Detector," P. 17, 2016.
- [12] M. S. Iswahyudi, I. Irmawati, J. A. Widiyans, G. S. Mahendra, M. Pratiwi, N. Hayati, S. Pormalingo, E. Miranda, W. Waryono Dan H. I. Yannuarsyah, *Aplikasi Machine Learning Di Berbagai Bidang*, Jambi: Pt Sonpedia Publishing Indonesia, 2023.
- [13] N. Siddique, S. Paheding, C. P. Elkin And V. Devabhaktuni, "U-Net And Its Variants For Medical Image Segmentation: A Review Of Theory And Applications," *Ieee Access*, Vol. 9, Pp. 82031-82057, 2021.
- [14] D. S. Farhani, Y. Sumaryana And T. Mufizar, "Pengembangan Aplikasi Media Pembelajaran Alat Musik Tradisional Berbasis Android Dengan Menggunakan Metode Multimedia Development Life Cycle (Mdlc)," *Jitet (Jurnal Informatika Dan Teknik Elektro Terapan)*, Vol. 12, No. 2, P. 12, 2024.
- [15] N. Khairunisa, C. And A. Jamaludin, "Analisis Perbandingan Algoritma Cnn Dan Yolo Dalam Mengidentifikasi Kerusakan Jalan," *Jitet (Jurnal Informatika Dan Teknik Elektro Terapan)*, Vol. 12, No. 3, P. 13, 2024.