

**DETEKSI RETAKAN JALAN MENGGUNAKAN KOMBINASI METODE
KONVOLUSI DAN *GAUSSIAN* NAÏVE *BAYES***

SKRIPSI

**Oleh :
FAKHAR MUHAMMAD HIDAYAT
NIM. 19650106**



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

**DETEKSI RETAKAN JALAN MENGGUNAKAN KOMBINASI METODE
KONVOLUSI DAN GAUSSIAN NAÏVE *BAYES***

SKRIPSI

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

Oleh :
FAKHAR MUHAMMAD HIDAYAT
NIM. 19650106

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

HALAMAN PERSETUJUAN

HALAMAN PERSETUJUAN

DETEKSI RETAKAN JALAN MENGGUNAKAN KOMBINASI METODE KONVOLUSI DAN GAUSSIAN NAÏVE BAYES

SKRIPSI

Oleh :
FAKHAR MUHAMMAD HIDAYAT
NIM. 19650106

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 19 November 2025

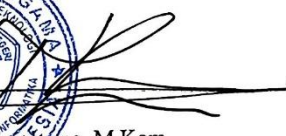
Pembimbing I,


Dr. Cahyo Crysdiar, M.Cs
NIP. 19740424 200901 1 008

Pembimbing II,


Tri Mukti Lestari, M.Kom
NIP. 19911108 202012 2 005

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



Supriyono, M.Kom
NIP. 19841010 201903 1 012

iii

HALAMAN PENGESAHAN

HALAMAN PENGESAHAN

DETEKSI RETAKAN JALAN MENGGUNAKAN KOMBINASI METODE KONVOLUSI DAN GAUSSIAN NAÏVE BAYES

SKRIPSI

Oleh :
FAKHAR MUHAMMAD HIDAYAT
NIM. 19650106

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 12 Desember 2025

Susunan Dewan Penguji

Ketua Penguji : Dr. Yunifa Miftachul Arif, M.T
NIP. 19830616 201101 1 004

Anggota Penguji I : Shoffin Nahwa Utama, M.T
NIP. 19860703 202012 1 003

Anggota Penguji II : Dr. Cahyo Crysdian, M.Cs
NIP. 19740424 200901 1 008

Anggota Penguji III : Tri Mukti Lestari, M.Kom
NIP. 19911108 202012 2 005

()
()
()
()

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



Supriyono, M.Kom
NIP. 19841010 201903 1 012

PERNYATAAN KEASLIAN TULISAN

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Fakhar Muhammad Hidayat

NIM : 19650106

Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika

Judul Skripsi : Deteksi Retakan Jalan Menggunakan Kombinasi Metode Konvolusi dan Gaussian Naïve Bayes.

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

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

Malang, 19 November 2025

Yang membuat pernyataan,



Fakhar Muhammad Hidayat
NIM.19650106

MOTTO

... Tanggung jawab dan selesaikan!...

HALAMAN PERSEMBAHAN

Puji syukur kehadiran Allah SWT karena telah memberikan rahmat dan petunjuk-Nya, sehingga saya mampu dan berhasil menyelesaikan skripsi ini. Saya persembahkan skripsi ini untuk kedua orang tua saya yang selalu memberikan doa dan dukungan penulisan skripsi saya. Saya juga berterima kasih kepada kedua saudara saya yang selalu memberikan motivasi dan semangat dalam menyelesaikan skripsi ini. Terima kasih yang tak terhingga bagi semua dukungan yang diberikan.

KATA PENGANTAR

Puji syukur Puji syukur penulis panjatkan kepada kehadiran Allah Subhanahu wa ta'ala yang telah melimpahkan nikmat serta karunia-Nya sehingga penulis mampu menyelesaikan penulisan Skripsi yang berjudul “Deteksi Retakan Jalan Menggunakan Kombinasi Metode Konvolusi dan *Gaussian Naïve Bayes*” dengan baik.

Dalam penulisan skripsi ini, penulis menyadari banyak pihak yang terlibat baik dalam proses membimbing penulisan dan juga memberikan semangat dan dukungan moril atau materiil. Penulis ingin menyampaikan terima kasih kepada:

1. Prof. Dr. Hj. Ilfi Nur Diana, M.Si., CAHRM., CRMP., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. H. Agus Mulyono, M.Si, selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Supriyono, M.Kom., selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Dr. Cahyo Crysdian, M.Cs., selaku Dosen Pembimbing I yang telah memberikan banyak dukungan dan bimbingan dalam penulisan skripsi ini.
5. Tri Mukti Lestari, M.Kom., selaku Dosen Pembimbing II yang telah memberikan banyak dukungan dan bimbingan dalam penulisan skripsi ini.
6. Dr. Yunifa Miftachul Arif, M.T dan Shoffin Nahwa Utama, M.T selaku dosen penguji yang telah memberikan banyak saran dalam menyelesaikan skripsi ini.

7. Segenap civitas akademik Program Studi Teknik Informatika, seluruh dosen yang telah memberikan ilmu serta arahan semasa kuliah.
8. Ayah Rochidajat, ibu Ani Karlina, kakak Rahadian Maulana Malik, dan adik Faris Muhammad Hidayat yang selalu memberikan dukungan, bantuan, serta doa yang tiada henti kepada penulis dalam menyelesaikan skripsi ini.
9. Baiq Aquillah Yasenda Anel Azhari, selaku rekan yang selalu membersamai selama masa pengerjaan skripsi. Terima kasih telah menemani, mendukung, memberikan bantuan, dan menghibur di masa-masa hilang harapan.
10. Sahabat dan teman-teman yang telah membantu dan memotivasi penulis saat pengerjaan skripsi.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat banyak kekurangan dan penulis berharap semoga skripsi ini dapat memberikan manfaat kepada pembaca khususnya penulis sendiri. *Amin Ya Rabbal Alamin.*

Wassalamu'alaikum Wr. Wb

Malang, 19 November 2025

Penulis

DAFTAR ISI

HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN.....	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
ABSTRAK	xiv
ABSTRACT	xv
البحث مستخلص.....	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Pernyataan Masalah	6
1.3 Batasan Masalah	6
1.4 Tujuan Penelitian	7
1.5 Manfaat Penelitian	7
BAB II STUDI PUSTAKA	8
2.1 Sistem Deteksi Retakan Jalan	8
BAB III DESAIN DAN IMPLEMENTASI	12
3.1 Pengumpulan Data	12
3.2 Desain Sistem.....	12
3.1.1 Preprocessing	13
3.1.2 Ekstraksi Fitur	16
3.1.3 <i>Gaussian Naïve Bayes</i>	17
3.2 Sistem Implementasi	20
3.2.1 Konfigurasi Pembagian Data	21
3.2.2 <i>K-Fold Cross Validation</i>	21
3.2.3 Visualisasi <i>Preprocessing</i> dan Ekstraksi Fitur	22
3.2.4 Evaluasi Model pada Data Uji	24
3.2.5 Prediksi Langsung.....	25
BAB IV UJI COBA DAN PEMBAHASAN	28
4.1 Skenario Pengujian	28
4.1.1. <i>Split data</i>	28
4.1.2. <i>K-Fold Cross Validation</i>	28
4.1.3. <i>Confusion Matrix</i>	30
4.2 Hasil Uji Coba.....	31
4.2.1. Hasil Uji Coba Skenario-1	31
4.2.2. Hasil Uji Coba Skenario-2	33
4.2.3. Hasil Uji Coba Skenario-3	35

4.3 Pembahasan.....	37
BAB V KESIMPULAN.....	44
5.1 Kesimpulan	44
5.2 Saran	44
DAFTAR PUSTAKA	
LAMPIRAN	

DAFTAR GAMBAR

Gambar 3.1 Desain sistem retakan jalan.....	12
Gambar 3.2 Ilustrasi grafik distribusi normal	18
Gambar 3.3 Konfigurasi pembagian data	21
Gambar 3.4 <i>K-Fold Cross Validation</i>	22
Gambar 3.5 Visualisasi tahap <i>preprocessing</i>	23
Gambar 3.6 Tabel ekstraksi fitur.....	23
Gambar 3.7 Kurva distribusi.....	24
Gambar 3.8 Evaluasi model	24
Gambar 3.9 Tampilan antarmuka pratinjau gambar.....	25
Gambar 3.10 Output dari prediksi langsung	26
Gambar 4.1 Ilustrasi <i>K-Fold Cross Validation</i>	29
Gambar 4.2 <i>Boxplot K-Fold Cross Validation</i>	38
Gambar 4.3 Metrik performa semua skenario	40

DAFTAR TABEL

Tabel 4.1 Pembagian dataset.....	28
Tabel 4.2 <i>Confusion matrix</i>	30
Tabel 4.3 Hasil <i>K-Fold Cross Validation</i> skenario-1	31
Tabel 4.4 <i>Confussion matrix</i> skenario-1	32
Tabel 4.5 Hasil performa skenario-1	32
Tabel 4.6 Hasil <i>K-Fold Cross Validation</i> skenario-2	33
Tabel 4.7 <i>Confussion matrix</i> skenario-2	34
Tabel 4.8 Performa skenario-2	35
Tabel 4.9 Hasil <i>K-Fold Cross Validation</i> skenario-3	35
Tabel 4.10 <i>Confussion matrix</i> Skenario-3	36
Tabel 4.11 Hasil performa skenario-3	36

ABSTRAK

Hidayat, Fakhar Muhammad. 2025. **Deteksi Retakan Jalan Menggunakan Kombinasi Metode Konvolusi dan Gaussian Naive Bayes**. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Cahyo Crysdian, M.Cs (II) Tri Mukti Lestari, M.Kom

Kata kunci: deteksi retakan, metode konvolusi, Gaussian Naive Bayes, klasifikasi citra.

Penelitian ini bertujuan untuk mengembangkan dan mengevaluasi sistem deteksi retakan jalan menggunakan kombinasi metode konvolusi dan klasifikasi *Gaussian Naive Bayes* (GNB). Pokok permasalahan yang diangkat adalah identifikasi retakan jalan yang efisien dan akurat guna mendukung pemeliharaan infrastruktur. Penelitian ini menggunakan *Road Surface Classification Dataset* (RSCD) yang terdiri dari 250 citra jalan beton beresolusi 360x240 piksel, dengan proporsi seimbang antara 125 citra retak dan 125 citra normal. Proses penelitian meliputi akuisisi citra, segmentasi, preprocessing menggunakan filter *Sobel* untuk mengekstraksi fitur tepi dan erosi untuk menyempurnakan representasi retakan. Fitur-fitur kuantitatif berupa banyak piksel hitam dan panjang tepi retakan kemudian diekstraksi dari hasil preprocessing. Model klasifikasi GNB selanjutnya dilatih dan dievaluasi dalam tiga skenario pembagian data: 70:30, 80:20, dan 90:10. Hasil penelitian menunjukkan bahwa skenario terbaik adalah pembagian data 90:10, yang menghasilkan performa model paling optimal dengan akurasi 88%, presisi 91.67%, *recall* 84.62%, dan *F1-Score* 88%.

ABSTRACT

Hidayat, Fakhar Muhammad. 2025. **Deteksi Retakan Jalan Menggunakan Kombinasi Metode Konvolusi dan Gaussian Naive Bayes**. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Promotor: (I) Dr. Cahyo Crysdian, M.Cs (II) Tri Mukti Lestari, M.Kom

This research aims to develop and evaluate an automatic road crack detection system using a combination of convolution methods and Gaussian Naive Bayes (GNB) classification. The core problem addressed is the efficient and accurate identification of road cracks to support infrastructure maintenance. This study utilizes the Road Surface Classification Dataset (RSCD), consisting of 250 concrete road images with a resolution of 360x240 pixels, with a balanced proportion of 125 *cracked* images and 125 normal images. The research process includes image acquisition, segmentation, preprocessing using the *Sobel* filter for edge feature extraction and erosion to refine crack representation. Quantitative features, namely the number of black pixels and crack edge length, are then extracted from the preprocessed results. The GNB classification model is subsequently trained and evaluated across three data split scenarios: 70:30, 80:20, and 90:10. The research findings indicate that the 90:10 data split scenario yields the most optimal model performance, achieving an accuracy of 88%, precision of 91.67%, *recall* of 84.62%, and an *F1-Score* of 88%.

Key words: crack detection, convolution method, Gaussian Naive Bayes, image classification.

البحث مستخلص

:البحث الجامعي. قسم هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرفون. الدكتور جاهيو كريسيديان، الماجستير؛ (2) تري موكتي ليستاري، الماجستير (1).

الكلمات المفتاحية: كشف التشققات، طريقة الالتفاف، غاوس بايز الساذج، تصنيف الصور.

يهدف هذا البحث إلى تطوير وتقييم نظام كشف تشققات الطريق باستخدام الجمع بين طريقة الالتفاف وتصنيف غاوس. تكمن المشكلة الرئيسية المطروحة في تحديد تشققات الطريق بشكل فعال ودقيق لدعم صيانة البنية التحتية. (GNB) بايز الساذج التي تتكون من 250 صورة لطريق خرساني بدقة (RSCD) يستخدم هذا البحث مجموعة بيانات تصنيف سطح الطريق، بكسل، بنسبة متوازنة بين 125 صورة متشققة و 125 صورة طبيعية. تشمل عملية البحث الحصول على الصور 360×240 لتحسين (Erosion) لاستخراج ميزات الحافة وعملية "التعرية" (Sobel) والتقسيم، والمعالجة المسبقة باستخدام مرشح "سويل" تمثيل التشققات. يتم استخراج الميزات الكمية المتمثلة في عدد البكسلات السوداء وطول حافة التشقق من نتائج المعالجة المسبقة. بعد وتقييمه في ثلاثة سيناريوهات لتقسيم البيانات: 70:30، و 80:20، و 90:10 GNB ذلك، يتم تدريب نموذج تصنيف أظهرت نتائج البحث أن السيناريو الأفضل هو تقسيم البيانات بنسبة 90:10، مما أدى إلى الأداء الأمثل للنموذج بدقة بنسبة 84.62٪، ودرجة (Recall) بنسبة 91.67٪، واستدعاء (Precision) تصل إلى 88٪، وإتقان (Accuracy) بنسبة 88 (F1-Score).

BAB I

PENDAHULUAN

1.1 Latar Belakang

Fasilitas jalan yang baik merupakan salah satu faktor penting dalam lalu lintas karena dapat memengaruhi konsentrasi, kenyamanan, dan keselamatan berkendara. Jalan yang rusak dapat mengakibatkan ketidaknyamanan bagi pengguna jalan, kerusakan pada kendaraan, bahkan dapat menimbulkan kecelakaan lalu lintas (Bhatt et al., 2017). Salah satu bentuk awal dari kerusakan ini adalah munculnya retakan pada permukaan jalan. Retakan kecil dapat dengan mudah berkembang menjadi lubang hanya karena hujan satu malam (Zou et al., 2019). Salah satu kecelakaan akibat retakan terjadi di Sulawesi Selatan pada akhir tahun 2019. Pengendara motor terjatuh dan terluka akibat ban motor terjebak di dalam retakan (Ansar, 2019).

Kecelakaan lainnya terjadi pada truk ikan di Jember pada tahun 2024. Truk saat itu sudah melaju dengan hati-hati saat berada di tikungan yang diketahui ada retakan jalan. Selisih permukaan jalan yang dinilai cukup tinggi membuat truk yang melaju menjadi miring dan terguling (Prasetiyo, 2024). Hal ini menunjukkan bahwa retakan tidak hanya menurunkan kualitas berkendara, tetapi juga meningkatkan risiko kecelakaan bagi pengguna jalan. Selain itu, penyebab retakan pada jalan raya juga sangat kompleks termasuk perubahan suhu, infiltrasi air, serta beban kendaraan yang berlebihan (Gavilán et al., 2011).

Berdasarkan dua kejadian tersebut, perbaikan jalan seharusnya segera dilakukan untuk mengurangi risiko kecelakaan. Musim hujan dapat memperburuk kondisi jalan karena membuat retakan dan lubang jalan tertutup air sehingga tidak terlihat oleh pengendara. Celah kecil pada retakan jalan juga dapat mengakibatkan pengendara motor tergelincir sehingga membahayakan pengendara lainnya. Identifikasi retakan jalan menjadi hal yang sangat penting untuk mencegah kerusakan lebih lanjut serta menjaga keselamatan jalan secara berkelanjutan (Yadav et al., 2024).

Pada umumnya, survei kondisi kerusakan jalan di Indonesia masih menggunakan metode konvensional dengan perangkat ukur sederhana seperti penggaris atau pita ukur, serta bergantung pada observasi visual teknisi untuk mencatat kerusakan (PUPR, 2016). Proses ini sering kali berlangsung lama dalam perbaikan sehingga memperburuk kondisi jalan dan meningkatkan risiko kecelakaan. Metode inspeksi manual seperti ini terbukti kurang efisien, membutuhkan banyak sumber daya, dan sulit memenuhi pemantauan jalan skala besar dengan frekuensi tinggi (Xu et al., 2024). Selain itu, efektivitas dari inspeksi manual juga tergantung dari pengalaman dan keahlian tiap orang yang berbeda-beda. Oleh karena itu, deteksi kerusakan jalan sangat penting agar proses perbaikan dan pemeliharaan jalan dapat dilakukan dengan segera (Kusumaningrum *et al.*, 2022).

Metode konvolusi adalah metode yang paling umum digunakan dalam sistem deteksi retakan. Konvolusi diterapkan menggunakan *kernel* atau *filter* (matriks kecil dengan nilai-nilai tertentu) yang dioperasikan pada setiap piksel pada

citra. Penelitian sebelumnya dilakukan oleh Punjabi *et al.*, (2008). dengan membuat sistem pintar untuk deteksi lubang jalan menggunakan *Sobel Operator* dan *Prewitt*. Hasil penelitian tersebut menunjukkan bahwa saat *preprocessing* metode *Prewitt* lebih akurat dalam mengurangi noise dalam deteksi tepi dibandingkan dengan *Sobel Operator* sehingga metode *Prewitt* digunakan sebagai input pada proses penelitian. Talab *et al.*, (2016) juga melakukan penelitian untuk deteksi retakan pada struktur beton menggunakan kombinasi *Sobel Operation* dan *Otsu Method*. Hasil dari kombinasi metode ini menunjukkan bahwa retakan dapat dipisahkan dengan baik dari objek lain di sekitar dan memiliki noise paling sedikit dibandingkan dengan metode lainnya.

Selain itu, Dorafshan *et al.* melakukan penelitian untuk membandingkan antara *deep learning* dengan metode deteksi tepi untuk gambar retakan pada beton (Dorafshan, Thomas and Maguire, 2018). Metode deteksi tepi menggunakan *Roberts*, *Prewitt*, *Canny*, *Sobel Operator*, *Laplacian of Gaussian (LoG)*, *Gaussian* dan *Butterworth* dengan *True Positive Rate (TPR)* mencapai 79%. Sedangkan metode *Deep Convolutional Network (DCNN)* dalam mode *Transfer Learning (TL)* mencapai TPR 86% dengan waktu yang lebih singkat dari seluruh metode deteksi tepi. Hal ini menunjukkan bahwa metode *deep learning* jauh lebih unggul dibandingkan metode deteksi tepi tradisional dalam hal akurasi dan kecepatan mendeteksi retakan halus. Meskipun algoritma *deep learning* seperti *Convolutional Neural Networks (CNN)* memberikan performa superior dalam ekstraksi fitur dan klasifikasi citra, pendekatan ini membutuhkan sumber daya komputasi yang tinggi dan dataset yang besar untuk pelatihan data (Cha *et al.*, 2017). Algoritma

pengolahan citra tradisional seperti deteksi tepi memang rentan terganggu oleh perubahan pencahayaan, keberadaan noda, sehingga menghambat akurasi dalam mendeteksi retakan (Wang et al., 2024).

Namun disisi lain, ketiga penelitian sebelumnya masih menggunakan algoritma pengolahan citra tradisional seperti deteksi tepi yang memiliki banyak kekurangan untuk deteksi retakan jalan. Penelitian yang dilakukan oleh Nnolim, (2020) berhasil melampaui akurasi *deep learning* dengan mengusulkan kombinasi *Canny Operator* dan segmentasi berbasis kontur aktif serta pengklasifikasian area berbasis saturasi. Penelitian tersebut menggunakan dua metode yang diusulkan, yaitu *Fusion of Edge Detection and Active Contour Segmentation* serta algoritma *HSI-based Crack Detection*. Hasil penelitian menunjukkan bahwa algoritma yang diusulkan menghasilkan presisi terbesar sampai 99% dibandingkan dengan algoritma *deep learning* lainnya. Akan tetapi, kompleksitas komputasi dan waktu pemrosesan yang lama menjadi kelemahan penelitian tersebut. Kekurangan dari seluruh penelitian sebelumnya dapat diatasi dengan mengintegrasikan antara algoritma pengolahan citra tradisional dengan *machine learning*.

Penelitian ini akan menggunakan kombinasi metode konvolusi untuk ekstraksi fitur dan *Gaussian Naïve Bayes* (GNB) sebagai model klasifikasi. Metode GNB dipilih karena secara khusus dirancang untuk menangani data kontinu dengan mengasumsikan distribusi *Gaussian* (Islam et al., 2022). Karakteristik tersebut relevan karena fitur pada citra yang dihasilkan dianggap sebagai data kontinu karena memiliki rentang distribusi yang besar. Selain itu, keunikan metode ini adalah menghitung probabilitas tiap kelas dengan asumsi bahwa semua kelas adalah

independen satu sama lain (Nugroho & Subanar, 2013). Metode *Naïve Bayes* juga memiliki kelebihan karena tidak memerlukan banyak data dan mudah diimplementasikan dengan akurasi yang cukup baik (Surya & Yamasari, 2024).

Menurut nilai-nilai ajaran islam, penelitian ini sejalan dengan nilai-nilai kebaikan dan kepedulian terhadap keselamatan sesama dalam mengurangi risiko kecelakaan di jalan raya. Sebagaimana dalam riwayat Muslim dari Abu Hurairah, Rasulullah shallallahu ‘alaihi wasallam bersabda:

مَرَّ رَجُلٌ بِعُصْنِ شَجَرَةٍ عَلَى ظَهْرِ طَرِيقٍ، فَقَالَ: وَاللَّهِ لَأُخَيِّرَ هَذَا عَنِ الْمُسْلِمِينَ لَا يُؤْذِيهِمْ فَأَدْخَلَ الْجَنَّةَ

“Dikisahkan ada seorang pria melewati dahan sebuah pohon di badan jalan. Ia lantas berkata, ‘Demi Allah, aku akan menyingkirkan dahan ini agar tidak menghalangi kaum Muslimin.’ Berkat amal itu, ia dimasukkan ke surga.” (H.R Muslim).

Berdasarkan hadits yang disebutkan diatas, menggambarkan tindakan sederhana dari seorang pria yang menyingkirkan dahan di jalan sehingga mendapatkan ridha Allah swt untuk dimasukkan ke surga. Dalam konteks penelitian “Deteksi Retakan Jalan Menggunakan Kombinasi Metode Konvolusi dan *Gaussian Naïve Bayes*”, hadits ini dapat dihubungkan dengan pemanfaatan teknologi untuk deteksi retakan jalan guna mengurangi risiko terjadinya kecelakaan lalu lintas. Selain itu, Allah juga berfirman dalam QS: al An’am ayat 11:

قُلْ سِيرُوا فِي الْأَرْضِ ثُمَّ أَنْظِرُوا كَيْفَ كَانَ عَاقِبَةُ الْمُكْذِبِينَ

“Katakanlah: "Berjalanlah di muka bumi, kemudian perhatikanlah bagaimana kesudahan orang-orang yang mendustakan itu" (Q.S al-An’am: 56).

Menurut tafsir Kementrian Agama RI, perintah ini mendorong manusia untuk menelaah jejak dan kisah masa lalu dengan pikiran yang jernih guna memahami sebab-akibat. Perintah imperatif untuk melakukan perjalanan investigatif (سِيرُوا) dan dilanjutkan dengan observasi kritis (انظُرُوا) terhadap

konsekuensi atau kesudahan dari suatu kondisi. Dalam konteks penelitian deteksi retakan jalan, proses pengumpulan data retakan jalan merupakan manifestasi modern dari perintah yang merekam kondisi aktual di lapangan. Selanjutnya, tahapan analisis data citra menggunakan metode konvolusi dan kecerdasan buatan untuk mengidentifikasi, mengklasifikasikan, dan memetakan pola retakan adalah representasi dari tindakan. Tujuan utamanya adalah untuk memahami indikator kerusakan pada tahap paling awal sehingga dapat mencegah terjadinya 'āqibah (konsekuensi) yang merugikan pengguna jalan.

Oleh sebab permasalahan tersebut dilakukanlah penelitian untuk mengidentifikasi retakan jalan dengan kombinasi metode konvolusi dan *Gaussian Naïve Bayes*. Objek yang diteliti pada penelitian ini yaitu citra gambar dari dataset *Road Surface Classification Dataset* (RSCD). Hasil dari penelitian ini diharapkan dapat mencapai akurasi yang tinggi dan bermanfaat untuk mendeteksi retakan jalan sehingga dapat mempermudah dalam mencatat informasi terkait kerusakan jalan.

1.2 Pernyataan Masalah

Seberapa baik performa kombinasi metode konvolusi dan *Gaussian Naïve Bayes* dalam melakukan deteksi retakan jalan?

1.3 Batasan Masalah

1. Data yang digunakan dalam penelitian ini adalah citra gambar *Road Surface Classification Dataset* (RSCD).
2. Material dataset yang diteliti adalah tipe jalan beton.

1.4 Tujuan Penelitian

Mengukur performa kombinasi metode konvolusi dan *Gaussian Naïve Bayes* dalam melakukan deteksi retakan jalan.

1.5 Manfaat Penelitian

1. Membantu Kementerian PUPR dalam pemeliharaan jalan dengan lebih cepat dan efektif dibandingkan dengan inspeksi manual.
2. Penelitian ini dapat memberikan kontribusi terkait dengan penggunaan metode konvolusi dan *Gaussian Naïve Bayes*.

BAB II

STUDI PUSTAKA

2.1 Sistem Deteksi Retakan Jalan

Efektivitas dari *U-Net Deep learning* dievaluasi dalam penelitian deteksi retakan jalan (Chakurkar et al., 2024). Metode tersebut diintegrasikan dengan *EfficientNet encoder* untuk memperkaya kumpulan fitur untuk deteksi retakan jalan yang lebih rinci, mendalam, dan akurat. Hasil penelitian menunjukkan keberhasilan metode ini dengan tingkat akurasi mencapai 97.49% dan *Mean Intersection over Union* (MIU/IoU) sebesar 74,9% yang secara signifikan lebih tinggi dibandingkan dengan metode lainnya. Meskipun nilai IoU-nya bukan yang tertinggi, namun peningkatan presisi dan *recall* ini menunjukkan bahwa model yang diusulkan efektif dalam mengurangi *false positive* dan *false negative*. Hal ini menunjukkan bahwa metode tersebut menjadi pendekatan yang efektif untuk deteksi retakan jalan.

Model LETNet (*Locally Enhanced Transformer Network*) diusulkan untuk mendeteksi retakan jalan secara *end-to-end* (Xu et al., 2024). Model ini mengintegrasikan *Transformer* untuk menangkap fitur global dengan modul peningkatan lokal untuk memperbaiki detail fitur lokal. LETNet dirancang menggunakan arsitektur encoder-decoder, dengan modul tambahan seperti *Defect Rectification* dan *Efficient Upsampling* untuk meningkatkan akurasi deteksi retakan. Dataset yang digunakan dalam penelitian ini mencakup *CrackNJ156*, *CrackTree260*, dan *CrackLS315*, dengan hasil penelitian menunjukkan bahwa LETNet mencapai IoU sebesar 89,91% dan *recall* sebesar 89,49%, mengungguli

model seperti *DeepCrack* dan *SwinUNet*. Kelebihan utama LETNet terletak pada efisiensinya dalam mendeteksi retakan pada kondisi jalan yang kompleks, meskipun memerlukan waktu pelatihan yang lebih lama.

Model SCL-GCN (*Stratified Contrastive Learning Graph Convolutional Network*) diusulkan untuk melakukan deteksi retakan jalan (Feng *et al.*, 2023). Model ini dirancang dengan memanfaatkan fitur spasial dan geometris melalui pendekatan stratifikasi, yang meningkatkan kemampuan deskriptif terhadap detail retakan. Pembelajaran kontrasif digunakan untuk mempersempit jarak antar fitur multi-skala, menghasilkan deskripsi yang lebih baik terhadap struktur retakan. Dataset utama yang digunakan adalah *data point cloud* jalan yang diperoleh dari *Mobile Laser Scanning* (MLS), yang mencakup informasi 3D seperti ketinggian dan intensitas. Hasil penelitian menunjukkan bahwa SCL-GCN mencapai presisi 75,7%, *recall* 75,1%, dan *F1-Score* 75,2%, melebihi kinerja metode sebelumnya seperti *CrackGCN* dan SD-GCN. Keunggulan utama penelitian ini adalah efisiensi pelatihan yang tinggi, ketergantungan data yang rendah, dan kemampuan mendeteksi retakan secara lebih lengkap, meskipun terdapat kekurangan dalam menangani fitur geometris retakan yang tidak mencolok. Dengan inovasi ini, penelitian Feng *et al.* berkontribusi signifikan dalam mempercepat dan meningkatkan akurasi deteksi retakan untuk mendukung perawatan jalan yang lebih efisien.

Model *CrackSegAN* yang merupakan sebuah jaringan berbasis *Generative Adversarial Network* (GAN) diperkenalkan untuk melakukan segmentasi retakan jalan (Pan *et al.*, 2023).. Model ini dilengkapi dengan *generator* dan *discriminator*

yang bekerja bersama untuk menghasilkan peta retakan yang mendekati *ground truth*. Fungsi *loss* gabungan digunakan untuk memastikan bahwa peta prediksi memiliki jarak yang kecil dengan peta *ground truth*, sementara augmentasi data elastis diterapkan untuk mengatasi variasi transformasi gambar. Dataset *CrackForest* dan *Crack500* digunakan untuk evaluasi, dengan model *CrackSegAN* mencapai peningkatan *F1-Score* sebesar 1.91% pada *CrackForest* dan 1.01% pada *Crack500* dibandingkan dengan model lain. Keunggulan utama model ini adalah kemampuannya untuk mengatasi noise latar belakang, menghasilkan segmentasi yang lebih detail, serta memiliki generalisasi yang lebih baik terhadap data lapangan dengan *noise* kompleks. Meskipun demikian, kekurangan model ini adalah ketergantungannya pada dataset besar untuk pelatihan dan keterbatasan dalam mengaplikasikan metode *unsupervised* atau *semi-supervised* untuk deteksi retakan tanpa dataset berlabel besar.

Sebuah metode deteksi kerusakan jalan diusulkan dengan mengandalkan jaringan YOLOv8-LF yang telah dioptimalkan untuk mendeteksi retakan jalan berbentuk kompleks (Yi *et al.*, 2024). Model ini memperkenalkan inovasi berupa modul *Context Guided Block* (CG block) yang meningkatkan kemampuan ekstraksi fitur, serta penggunaan *GSCov* dan VoV-GSCSP yang efektif untuk mengurangi ukuran model sambil mempercepat proses komputasi. Eksperimen yang dilakukan dengan dataset RDD-2022 menunjukkan bahwa model YOLOv8-LF yang diusulkan tidak hanya mengurangi parameter model hingga 73,5%, tetapi juga menunjukkan peningkatan yang signifikan dalam akurasi (6,6%), *F1-Score* (4,3%), dan FPS (116) dibandingkan dengan model dasar YOLOv8. Model YOLOv8-LF

juga berhasil mendeteksi retakan dalam berbagai kondisi sulit seperti tekstur yang tidak jelas dan latar belakang yang kompleks, dengan tingkat efisiensi yang lebih tinggi dalam deteksi *real-time*. Selain itu, dibandingkan dengan model deteksi lainnya, seperti SSD, DETR, dan YOLOv7-tiny, YOLOv8-LF menunjukkan kinerja superior dalam hal akurasi deteksi sambil menjaga kompleksitas komputasi yang rendah.

BAB III

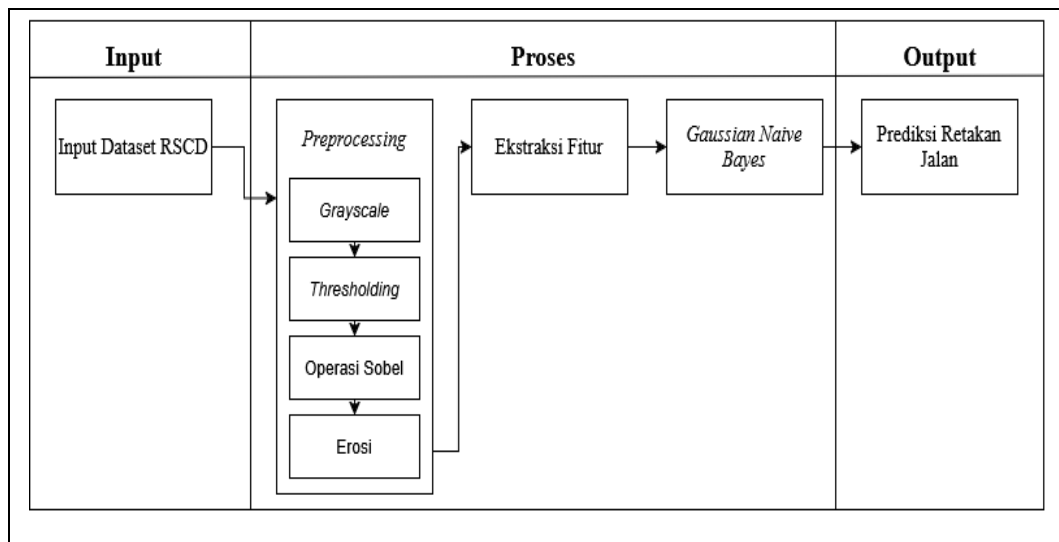
DESAIN DAN IMPLEMENTASI

3.1 Pengumpulan Data

Pada penelitian ini menggunakan dataset *Road Surface Classification Dataset* atau RSCD. Dataset ini bersifat publik yang menangkap gambar jalan dalam berbagai kondisi (Zhao *et al.*, 2023). Pengumpulan data berasal dari Beijing mulai Maret hingga Juli 2022. Setiap data mempunyai ukuran 360×240 pixel dengan jumlah total 250 data yang dibagi menjadi 125 retak dan 125 normal.

3.2 Desain Sistem

Pada penelitian ini akan mengimplementasikan desain sistem yang ditunjukkan pada gambar 3.1. Berdasarkan gambar tersebut, setiap subbab ini akan menguraikan tentang langkah-langkah yang akan dijalankan dalam deteksi retakan jalan menggunakan kombinasi metode konvolusi dan *Gaussian Naïve Bayes*.



Gambar 3.1 Desain sistem retakan jalan

3.1.1 Preprocessing

3.1.1.1 Grayscale

Penelitian ini akan dibangun dan diimplementasikan seperti pada gambar 3.1. Berdasarkan gambar tersebut, sistem ini dimulai dengan input citra jalan dari dataset yang telah disiapkan. Data ini berupa gambar berwarna yang kemudian akan melalui serangkaian tahapan *preprocessing* sebelum dapat dianalisis oleh model. Tahap awal *preprocessing* adalah konversi citra berwarna *Red Green Blue* (RGB) menjadi citra *grayscale* atau keabuan. Langkah ini bertujuan menyederhanakan data citra guna mengurangi kebutuhan penyimpanan dan biaya komputasi (Khudhair *et al.*, 2017).

Persamaan matematis yang umum digunakan untuk mendapatkan nilai derajat keabuan (Y) dari komponen merah (R), hijau (G), dan biru (B) diformulasikan sebagai berikut:

$$Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \quad (3.1)$$

Melalui perhitungan linear ini, setiap piksel yang direpresentasikan oleh vektor tiga dimensi ditransformasikan menjadi skalar satu dimensi yang merepresentasikan tingkat kecerahan dengan rentang nilai integer 0 (hitam) hingga 255 (Anh and Tam, 2022).

3.1.1.2 Thresholding

Setelah konversi ke *grayscale*, tahap selanjutnya adalah operasi *thresholding*. Tahap ini sangat penting karena bertujuan untuk melakukan segmentasi awal, yaitu memisahkan objek retakan (*foreground*) dari permukaan jalan (*background*) berdasarkan perbedaan intensitas piksel (Dong *et al.*, 2025).

Citra *grayscale* yang mengandung retakan akan diisolasi menggunakan metode *inverse binary thresholding* dengan nilai ambang batas (*threshold*) yang ditetapkan sebesar 90. Intensitas piksel yang nilainya di bawah ambang batas akan dipertahankan sebagai hitam (0), sedangkan piksel lainnya akan diubah menjadi putih (255). Hasil tahap ini adalah citra biner yang area retakan telah terisolasi secara kasar dari latar belakangnya.

3.1.1.3 Operasi *Sobel*

Selanjutnya, citra biner hasil segmentasi tersebut akan diproses menggunakan operasi konvolusi dengan kernel filter *Sobel*. Konvolusi adalah operasi matematika fundamental dalam pengolahan citra yang memodifikasi citra input dengan menerapkan kernel pada fungsi masukan (Sitinjak, 2020). Dalam penelitian ini, filter *Sobel* berfungsi untuk mendeteksi tepi dengan menghitung perbedaan intensitas dalam segmentasi sebelumnya (Grace *et al.*, 2025). Operasi *Sobel* dipilih karena perhitungannya yang relatif sederhana namun mampu mendeteksi perubahan gradien tepi secara signifikan (Supiyandi *et al.*, 2024).

Filter *Sobel* adalah sebuah matriks berukuran 3×3 yang didesain khusus untuk mengidentifikasi tepi horizontal dan vertikal. Operasi ini bekerja dengan menggerakkan dua matriks kernel pada setiap pixel di seluruh matriks pada citra (Hopkins, 2019). Rumus 3.1 dan 3.2 menunjukkan matriks G_x yang berfungsi untuk mendeteksi tepi vertikal sedangkan matriks G_y berfungsi mendeteksi tepi horizontal. Keduanya akan mengukur gradien atau perubahan intensitas yang tajam pada arah sumbu x dan y. Hasil dari kedua operasi ini kemudian digabungkan untuk

menghasilkan *magnitude gradien* total G , yang merepresentasikan kekuatan tepi pada setiap piksel. Berikut rumus untuk kernel dan menghitung *magnitude gradien*.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (3.2)$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (3.3)$$

$$|G| = \sqrt{G_x + G_y} \quad (3.4)$$

3.1.1.4 Erosi

Langkah terakhir dalam tahap *preprocessing* adalah operasi erosi. Meskipun *thresholding* berhasil segmentasi fitur tepi, sering kali citra biner yang dihasilkan masih mengandung noise berupa piksel tepi yang tidak relevan. Erosi adalah operasi dalam pemrosesan citra yang berfungsi untuk menghilangkan *noise* dan struktur kecil atau tipis (Fahmi *et al.*, 2023). Operasi erosi dilakukan pada citra biner yang tepi retakan direpresentasikan sebagai piksel putih (*foreground*) untuk menghilangkan noise, sebelum akhirnya divisualisasikan kembali sebagai piksel hitam.

Secara teknis, proses ini melibatkan konvolusi citra biner dengan sebuah *structuring element* atau kernel berukuran . Prinsip kerja erosi didasarkan pada teori himpunan, yaitu piksel pada koordinat hanya akan dipertahankan bernilai 1 (putih) jika elemen struktural yang ditempatkan pada posisi tersebut berada sepenuhnya di dalam himpunan objek (Ma *et al.*, 2020).

Dengan mekanisme ini, *noise* yang tidak dapat menampung kernel secara utuh akan diubah menjadi hitam, sementara struktur utama retakan yang lebih tebal dan relevan akan tetap dipertahankan meskipun mengalami sedikit penipisan. Hasil akhirnya adalah citra biner yang lebih bersih dan siap untuk diproses pada tahap ekstraksi fitur.

3.1.2 Ekstraksi Fitur

Setelah *preprocessing* selesai, hasil citra biner akan diekstrak fiturnya sebelum diproses di model klasifikasi. Citra biner yang hanya berisi tepi retakan akan diekstrak berdasarkan dua fitur, yaitu banyak piksel hitam (*black pixel count*) dan panjang tepi hitam (*black edge length*). Luas retakan direpresentasikan oleh total akumulasi piksel objek retakan, sedangkan panjang retakan direpresentasikan oleh jumlah piksel pada kerangka hasil penipisan morfologi atau skeletonisasi (Oliveira et al., 2013).

Secara teknis, perhitungan luas retakan dilakukan dengan mengakumulasi seluruh piksel yang bernilai 0 (hitam) pada citra biner hasil erosi. Namun, untuk menghitung fitur panjang retakan yang akurat, metode penjumlahan piksel biasa tidak dapat diterapkan karena ketebalan retakan yang bervariasi. Skeletonisasi bekerja dengan cara mengupas kontur objek secara iteratif hingga mencapai lebar satu piksel pada sumbu tengahnya (Fereidoon et al., 2016). Dengan menggunakan algoritma ini, area retakan yang tebal direduksi menjadi garis kerangka (*skeleton*) setipis 1 piksel. Panjang retakan kemudian dihitung dengan menjumlahkan total piksel yang membentuk garis kerangka tersebut.

Pentingnya pemilihan fitur ini didasarkan pada signifikansi teknisnya dalam penilaian kondisi jalan. Karakteristik retakan seperti jenis, jumlah, dan lebar dapat menjadi indikator awal tingkat kerusakan dan kemampuan struktur beton dalam menopang beban (Mohan & Poobal, 2018). Oleh karena itu, fitur panjang dan luas retakan akan menjadi *input numerik* yang merepresentasikan karakteristik setiap gambar yang akan diproses oleh model klasifikasi *Gaussian Naïve Bayes*.

3.1.3 *Gaussian Naïve Bayes*

Gaussian Naïve Bayes (GNB) merupakan salah satu varian dari algoritma *Naïve Bayes* yang digunakan untuk tugas klasifikasi probabilistik. Algoritma ini didasarkan pada teorema *Bayes* yang memprediksi peluang di masa yang akan datang berdasarkan pengalaman di masa sebelumnya (Hasanah et al., 2022). *Gaussian Naïve Bayes* secara khusus mengasumsikan bahwa fitur-fitur yang digunakan untuk klasifikasi mengikuti distribusi *Gaussian* atau distribusi normal. Distribusi tersebut mengasumsikan bahwa setiap fitur pada data tidak saling mempengaruhi atau independen dalam memprediksi target (Ha, 2023). Berikut rumus persamaan distribusi *Gaussian*:

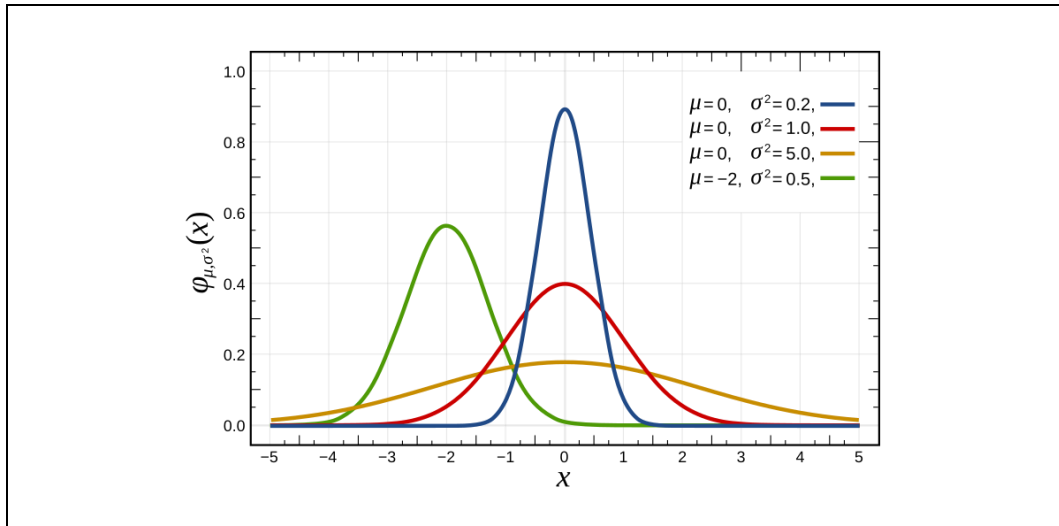
$$f(x) = \frac{1}{\sqrt{2\sigma\pi}} \exp^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (3.5)$$

dengan μ adalah rata-rata (*mean*) dan σ adalah standar deviasi (Nurul A'ayunnisa et al., 2022). Dengan rumus untuk mencari μ dan σ adalah sebagai berikut.

$$\mu = \frac{\sum_{i=1}^n x_i}{n} \quad (3.6)$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{n}} \quad (3.7)$$

Distribusi normal memiliki dua parameter utama, yaitu rata-rata (μ) dan standar deviasi (σ). Rata-rata menunjukkan nilai tengah dari distribusi, sedangkan standar deviasi mengindikasikan sejauh mana nilai-nilai tersebar di sekitar rata-rata (Kashishdafa, 2024). Jika digambarkan distribusi nilai dari setiap fitur dalam bentuk grafik, maka hasilnya akan membentuk pola kurva simetris seperti bentuk lonceng. Berikut ilustrasi grafik dari distribusi normal.



Gambar 3.2 Ilustrasi grafik distribusi normal

Gaussian Naïve Bayes (GNB) dalam penelitian ini digunakan untuk mengklasifikasikan gambar jalan menjadi *cracked* atau normal berdasarkan dua fitur, yaitu banyak piksel hitam dan panjang piksel hitam. Kedua fitur tersebut akan dilakukan perhitungan *mean* dan standar deviasi dalam proses pelatihan data pada setiap kelas. Nilai-nilai ini sangat penting karena akan digunakan untuk membentuk kurva distribusi *Gaussian* yang merepresentasikan karakteristik masing-masing fitur di setiap kelas.

Secara umum, arsitektur GNB terbagi menjadi tiga tahap utama: perhitungan probabilitas *prior*, perhitungan *likelihood*, dan perhitungan probabilitas *posterior*. Tiga tahapan ini diimplementasikan dalam proses prediksi kelas menggunakan data uji (*test data*). Keputusan akhir prediksi ditentukan dengan membandingkan nilai Probabilitas Posterior antara kelas *cracked* dan normal. Kelas yang menghasilkan nilai probabilitas posterior tertinggi dianggap sebagai label prediksi yang paling valid untuk citra tersebut.

Probabilitas *prior* adalah probabilitas awal suatu kejadian yang dihitung sebelum mendapatkan informasi baru. Dalam konteks penelitian ini, probabilitas *prior* adalah probabilitas munculnya kelas "*cracked*" atau "normal" secara independen dari fitur apa pun. Nilai ini dihitung dengan membagi jumlah sampel dari setiap kelas dengan jumlah total sampel. Rumus dari probabilitas *prior* adalah sebagai berikut.

$$\pi(\mu) = \sum_{j=1}^n \frac{x_j}{n} \quad (3.8)$$

Keterangan:

$\pi(\mu)$: distribusi *prior* untuk parameter μ

x_j : jumlah sampel pada kelas ke -j di dalam dataset pelatihan

n : jumlah total sampel di seluruh dataset pelatihan.

Likelihood adalah istilah yang menggambarkan seberapa mungkin suatu data (atau fitur) terjadi, diberikan suatu kelas atau kondisi tertentu. Fungsi *likelihood* menunjukkan seberapa besar kemungkinan data $x_1, x_2, x_3 \dots, x_n$ yang dianggap berasal dari distribusi normal dengan parameter nilai rata-rata μ dan varians σ^2 .

Likelihood dihitung menggunakan distribusi *Gaussian* sebagai berikut:

$$L(\mu) = \prod_{j=1}^n \frac{1}{\sqrt{2\pi\sigma_v}} \exp\left(-\frac{1}{2} \left(\frac{x_j - \mu}{\sigma_v}\right)^2\right) \quad (3.9)$$

Keterangan:

$\frac{1}{\sqrt{2\pi\sigma_v}}$: fungsi densitas probabilitas untuk distribusi normal
 x_j : data yang diamati dengan indeks data j
 μ : nilai rata-rata yang dicari
 σ_v : standar deviasi dari distribusi normal untuk data x_j

Evidence berperan penting dalam menentukan probabilitas *posterior* dengan cara menggabungkan *prior* dan *likelihood*. *Evidence* adalah normalisasi dari *likelihood* berdasarkan *prior* yang digunakan dalam teorema *Bayes* untuk menghitung probabilitas *posterior* suatu model (Trotta, 2008). Berikut adalah rumus matematis dari *evidence*.

$$P(x) = \sum_{\mu=1} \pi(\mu) L(\mu) \quad (3.10)$$

Keterangan:

$P(x)$: *evidence*
 $L(\mu)$: fungsi *likelihood* pada kelas ke μ
 $\pi(\mu)$: fungsi *prior* pada kelas ke μ

Probabilitas *posterior* adalah keyakinan yang diperbarui setelah menggabungkan probabilitas *prior* dengan data baru (*likelihood*). Pembaruan keyakinan probabilitas *posterior* menggunakan teorema *Bayes* sebagai berikut.

$$g(\mu|x) = \pi(\mu)L(\mu) = \sum_{j=1}^n \frac{x_j}{n} \prod_{j=1}^n \frac{1}{\sqrt{2\pi\sigma_v}} \exp\left(-\frac{1}{2} \left(\frac{x_j - \mu}{\sigma_v}\right)^2\right) \quad (3.11)$$

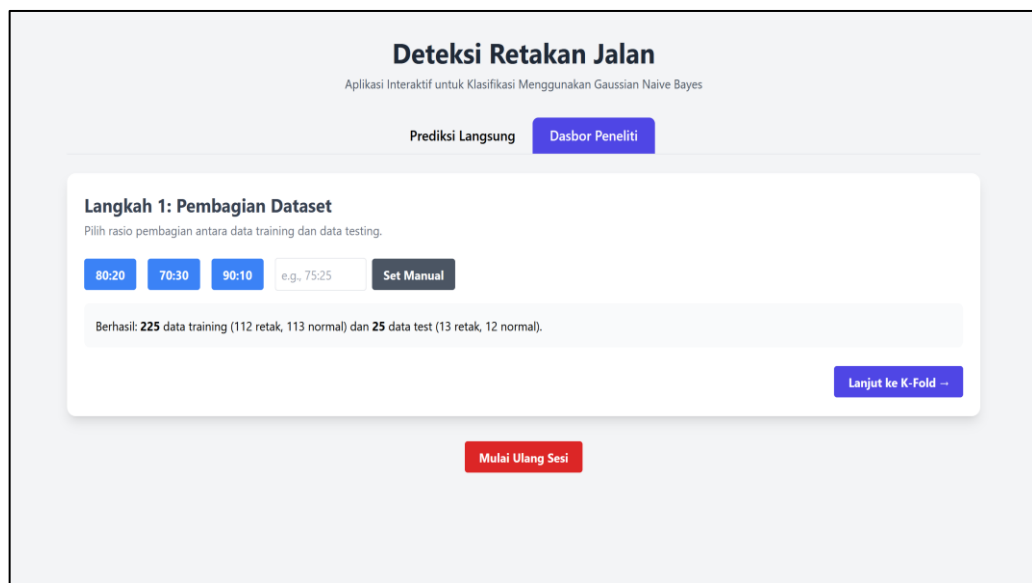
3.2 Sistem Implementasi

Implementasi sistem dibangun berbasis web menggunakan kerangka kerja Flask (Python) untuk *backend* dan antarmuka interaktif berbasis Javascript. Sistem ini terdiri dari dua modul utama, yaitu Dasbor Peneliti untuk eksperimen pelatihan

model dan Prediksi Langsung untuk pengujian satuan. Berikut adalah rincian antarmuka dari setiap tahapan proses.

3.2.1 Konfigurasi Pembagian Data

Tahap awal implementasi dimulai dengan pembagian dataset pada menu dasbor peneliti yang ditunjukkan Gambar 3.3. Sistem menyediakan antarmuka interaktif yang memungkinkan pengguna memilih skenario pembagian data (70:30, 80:20, atau 90:10) tanpa perlu mengubah kode sumber. Tahap ini secara otomatis memisahkan citra ke dalam himpunan latih dan himpunan uji sesuai proporsi yang dipilih, serta menampilkan jumlah distribusi kelasnya.



Gambar 3.3 Konfigurasi pembagian data

3.2.2 *K-Fold Cross Validation*

Setelah data terbagi, tahap selanjutnya sistem menjalankan validasi silang untuk mengukur stabilitas model. Implementasi tahap *K-Fold* ditampilkan Gambar 3.4. Tabel hasil validasi menyajikan metrik performa (akurasi, presisi, *recall*, *F1*-

Score) untuk setiap lipatan (*fold*). Fitur penting pada implementasi ini adalah perhitungan standar deviasi dan rata-rata setiap metrik performa yang berfungsi sebagai indikator konsistensi model dalam mengenali pola retakan pada berbagai variasi data latih. Terdapat *button* “Lanjut ke Pelatihan Final” untuk melanjutkan ke tahap berikutnya.



The screenshot shows a web application titled "Deteksi Retakan Jalan" (Crack Detection on Roads). It is an interactive application for classification using Gaussian Naive Bayes. The interface has two main buttons: "Prediksi Langsung" (Direct Prediction) and "Dasbor Peneliti" (Researcher Dashboard). The "Dasbor Peneliti" button is highlighted. Below the buttons, there is a section titled "Langkah 2: Validasi Silang (K-Fold)" (Step 2: Cross-Validation (K-Fold)). This section explains that the model is evaluated on training data using 5-Fold Cross-Validation to measure performance stability. A table titled "Hasil Validasi K-Fold (K=5)" (K-Fold Validation Results (K=5)) displays the performance metrics for each fold and the overall average. The metrics are Accuracy, Precision (Cracked), Recall (Cracked), and F1-Score (Cracked). The overall average accuracy is 89.78%, precision is 96.05%, recall is 83.36%, and F1-score is 89.23%. The standard deviation for each metric is also shown.

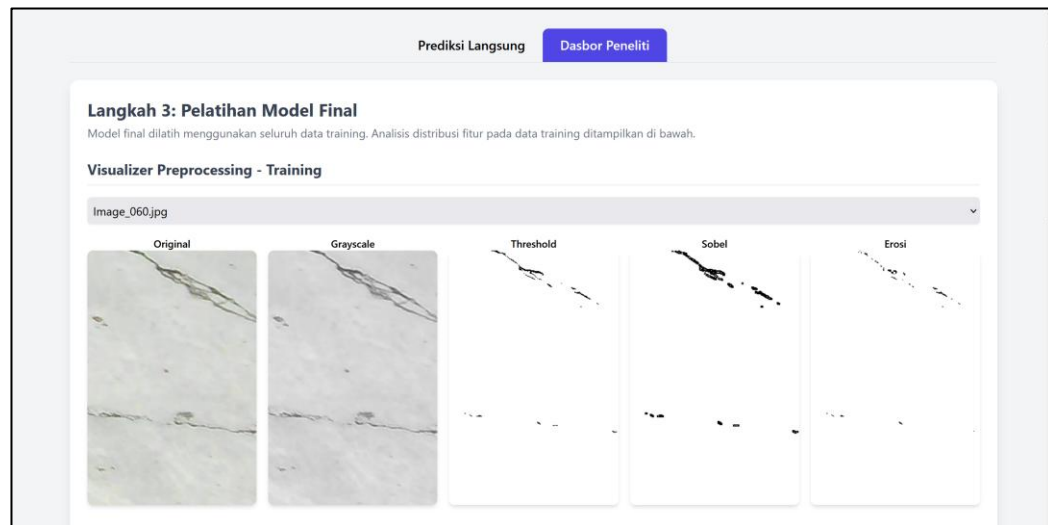
FOLD	AKURASI	PREKISI (CRACKED)	RECALL (CRACKED)	F1-SCORE (CRACKED)
1	84.44%	90.91%	80.00%	85.11%
2	91.11%	100.00%	82.61%	90.48%
3	86.67%	95.24%	80.00%	86.96%
4	95.56%	100.00%	90.00%	94.74%
5	91.11%	94.12%	84.21%	88.89%
Rata-rata	89.78%	96.05%	83.36%	89.23%
Standar Deviasi (±)	± 3.87%	± 3.52%	± 3.69%	± 3.29%

At the bottom right of the table, there is a button labeled "Lanjut ke Pelatihan Final --" (Continue to Final Training --).

Gambar 3.4 K-Fold Cross Validation

3.2.3 Visualisasi *Preprocessing* dan Ekstraksi Fitur

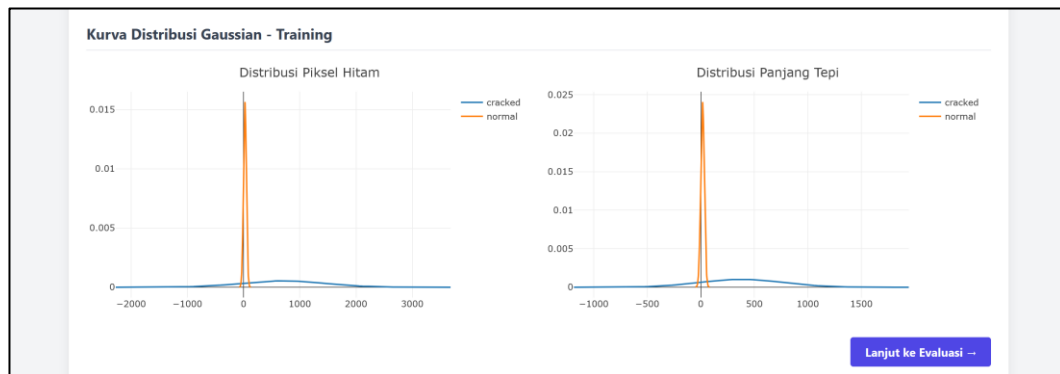
Tahap selanjutnya adalah *preprocessing* dan ekstraksi fitur. Pada tahap *preprocessing* perubahan citra dari bentuk asli (RGB), *grayscale*, *thresholding*, deteksi tepi *Sobel*, hingga hasil akhir erosi ditunjukkan Gambar 3.5. Semua visualiasi dapat diakses pada repositori digital: https://bit.ly/visual_dataset.

Gambar 3.5 Visualisasi tahap *preprocessing*

Selanjutnya, hasil dari proses tersebut dikuantifikasi menjadi nilai numerik yang ditunjukkan Gambar 3.6. Pada gambar tersebut menampilkan tabel ekstraksi fitur yang berisi nilai banyak piksel hitam dan panjang tepi hitam untuk setiap citra. Data ini kemudian digunakan untuk menghitung nilai rata-rata tiap kelas guna membentuk kurva distribusi probabilitas *Gaussian Naïve Bayes* yang ditunjukkan Gambar 3.7.

Tabel Fitur Training			
FILENAME	KELAS	PIKSEL HITAM	PANJANG TEPI
Image_134.jpg	normal	0.00	0.00
Image_014.jpg	cracked	497.00	248.00
Image_045.jpg	cracked	0.00	0.00
Image_004.jpg	cracked	782.00	534.00
Image_063.jpg	cracked	292.00	159.00
Image_110.jpg	cracked	1308.00	690.00
Image_060.jpg	cracked	208.00	108.00
Image_117.jpg	cracked	130.00	69.00
Image_085.jpg	cracked	2685.00	1335.00
Tabel Rata-rata Fitur Training			
KELAS	AVG. PIKSEL HITAM	AVG. PANJANG TEPI	
cracked	703.57	378.97	
normal	23.95	16.55	

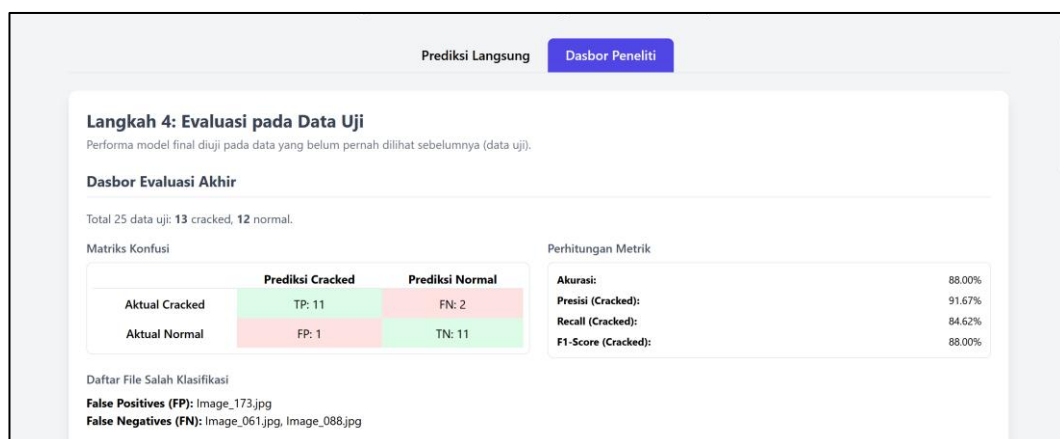
Gambar 3.6 Tabel ekstraksi fitur



Gambar 3.7 Kurva distribusi

3.2.4 Evaluasi Model pada Data Uji

Tahap akhir dari pelatihan adalah pengujian model final menggunakan data uji. Hasil evaluasi ditampilkan pada Gambar 3.8 yang mencakup *confusion matrix* untuk memetakan prediksi benar (*true positive/true negative*) dan kesalahan prediksi (*false positive/false negative*). Sistem juga secara transparan menampilkan daftar nama file yang mengalami kesalahan klasifikasi untuk keperluan analisis kesalahan lebih lanjut. Selain itu, bagian ini juga menampilkan visual *preprocessing*, tabel fitur, tabel rata-rata fitur, dan kurva distribusi.

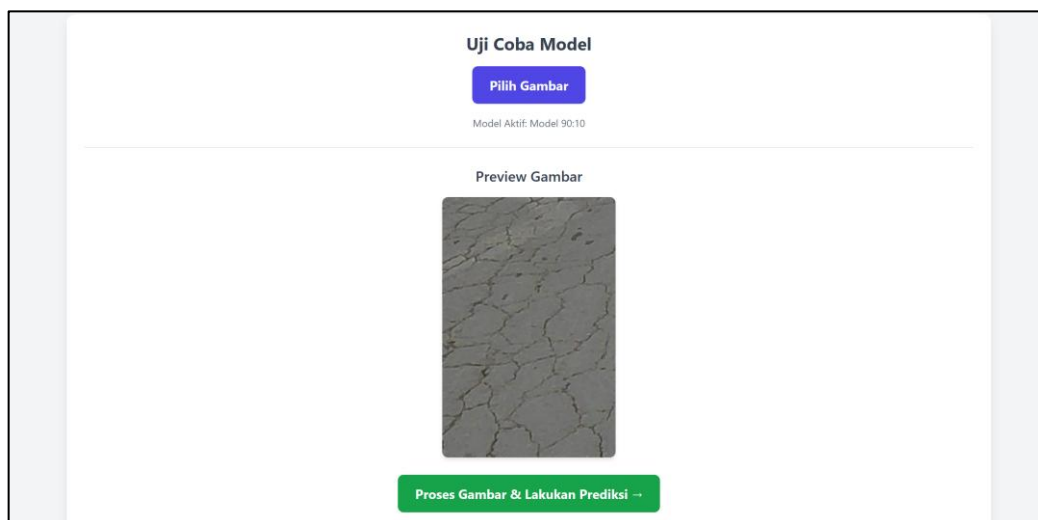


Gambar 3.8 Evaluasi model

3.2.5 Prediksi Langsung

Setelah eksperimen modul dasbor peneliti, sistem juga mengimplementasikan fitur prediksi langsung untuk simulasi penggunaan nyata. Modul ini terintegrasi langsung dengan dasbor peneliti dengan model yang sedang aktif (telah dilatih pada dasbor peneliti) akan digunakan sebagai acuan klasifikasi.

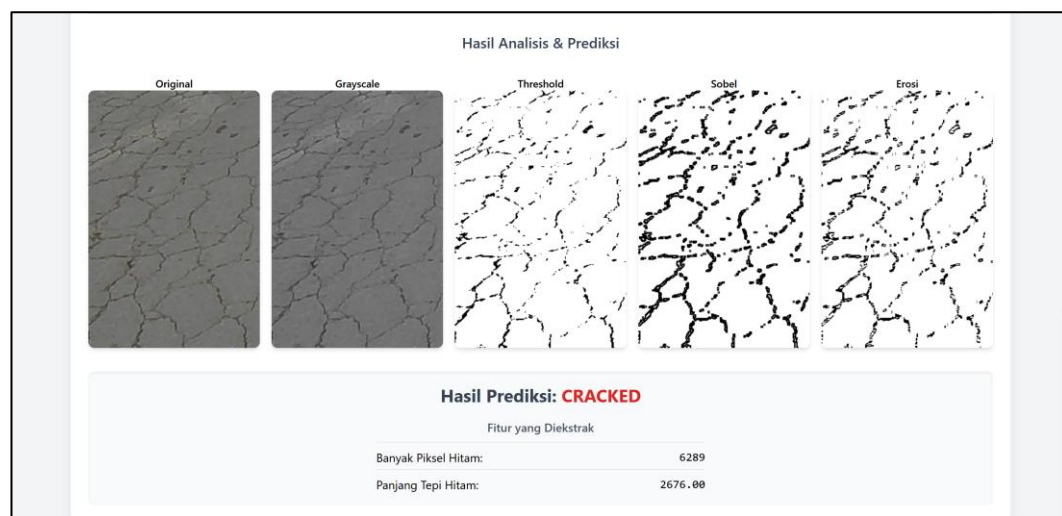
Pada tahap awal pengguna disajikan antarmuka unggahan citra sederhana. Seperti terlihat pada Gambar 3.9, sistem menampilkan status "Model Aktif" untuk memberikan transparansi mengenai parameter model yang digunakan. Setelah pengguna memilih citra dari penyimpanan lokal, sistem menampilkan pratinjau (*preview*) citra tersebut untuk memastikan data yang akan diproses sudah sesuai sebelum tombol eksekusi ditekan.



Gambar 3.9 Tampilan antarmuka pratinjau gambar

Pada tombol proses gambar dan lakukan prediksi, sistem akan melakukan komputasi secara *real-time* melewati tahapan *preprocessing*. Hasil *output* sistem ditunjukkan pada Gambar 3.10. Hasil dari proses tersebut menyajikan tiga informasi penting, yaitu: menampilkan transformasi citra *preprocessing* dari

gambar asli hingga erosi untuk memverifikasi bahwa sistem berhasil mengisolasi pola retakan dari latar belakang jalan, menampilkan label prediksi akhir (*cracked* atau *normal*), menampilkan nilai kuantitatif banyak piksel hitam dan panjang tepi hitam yang dihitung dari citra tersebut. Nilai tersebut yang akan dihitung probabilitasnya oleh algoritma *Gaussian Naïve Bayes* untuk menghasilkan keputusan akhir.



Gambar 3.10 Output dari prediksi langsung

BAB IV

UJI COBA DAN PEMBAHASAN

4.1 Skenario Pengujian

4.1.1. *Split data*

Penelitian ini menggunakan tiga model *split data* yang berbeda. Saat melakukan *split data*, data dipisahkan menjadi dua kategori, yaitu data *training* dan data *testing*. Model dibangun dengan menggunakan data *training* dan diuji menggunakan data *testing* untuk menghitung performanya. Pada penelitian ini akan menggunakan *split data* dengan rasio 70%:30%, 80%:20%, dan 90%:10%. Tiga skenario tersebut dilakukan untuk mencari performa terbaik dari model (Mustolih et al., 2023). Pembagian ini memastikan bahwa proporsi kedua kelas dalam setiap set data tetap seimbang karena ketidakseimbangan data dapat menyebabkan penurunan kinerja pada kelas minoritas (Fitriani et al., 2021). Dengan adanya perbedaan skenario ini akan ditemukan perbedaan hasil prediksi, sehingga dapat dipilih model dengan hasil terbaik.

Tabel 4.1 Pembagian dataset

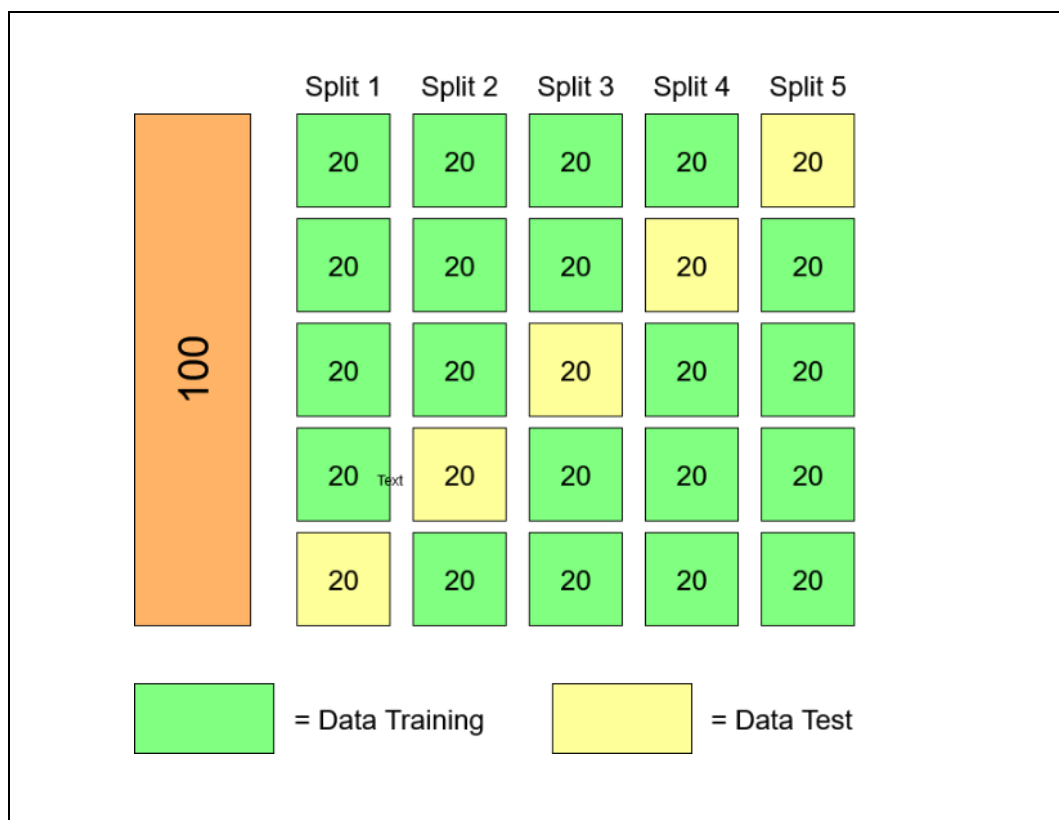
No	Skenario	Komposisi <i>Split data</i>	Jumlah Data <i>Training</i>	Jumlah Data <i>Testing</i>	Jumlah Data Total
1	Skenario 1	70:30	175	75	250
2	Skenario 2	80:20	200	50	250
3	Skenario 3	90:10	225	25	250

4.1.2. *K-Fold Cross Validation*

Dalam strategi validasi silang k-fold, dataset dibagi secara sistematis sesuai dengan nilai k yang ditentukan (Pratama et al., 2025). Nilai umum yang sering digunakan dalam k adalah 5 dan 10 (Anjas and Idham., 2024). Pada penelitian ini

akan dilakukan parameter tuning dengan membandingkan stabilitas antara $k=5$ dan $k=10$. Hasil komparasi menunjukkan bahwa penggunaan *5-fold* menghasilkan standar deviasi performa yang lebih rendah dibandingkan *10-fold*, sehingga memberikan evaluasi yang lebih konsisten (*robust*) terhadap ukuran dataset yang digunakan.

Pada penelitian ini akan mengimplementasikan validasi silang *5-fold*. Prosedur ini melibatkan pembagian himpunan data latih menjadi lima himpunan bagian (*folds*) yang berukuran sama. Pada setiap iterasinya, sebuah model dilatih menggunakan gabungan $4/5$ *folds* sementara satu *fold* dimanfaatkan sebagai himpunan data validasi untuk mengevaluasi performa model yang ditunjukkan pada Gambar 4.1.



Gambar 4.1 Ilustrasi *K-Fold Cross Validation*

4.1.3. Confusion Matrix

Terdapat empat komponen utama dalam *confussion matrix* yang merepresentasikan hasil prediksi. *True Positive* (TP) adalah jumlah prediksi yang benar dan model secara tepat mengklasifikasikan gambar *cracked* sebagai *cracked*. Sebaliknya, *True Negative* (TN) adalah prediksi benar untuk kelas negatif, yaitu model secara akurat mengidentifikasi gambar normal sebagai normal. Selain itu, matriks ini juga mengidentifikasi dua jenis kesalahan, yaitu *False Positive* (FP) dan *False Negative* (FN). *False Positive* (FP) terjadi ketika model secara keliru mengklasifikasikan gambar normal sebagai *cracked* sedangkan *False Negative* (FN) terjadi ketika model gagal mendeteksi gambar *cracked* dan malah mengklasifikasikannya sebagai normal. Struktur *confussion matrix* untuk kelas retak dan normal dapat direpresentasikan sebagai tabel 4.2 berikut.

Tabel 4.2 *Confusion matrix*

Aktual	Prediksi	
Aktual <i>Cracked</i>	True Positif (TP)	False Negative (FN)
Aktual Normal	<i>False Positif</i> (FP)	True Negatif (TN)

Berdasarkan nilai-nilai dari *confusion matrix*, performa model diukur menggunakan empat metrik evaluasi: akurasi, presisi, *recall*, dan *F1-Score* (Susanto & Cahyana, 2025). Rumus perhitungan untuk keempat metrik tersebut adalah sebagai berikut:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4.1)$$

$$Precision = \frac{TP}{TP+FP} \quad (4.2)$$

$$Recall = \frac{TP}{TP+FN} \quad (4.3)$$

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4.4)$$

4.2 Hasil Uji Coba

4.2.1. Hasil Uji Coba Skenario-1

Pada skenario pengujian pertama, pembagian dataset dilakukan dengan rasio 70:30, menghasilkan data latih sebanyak 175 gambar dan data uji sebanyak 75 gambar pada setiap kelas. Data pelatihan memiliki total 175 gambar yang terbagi menjadi 87 data *cracked* dan 88 data normal. Sebelum melatih model final, validasi desain model dilakukan pada data latih menggunakan prosedur *5-Fold Cross Validation*. Hasil dari validasi silang ini disajikan pada Tabel 4.3 yang menunjukkan estimasi performa yang menjanjikan dengan standar deviasi dan rata-rata akurasi mencapai 90.29%, presisi 95.33%, *recall* 83.90% dan *F1-Score* 88.80%.

Tabel 4.3 Hasil *K-Fold Cross Validation* skenario-1

Fold	Akurasi	Presisi	Recall	F1-Score
1	85.71%	93.33%	77.78%	84.85%
2	85.71%	100.00%	70.59%	82.76%
3	91.43%	100.00%	85.00%	91.89%
4	97.14%	100.00%	95.24%	97.56%
5	91.43%	83.33%	90.91%	86.96%
Rata-rata	90.29%	95.33%	83.90%	88.80%
Standar Deviasi	± 4.28%	± 6.53%	± 8.87%	± 5.33%

Evaluasi model final dilakukan pada 75 data uji yang belum pernah dilihat sebelumnya dengan pembagian 38 data *cracked* dan 37 data normal. Hasil prediksi dibandingkan dengan label aktual dan dirangkum dalam *confussion matrix* pada Tabel 4.4 berikut.

Tabel 4.4 *Confussion matrix* skenario-1

Aktual	Prediksi	
Aktual <i>Cracked</i>	TP = 31	FN = 7
Aktual Normal	FP = 2	TN = 35

Berdasarkan *confussion matrix*, diperoleh 31 data yang diklasifikasikan dengan benar sebagai *cracked* (*True Positive* - TP) dan data 35 data yang diklasifikasikan dengan benar sebagai normal (*True Negative* - TN). Model ini juga menghasilkan dua jenis kesalahan. Teridentifikasi 2 data normal yang salah diklasifikasikan sebagai *cracked*, yang merupakan *False Positive* (FP), serta 7 data *cracked* yang gagal dideteksi dan salah diklasifikasikan sebagai normal (*False Negative*). Nilai-nilai ini selanjutnya digunakan untuk menghitung metrik performa kuantitatif yang disajikan pada Tabel 4.5.

Tabel 4.5 Hasil performa skenario-1

Akurasi	Presisi	<i>Recall</i>	F1-Score
88%	93.94%	81.58%	87.32%

Hasil evaluasi menunjukkan akurasi keseluruhan sebesar 88%. Nilai presisi yang tinggi sebanyak 93.93% mengindikasikan bahwa ketika model memprediksi sebuah jalan sebagai *cracked*, prediksi tersebut memiliki probabilitas kebenaran yang tinggi. Sementara itu, nilai *recall* sebesar 81.58% menunjukkan kemampuan model yang baik dalam menemukan sebagian besar retakan yang ada pada data uji,

meskipun masih ada beberapa kasus yang terlewatkan. *F1-Score* sebesar 87.32% menunjukkan bahwa model memiliki keseimbangan yang solid antara presisi dan *recall* sehingga menjadikannya model dengan performa yang baik untuk skenario pertama pengujian ini.

Pada skenario ini, model menghasilkan 2 kasus *False Positive* (FP) dan 7 kasus *False Negative* (FN). Tingginya angka *False Negatif* menunjukkan bahwa model pada skenario ini cenderung terlalu sensitif dalam mengklasifikasikan 7 gambar jalan *cracked* sebagai normal. Sebaliknya, 2 kasus *False Positif* menunjukkan adanya kegagalan model dalam mengenali gambar normal yang ada. Hasil dari klasifikasi pada skenario-1 ini disajikan pada Tabel 7.1 di lampiran.

4.2.2. Hasil Uji Coba Skenario-2

Pada skenario kedua, pengujian diinisiasi dengan pembagian dataset menggunakan rasio 80:20, menghasilkan 200 data latih dan 50 data uji. Data pelatihan memiliki total 200 gambar yang terbagi menjadi 100 data *cracked* dan 100 data normal. Prosedur validasi desain model diterapkan pada data latih menggunakan *5-Fold Cross Validation*. Sebagaimana disajikan pada Tabel 4.6, hasil menunjukkan performa model yang konsisten dengan standar deviasi dan rata-rata akurasi mencapai 91.5%, presisi 96.72%, *recall* 86.32% dan *F1-Score* 91.05%.

Tabel 4.6 Hasil *K-Fold Cross Validation* skenario-2

Fold	Akurasi	Presisi (Cracked)	Recall (Cracked)	F1-Score (Cracked)
1	92.50%	100.00%	84.21%	91.43%
2	82.50%	94.12%	72.73%	82.05%
3	95.00%	100.00%	90.48%	95.00%

4	97.50%	100.00%	94.74%	97.30%
5	90.00%	89.47%	89.47%	89.47%
Rata-rata	91.50%	96.72%	86.32%	91.05%
Standar Deviasi	$\pm 5.15\%$	$\pm 4.28\%$	$\pm 7.58\%$	$\pm 5.26\%$

Selanjutnya, model yang telah dilatih digunakan untuk memprediksi 50 data uji yang terdiri dari 25 data *cracked* dan 25 data normal. Kinerja klasifikasi dievaluasi secara kuantitatif menggunakan *confussion matrix* yang disajikan pada Tabel 4.7.

Tabel 4.7 *Confussion matrix* skenario-2

Aktual	Prediksi	
Aktual <i>Cracked</i>	TP = 19	FN = 6
Aktual Normal	FP = 1	TN = 24

Berdasarkan *confussion matrix*, diperoleh 19 data yang diklasifikasikan dengan benar sebagai *cracked* (*True Positive* - TP) dan 24 data yang diklasifikasikan dengan benar sebagai normal (*True Negative* - TN). Model ini juga menghasilkan dua jenis kesalahan. Teridentifikasi 1 data normal yang salah diklasifikasikan sebagai *cracked* yang merupakan *False Positive* (FP), serta 6 data *cracked* yang gagal dideteksi dan salah diklasifikasikan sebagai normal (*False Negative*). Hasilnya seperti yang tertera pada Tabel 4.8 yang menunjukkan bahwa model mencapai akurasi sebesar 86%, presisi 95%, *recall* 76%, dan *F1-Score* 84.4%

Tabel 4.8 Performa skenario-2

Akurasi	Presisi	<i>Recall</i>	F1-Score
86%	95%	76%	84.44%

Pada skenario ini, model menghasilkan 1 kasus False Positive (FP) dan 6 kasus False Negative (FN). False Positive menunjukkan bahwa model pada skenario ini cenderung terlalu sensitif yang keliru mengklasifikasikan gambar jalan normal sebagai *cracked*. Sebaliknya, 6 kasus False Negative menunjukkan adanya kegagalan model dalam mengenali retakan yang ada. Hasil dari klasifikasi oleh pada skenario-2 ini disajikan pada Tabel 7.2.

4.2.3. Hasil Uji Coba Skenario-3

Skenario pengujian ketiga menggunakan rasio pembagian dataset 90:10, yang terdiri dari 225 data latih dan 25 data uji. Data pelatihan memiliki total 225 gambar yang terbagi menjadi 112 data *cracked* dan 113 data normal. Validasi silang 5-Fold pada data latih yang lebih besar ini menghasilkan standar deviasi dan rata-rata akurasi 89.78%, presisi 96.05%, *recall* 83.36% dan *F1-Score* 89.23% seperti pada tabel 4.9.

Tabel 4.9 Hasil *K-Fold Cross Validation* skenario-3

Fold	Akurasi	Presisi (<i>Cracked</i>)	<i>Recall</i> (<i>Cracked</i>)	<i>F1-Score</i> (<i>Cracked</i>)
1	84.44%	90.91%	80.00%	85.11%
2	91.11%	100.00%	82.61%	90.48%
3	86.67%	95.24%	80.00%	86.96%
4	95.56%	100.00%	90.00%	94.74%
5	91.11%	94.12%	84.21%	88.89%
Rata-rata	89.78%	96.05%	83.36%	89.23%

Standar Deviasi	$\pm 3.87\%$	$\pm 3.87\%$	$\pm 3.87\%$	$\pm 3.87\%$
--------------------	--------------	--------------	--------------	--------------

Evaluasi akhir dilakukan pada 25 data uji yang dibagi menjadi 13 data *cracked* dan 12 data Normal. *Confussion matrix* ditunjukkan pada Tabel 4.10 yang dihasilkan dari proses prediksi menunjukkan peningkatan performa yang signifikan. Berdasarkan *confussion matrix*, sistem memprediksi retak dan data aktual menyatakan retak sebesar 11 gambar. Selain itu terdapat 24 data yang diklasifikasikan dengan benar sebagai normal (*True Negative – TN*). Namun, model ini gagal mendeteksi 2 gambar *cracked* (*False Negatif*) dan 1 gambar normal (*False Positif*). Nilai-nilai ini kemudian digunakan untuk menghitung metrik performa kuantitatif yang disajikan pada Tabel 4.11.

Tabel 4.10 *Confussion matrix* Skenario-3

Aktual	Prediksi	
Aktual <i>Cracked</i>	TP = 11	FN = 2
Aktual Normal	FP = 1	TN = 11

Tabel 4.11 Hasil performa skenario-3

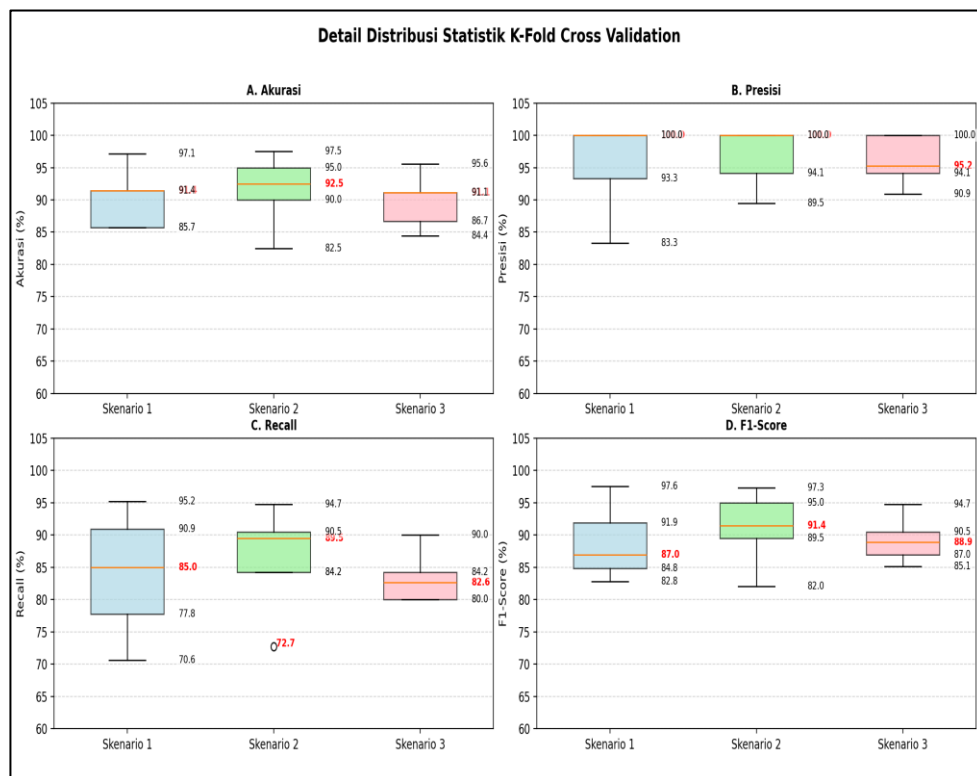
Akurasi	Presisi	<i>Recall</i>	F1-Score
88%	91.67%	84.62%	88%

Berdasarkan perhitungan metrik pada Tabel 4.11, model pada skenario ini berhasil mencapai akurasi 88%. Performa ini didukung oleh presisi yang cukup tinggi sebesar 91.67%. Nilai *recall* tercatat sebesar 84.62% dengan *F1-Score* mencapai 88% yang merupakan skor tertinggi dari ketiga skenario pengujian. Hasil ini menunjukkan bahwa penambahan jumlah data latih secara efektif meningkatkan kemampuan generalisasi dan keandalan model.

Pada skenario ini, model menghasilkan 1 kasus False Positive (FP) dan 2 kasus False Negative (FN). Adanya *False Negatif* menunjukkan bahwa model pada skenario ini cenderung terlalu sensitif dalam mengklasifikasikan gambar jalan *cracked* sebagai normal. Sebaliknya, 1 kasus *False Positif* menunjukkan adanya kegagalan model dalam mengenali gambar normal yang ada. Hasil dari klasifikasi pada skenario-3 ini disajikan pada Tabel 7.3.

4.3 Pembahasan

Evaluasi performa model dilakukan melalui dua tahapan analisis yang saling melengkapi. Tahap pertama adalah tinjauan stabilitas menggunakan *K-Fold Cross Validation* untuk melihat konsistensi model selama proses pelatihan. Tahap kedua adalah pengujian akhir pada data uji untuk mengukur kemampuan generalisasi model dalam mendeteksi retakan pada citra baru. Pendekatan dua tahap ini bertujuan untuk menghindari bias kesimpulan yang hanya didasarkan pada satu kali pengujian. Analisis distribusi performa antar skenario divisualisasikan menggunakan diagram *boxplot* pada Gambar 4.2. Diagram ini memberikan wawasan mendalam mengenai variabilitas dan risiko model yang tidak terlihat jika hanya meninjau dari nilai rata-rata.



Gambar 4.2 Boxplot K-Fold Cross Validation

Pada metrik akurasi (Grafik A), skenario 2 (80:20) menunjukkan nilai *median* tertinggi di angka 92.5%, namun diiringi dengan rentang sebaran yang lebar hingga batas bawah 82.5%. Hal ini berbeda dengan Skenario 3 (90:10) yang meskipun memiliki median di angka 91.1%, namun menampilkan *Interquartile Range* (IQR) yang lebih ringkas, menandakan konsistensi akurasi yang lebih terjaga di berbagai kondisi data.

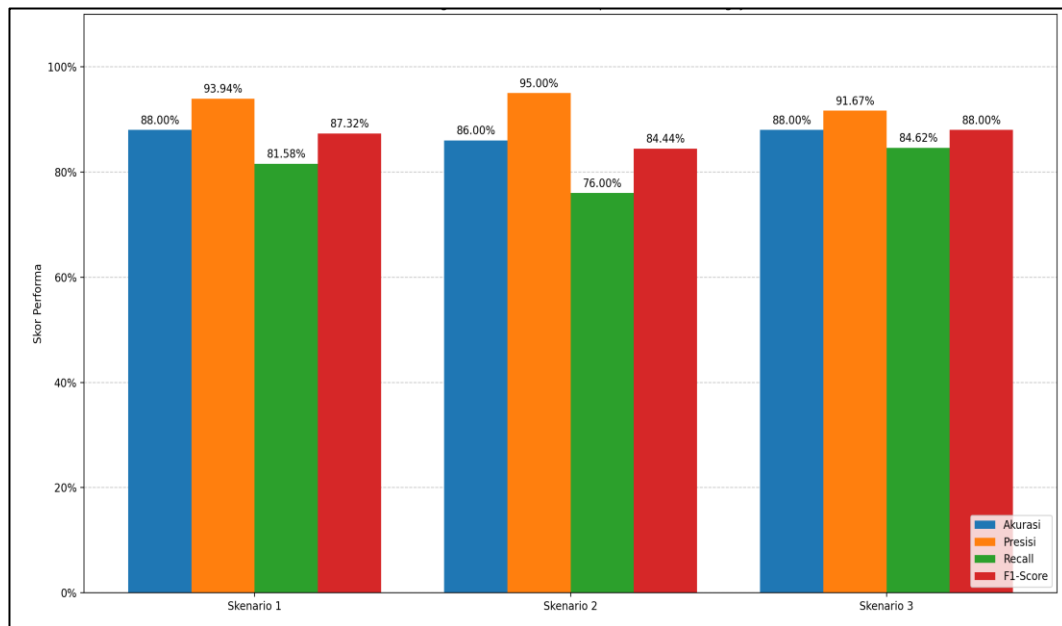
Dominasi performa pada metrik presisi (Grafik B), terlihat pada kedua skenario yang mampu mencapai median sempurna (100%). Hal ini mengindikasikan kemampuan model yang sangat baik dalam meminimalisir *false positive*. Meskipun demikian, skenario 1 (70:30) menunjukkan ketidakstabilan paling tinggi dengan penurunan performa terendah mencapai 83.3%, sedangkan

skenario 3 menunjukkan reliabilitas terbaik dengan kuartil bawah yang tetap bertahan di angka 90.9% dibandingkan dengan skenario 2.

Dinamika paling signifikan terlihat pada metrik *recall* (Grafik C) yang merepresentasikan sensitivitas model. Skenario 2 terindikasi memiliki volatilitas tinggi dengan adanya *outlier* (pencilan) ekstrem pada angka 72.7%. Keberadaan *outlier* ini menjadi sinyal peringatan bahwa skenario 2 memiliki risiko kegagalan deteksi (*false negative*) yang signifikan pada lipatan data tertentu. Namun disisi lain, skenario 3 menunjukkan karakteristik paling *robust* dengan rentang *whisker* yang terjaga di atas 80.0% dibandingkan dengan skenario 1 yang hanya di angka 70.6%. Hal tersebut menunjukkan bahwa skenario 3 memberikan jaminan keamanan performa minimal yang lebih baik.

Terakhir, bagian *F1-Score* (Grafik D) yang merepresentasikan keseimbangan antara presisi dan *recall*, skenario 3 menunjukkan stabilitas dengan bentuk kotak yang paling pendek (varians rendah) dan nilai minimum tertinggi (85.1%). Hal ini mengonfirmasi bahwa skenario 3 memiliki profil risiko paling rendah dibandingkan skenario 1 yang memiliki fluktuasi performa cukup lebar dan skenario 2 yang rentan terhadap anomali data dari nilai *recall*.

Pengujian akhir dilakukan pada himpunan data uji untuk memvalidasi stabilitas performa. Hasil komparasi metrik performa (akurasi, presisi, *recall*, dan *F1-Score*) disajikan pada Gambar 4.3.



Gambar 4.3 Metrik performa semua skenario

Hasil pengujian pada data uji mengonfirmasi indikasi yang ditemukan pada analisis Boxplot sebelumnya. Kekhawatiran mengenai ketidakstabilan skenario 2 terbukti pada tahap ini. Terjadi degradasi performa yang signifikan pada nilai *recall* yang anjlok menjadi 76.00% (turun drastis dari ekspektasi median validasi ~90%). Rendahnya nilai *recall* ini menunjukkan tingginya angka *false negative* yang berarti model gagal mendeteksi retakan jalan yang sebenarnya ada. Hal ini membuktikan bahwa skenario 2 mengalami *overfitting*, yaitu mampu mempelajari pola data latih dengan baik namun gagal melakukan generalisasi pada data baru.

Sementara itu, skenario 1 (70:30) menunjukkan performa yang cukup kompetitif pada data uji. Skenario ini berhasil mencatatkan Akurasi 88.00%, setara dengan Skenario 3. Namun, ketika ditinjau dari aspek sensitivitas deteksi (*Recall*), skenario 1 hanya mencapai angka 81.58% dengan *F1-Score* 87.32%. Meskipun

hasilnya jauh lebih stabil dibandingkan skenario 2, hasil metrik kuncinya masih sedikit inferior dibandingkan skenario 3.

Sebaliknya, skenario 3 (90:10) terbukti mampu mempertahankan performanya secara konsisten sesuai prediksi stabilitas *K-Fold*. Pada data uji, skenario 3 mencatatkan keseimbangan metrik terbaik dengan akurasi 88.00%, presisi 91.67%, dan *recall* 84.62%. Tidak terjadi penurunan performa ekstrem seperti pada skenario 2. Keunggulan margin *recall* sebesar 3% dibandingkan skenario 1 dan 8% dibandingkan skenario 2 menjadi faktor penentu, mengingat pentingnya meminimalisir luputnya deteksi retakan.

Skenario 3 terbukti mampu mempertahankan performanya secara konsisten sesuai prediksi stabilitas *K-Fold*. Pada data uji, Skenario 3 mencatatkan keseimbangan metrik terbaik dengan akurasi 88.00%, presisi 91.67%, *F1-Score* 88.00%, dan *recall* 84.62%. Tidak terjadi penurunan performa ekstrem seperti pada skenario 2 dan sedikit lebih unggul dibandingkan skenario 1.

Dalam perspektif Islam, upaya untuk menjaga dan memelihara fasilitas publik seperti jalan raya sejalan dengan prinsip untuk tidak berbuat kerusakan di muka bumi. Allah SWT berfirman dalam Al-Qur'an Surah Al-A'raf ayat 56:

وَلَا تُفْسِدُوا فِي الْأَرْضِ بَعْدَ إِصْلَاحِهَا وَادْعُوهُ خَوْفًا وَطَمَعًا إِنَّ رَحْمَتَ اللَّهِ قَرِيبٌ مِّنَ الْمُحْسِنِينَ

“Dan janganlah kamu membuat kerusakan di muka bumi, sesudah (Allah) memperbaikinya dan berdoalah kepada-Nya dengan rasa takut (tidak akan diterima) dan harapan (akan dikabulkan). Sesungguhnya rahmat Allah amat dekat kepada orang-orang yang berbuat baik.” (Q.S al-A'raf: 56).

Menurut tafsir Kementerian Agama Republik Indonesia, larangan berbuat kerusakan ini diikuti dengan sebuah analogi yang indah tentang rahmat Allah. Tafsir

tersebut menjelaskan bagaimana Allah SWT meniupkan angin sebagai pembawa kabar gembira yang mendahului hujan. Kemudian hujan tersebut kemudian dihalau ke "daerah yang tandus" atau "tanah yang sudah mati" untuk menjadikannya subur dan menumbuhkan berbagai tanaman. Tafsir ini menggarisbawahi kuasa Allah dalam menghidupkan kembali sesuatu yang tampak telah mati atau rusak.

Dalam konteks penelitian ini, keretakan jalan dapat dianalogikan sebagai tanda awal dari "tanah yang tandus" atau infrastruktur yang mulai "mati". Jika dibiarkan, kerusakan kecil ini akan menyebar dan berpotensi menimbulkan bahaya serta kerugian yang lebih besar. Penelitian ini dapat dipandang sebagai bentuk ikhtiar dalam mengimplementasikan perintah tersebut. Pengembangan teknologi deteksi dini berfungsi layaknya "angin pembawa kabar gembira" yang memberikan peringatan awal sebelum kerusakan menjadi parah. Pengembangan teknologi deteksi dini merupakan wujud dari berbuat kebaikan (*ihsan*), yakni sebuah upaya preventif untuk memelihara kemaslahatan umum, menjaga keselamatan pengguna jalan, dan mengelola amanah infrastruktur dengan cara yang paling efisien dan bertanggung jawab.

Selain itu, penelitian ini secara substansial mengimplementasikan prinsip *Maqashid Syariah*, khususnya pada aspek *Hifdzu An-Nafs* (menjaga jiwa) dan *Hifdzu Al-Maal* (menjaga harta). Pertama, dalam konteks *Hifdzu An-Nafs*, sistem deteksi retakan jalan ini berfungsi sebagai upaya preventif untuk meminimalisir risiko kecelakaan lalu lintas yang dapat mengancam keselamatan jiwa pengguna jalan. Hal ini sejalan dengan perintah Allah tentang segala usaha untuk pemeliharaan jiwa termasuk perbuatan baik (Suhaimi and Muhamad Rezi, 2023).

Kedua, dalam aspek *Hifdzu Al-Maal*, teknologi ini mendukung efisiensi pemeliharaan infrastruktur. Dengan mendeteksi kerusakan sejak dini (tahap retakan), perbaikan dapat dilakukan sebelum kerusakan meluas menjadi lubang besar yang membutuhkan biaya tinggi. Selain itu, kondisi jalan yang baik juga melindungi harta benda (kendaraan) masyarakat dari kerusakan akibat infrastruktur yang buruk. Dengan demikian, kemampuan untuk menjaga dan memelihara harta termasuk kemaslahatan (Irwan, 2021)

BAB V

KESIMPULAN

5.1 Kesimpulan

Terdapat perbedaan karakteristik pada ketiga skenario pembagian data. Meskipun skenario 2 (80:20) memiliki nilai median validasi tertinggi, terbukti mengalami *overfitting* yang ditandai dengan penurunan nilai *recall* hingga 76.00% pada data uji. Sementara itu, Skenario 1 (70:30) mampu menghasilkan akurasi akhir yang kompetitif 88.00% setara dengan skenario 1, namun memiliki nilai penyebaran data yang lebih lebar dengan tingkat *recall* yang lebih rendah sebesar 81.58%. Sebaliknya, skenario 3 (90:10) menunjukkan performa paling stabil dengan distribusi *Interquartile Range* yang ringkas dan tidak ditemukan bias performa selama validasi silang.

Secara keseluruhan, skenario 3 terbukti yang paling optimal dan *robust*. Hal ini ditunjukkan oleh keseimbangan metrik yang superior pada data uji tunggal, dengan *F1-Score* tertinggi sebesar 88.00% dan nilai *recall* 84.62% yang lebih baik dibandingkan skenario 1. Dengan demikian, dapat disimpulkan bahwa kombinasi metode konvolusi dan *Gaussian Naïve Bayes* (GNB) berhasil mengklasifikasikan citra kondisi jalan antara *cracked* dan normal dengan performa yang baik.

5.2 Saran

Meskipun penelitian ini telah berhasil mencapai tujuannya, terdapat beberapa ruang untuk pengembangan dan penelitian di masa mendatang guna meningkatkan performa dan *robust* model. Berikut beberapa saran yang perlu dilakukan untuk penelitian selanjutnya:

1. Eksplorasi Algoritma Klasifikasi Alternatif: Penelitian ini menggunakan *Gaussian Naïve Bayes* yang efisien secara komputasi. Untuk penelitian selanjutnya, disarankan untuk melakukan studi komparatif dengan algoritma machine learning lain yang lebih kompleks, seperti *Support Vector Machine* (SVM) atau *Random Forest*.
2. Pengembangan dan Seleksi Fitur Lanjutan: Dua fitur yang digunakan saat ini terbukti efektif. Namun, penelitian di masa depan dapat mengeksplorasi ekstraksi fitur yang lebih canggih, seperti fitur tekstur menggunakan *Gray-Level Co-occurrence Matrix* (GLCM) atau fitur bentuk lainnya.

DAFTAR PUSTAKA

- Anh, T., & Tam, M. (2022). *Stripping resistance evaluation of bead coating via Hamburg wheel tracking test and image analysis* 함부르크휠 시험과 이미지 분석을 통한 비드코트 탈리 저항성 평가. 24(6), 47–52.
- Anjas, M. (2024). *Optimization of Classification Algorithms Performance with k-Fold Cross Validation*. 2.
- Ansar. (2019). *Ruas Jalan di Bantimurung Retak, Pengendara Mengeluh*. TribunMaros.Com. <https://makassar.tribunnews.com/2019/01/10/ruas-jalan-di-bantimurung-retak-pengendara-mengeluh>
- Bhatt, U., Mani, S., Xi, E., & Kolter, J. Z. (2017). *Intelligent Pothole Detection and Road Condition Assessment*. September. <http://arxiv.org/abs/1710.02595>
- Cha, Y. J., Choi, W., & Büyüköztürk, O. (2017). *Deep learning-Based Crack Damage Detection Using Convolutional Neural Networks*. *Computer-Aided Civil and Infrastructure Engineering*, 32(5), 361–378. <https://doi.org/10.1111/mice.12263>
- Dorafshan, S., Thomas, R. J., & Maguire, M. (2018). Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. *Construction and Building Materials*, 186, 1031–1045. <https://doi.org/10.1016/j.conbuildmat.2018.08.011>
- Fahmi, M., Yudhana, A., & Sunardi, S. (2023). Image Processing Using Morphology on Support Vector Machine Classification Model for Waste Image. *MATRIK : Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, 22(3), 553–566. <https://doi.org/10.30812/matrik.v22i3.2819>
- Fereidoon, H. Z., Nejad, M., & Fahimifar, A. (2016). Image Based Techniques for Crack Detection , Classification and Quantification in Asphalt Pavement : A Review. *Archives of Computational Methods in Engineering*, 424. <https://doi.org/10.1007/s11831-016-9194-z>
- Fitriani, R. D., Yasin, H., & Tarno, T. (2021). Penanganan Klasifikasi Kelas Data Tidak Seimbang Dengan Random Oversampling Pada Naive Bayes (Studi Kasus: Status Peserta KB IUD di Kabupaten Kendal). *Jurnal Gaussian*, 10(1), 11–20. <https://doi.org/10.14710/j.gauss.v10i1.30243>
- G, O. E., C, A. D., & N, O. J. (2025). *Exploring the Effectiveness of Sobel , Canny , and Prewitt Edge Detection Algorithms on Digital Images*.
- Gavilán, M., Balcones, D., Marcos, O., Llorca, D. F., Sotelo, M. A., Parra, I., Ocaña, M., Aliseda, P., Yarza, P., & Amírola, A. (2011). Adaptive road crack detection system by pavement classification. *Sensors*, 11(10), 9628–9657.

<https://doi.org/10.3390/s111009628>

- Ha, M. (2023). *Gaussian Naive Bayes*. Medium. <https://medium.com/@chellehdwjjy/gaussian-naive-Bayes-f05ec0b61d91>
- Hasanah, Q., Oktavianto, H., & Rahayu, Y. D. (2022). Analisis Algoritma Gaussian Naive Bayes Terhadap Klasifikasi Data Pasien Penderita Gagal Jantung Gaussian Naive Bayes Algorithm Analysis Of Data Classification Of Heart Failure Patiens. *Jurnal Smart Teknologi*, 3(4), 2774–1702. <http://jurnal.unmuhjember.ac.id/index.php/JST>
- Hopkins, J. (2019). *How the Sobel Operator Works*. <https://automaticaddison.com/how-the-Sobel-operator-works/>
- Irwan, M., & Kunci, K. (2021). *Kebutuhan dan Pengelolaan Harta Dalam Maqashid Syariah*. 3(2), 160–174.
- Islam, R., Devnath, M. K., Samad, M. D., & Jaffrey Al Kadry, S. M. (2022). GGNB: Graph-based Gaussian naive Bayes intrusion detection system for CAN bus. *Vehicular Communications*, 33(November). <https://doi.org/10.1016/j.vehcom.2021.100442>
- Kashishdafa. (2024). *Gaussian Naive Bayes: Understanding the Basics and Applications*. Medium. <https://medium.com/@kashishdafa0410/gaussian-naive-Bayes-understanding-the-basics-and-applications-52098087b963>
- Khudhair, Zaid N., Ahmed Nidhal, Nidhal K. El Abbadi, Farhan Mohamed, Tanzila Saba, Faten S. Alamri, A. R. (2017). Color to Grayscale Image Conversion Based on Singular Value Decomposition. *IEEE Access*, xx. [https://doi.org/n:DOI 10.1109/ACCESS.2023.3279734](https://doi.org/n:DOI%2010.1109/ACCESS.2023.3279734)
- Kusumaningrum, J., Madenda, S., Karmilasari, & Nahdalina. (2022). Detection and Classification of Road Damage Based on Image Morphology and K-NN Method (K Nearest Neighbour). *International Journal of Engineering and Advanced Technology*, 11(5), 86–90. <https://doi.org/10.35940/ijeat.e3543.0611522>
- Lijie Dong, Kailong Zhang, Mingyue He, S. Z. and C. O. (2025). A Local Thresholding Algorithm for Image Segmentation by Using Gradient Orientation Histogram. *MDPI*, 15, 9808. <https://doi.org/https://doi.org/10.3390/app15179808>
- Ma, S., Khalil, A., Hajjdiab, H., & Eleuch, H. (2020). *applied sciences*. 1–13. <https://doi.org/10.3390/app10114040>
- Mohan, A., & Poobal, S. (2018). Crack detection using image processing: A critical review and analysis. *Alexandria Engineering Journal*, 57(2), 787–798. <https://doi.org/10.1016/j.aej.2017.01.020>
- Mustolih, A., Arsi, P., Subarkah, P., Purwokerto, U. A., Info, A., Mining, D., Classifier, N. B., & Analysis, S. (2023). *Sentiment Analysis Motorku X using*

Applications Naive Bayes Classifier Method. 6(2), 231–242.

- Nnolim, U. A. (2020). Automated crack segmentation via saturation channel thresholding, area classification and fusion of modified level set segmentation with Canny edge detection. *Heliyon*, 6(12), e05748. <https://doi.org/10.1016/j.heliyon.2020.e05748>
- Nugroho, A., & Subanar, S. (2013). Klasifikasi Naïve Bayes untuk Prediksi Kelahiran pada Data Ibu Hamil. *Bimipa*, 23(3), 297–308.
- Nurul A'ayunnisa, Salim, Y., & Azis, H. (2022). Analisis Performa Metode Gaussian Naïve Bayes untuk Klasifikasi Citra Tulisan Tangan Karakter Arab. *Indonesian Journal of Data and Science*, 3(3), 115–121. <https://doi.org/10.56705/ijodas.v3i3.54>
- Oliveira, H., Correia, P. L., & Member, S. (2013). *and Characterization*. 14(1), 155–168.
- Prasetiyo, W. (2024). *Jalan Retak di Jember Bikin Celaka: Truk Ikan Terguling, Sopir Luka*. KumparanNews. <https://kumparan.com/kumparannews/jalan-retak-di-jember-bikin-celaka-truk-ikan-terguling-sopir-luka-22aTCCcTCUW>
- Pratama, N. B., Andrian, R., Agustiansyah, A., & Syarif, A. (2025). Classification of Rice Seedling Structures Using ResNet-50 with Data Augmentation and K-Fold Cross Validation. *Jurnal Pepadun*, 6(1), 10–20. <https://doi.org/10.23960/pepadun.v6i1.258>
- PUPR, K. (2016). *Pedoman Penentuan Indeks Kondisi Perkerasan (IKP)*. SE Menteri PUPR.
- Sitinjak, S. (2020). Pengujian Modifikasi Kernel Konvolusi Untuk Penajaman Dan Penghalusan Citra Berwarna. *Faktor Exacta*, 13(2), 96. <https://doi.org/10.30998/faktorexacta.v13i2.6585>
- Suhaimi, Muhamad Rezi, M. R. H. (2023). *AL-MAQÂSHID AL-SYARÎ'AH; Teori dan Implementasi*. 2(1), 153–170.
- Supiyandi Supiyandi, Trisatin Panggabean, Nuzul Ramadhan, Sri Ratna Dewi, & Salsabila Yusra. (2024). Deteksi Tepi Sederhana Pada Citra Menggunakan Operator Sobel. *Repeater : Publikasi Teknik Informatika Dan Jaringan*, 2(3), 43–56. <https://doi.org/10.62951/repeater.v2i3.90>
- Surya, A. A., & Yamasari, Y. (2024). Penerapan Algoritma Naïve Bayes (NB) untuk Klasifikasi Penyakit Jantung. *Journal of Informatics and Computer Science (JINACS)*, 5(03), 447–455. <https://doi.org/10.26740/jinacs.v5n03.p447-455>
- Talab, A. M. A., Huang, Z., Xi, F., & Haiming, L. (2016). Detection crack in image using Otsu method and multiple filtering in image processing techniques. *Optik*, 127(3), 1030–1033. <https://doi.org/10.1016/j.ijleo.2015.09.147>
- Tong Zhao, Junxiang He, Jingcheng Lv, Delei Min, Y. W. (2023). A

Comprehensive Implementation of Road Surface Classification for Vehicle Driving Assistance: Dataset, Models, and Deployment. *IEEE Transactions on Intelligent Transportation Systems*, 24, 8361–8370. <https://doi.org/10.1109/TITS.2023.3264588>

Trotta, R. (2008). *Bayes in the sky: Bayesian inference and model selection in cosmology*. *Contemporary Physics*, 00(00), 1–41. <https://doi.org/10.1080/0010751YYxxxxxxx>

Wang, R., Liu, Z., Liu, H., Su, B., & Ma, C. (2024). FDSC-YOLOv8: Advancements in Automated Crack Identification for Enhanced Safety in Underground Engineering. *CMES - Computer Modeling in Engineering and Sciences*, 140(3), 3035–3049. <https://doi.org/10.32604/cmes.2024.050806>

Www, W. :, Punjabi, H., Nanwani, R., Vaswani, A., Jotwani, R., Kunte, A., Shahani, T., & College, E. (2008). International Journal of Emerging Technology and Advanced Engineering Intelligent Pothole Detection System. *Certified Journal*, 9001(7), 590. www.ijetae.com

Xu, Y., Xia, Y., Zhao, Q., Yang, K., & Li, Q. (2024). A Road Crack Segmentation Method Based on Transformer and Multi-Scale Feature Fusion. *Electronics (Switzerland)*, 13(12), 1–21. <https://doi.org/10.3390/electronics13122257>

Yadav, D. P., Sharma, B., Chauhan, S., Amin, F., & Abbasi, R. (2024). Enhancing Road Crack Localization for Sustainable Road Safety Using HCTNet. *Sustainability (Switzerland)*, 16(11). <https://doi.org/10.3390/su16114409>

Zou, Q., Zhang, Z., Li, Q., Qi, X., Wang, Q., & Wang, S. (2019). DeepCrack: Learning hierarchical convolutional features for crack detection. *IEEE Transactions on Image Processing*, 28(3), 1498–1512. <https://doi.org/10.1109/TIP.2018.2878966>

LAMPIRAN

Lampiran 1: Hasil Klasifikasi Skenario-1

No	filename	kelas	prediksi	banyak pixel hitam	panjang tepi hitam
1	Image_211.jpg	normal	normal	27	14
2	Image_106.jpg	<i>cracked</i>	<i>cracked</i>	648	336
3	Image_033.jpg	<i>cracked</i>	<i>cracked</i>	1982	970
4	Image_113.jpg	<i>cracked</i>	<i>cracked</i>	108	56
5	Image_049.jpg	<i>cracked</i>	<i>cracked</i>	116	103
6	Image_061.jpg	<i>cracked</i>	normal	0	0
7	Image_152.jpg	normal	normal	6	3
8	Image_052.jpg	<i>cracked</i>	<i>cracked</i>	140	90
9	Image_148.jpg	normal	normal	21	24
10	Image_213.jpg	normal	normal	18	17
11	Image_115.jpg	<i>cracked</i>	<i>cracked</i>	289	167
12	Image_140.jpg	normal	normal	2	4
13	Image_053.jpg	<i>cracked</i>	<i>cracked</i>	475	311
14	Image_060.jpg	<i>cracked</i>	<i>cracked</i>	208	108
15	Image_248.jpg	normal	normal	28	28
16	Image_108.jpg	<i>cracked</i>	<i>cracked</i>	345	217
17	Image_130.jpg	normal	normal	19	15
18	Image_160.jpg	normal	normal	18	10

19	Image_112.jpg	<i>cracked</i>	normal	22	15
20	Image_107.jpg	<i>cracked</i>	<i>cracked</i>	528	320
21	Image_104.jpg	<i>cracked</i>	<i>cracked</i>	135	83
22	Image_076.jpg	<i>cracked</i>	<i>cracked</i>	717	402
23	Image_229.jpg	normal	normal	32	28
24	Image_226.jpg	normal	normal	8	6
25	Image_134.jpg	normal	normal	0	0
26	Image_059.jpg	<i>cracked</i>	<i>cracked</i>	464	268
27	Image_089.jpg	<i>cracked</i>	<i>cracked</i>	2112	1137
28	Image_022.jpg	<i>cracked</i>	<i>cracked</i>	587	348
29	Image_126.jpg	normal	normal	36	42
30	Image_103.jpg	<i>cracked</i>	<i>cracked</i>	284	162
31	Image_177.jpg	normal	normal	32	25
32	Image_129.jpg	normal	normal	12	14
33	Image_188.jpg	normal	<i>cracked</i>	111	53
34	Image_015.jpg	<i>cracked</i>	<i>cracked</i>	502	307
35	Image_173.jpg	normal	<i>cracked</i>	255	113
36	Image_224.jpg	normal	normal	46	35
37	Image_075.jpg	<i>cracked</i>	<i>cracked</i>	294	182
38	Image_064.jpg	<i>cracked</i>	<i>cracked</i>	832	420

39	Image_202.jpg	normal	normal	0	0
40	Image_203.jpg	normal	normal	0	0
41	Image_093.jpg	<i>cracked</i>	<i>cracked</i>	1083	633
42	Image_204.jpg	normal	normal	81	59
43	Image_088.jpg	<i>cracked</i>	normal	16	31
44	Image_216.jpg	normal	normal	39	37
45	Image_042.jpg	<i>cracked</i>	<i>cracked</i>	803	396
46	Image_087.jpg	<i>cracked</i>	<i>cracked</i>	135	108
47	Image_003.jpg	<i>cracked</i>	normal	30	32
48	Image_091.jpg	<i>cracked</i>	normal	35	20
49	Image_168.jpg	normal	normal	51	29
50	Image_038.jpg	<i>cracked</i>	<i>cracked</i>	303	142
51	Image_172.jpg	normal	normal	36	27
52	Image_176.jpg	normal	normal	77	42
53	Image_237.jpg	normal	normal	17	17
54	Image_100.jpg	<i>cracked</i>	<i>cracked</i>	335	269
55	Image_230.jpg	normal	normal	20	12
56	Image_139.jpg	normal	normal	2	4
57	Image_021.jpg	<i>cracked</i>	<i>cracked</i>	467	184
58	Image_072.jpg	<i>cracked</i>	<i>cracked</i>	439	259

59	Image_167.jpg	normal	normal	0	0
60	Image_187.jpg	normal	normal	22	34
61	Image_169.jpg	normal	normal	23	20
62	Image_162.jpg	normal	normal	11	3
63	Image_080.jpg	<i>cracked</i>	<i>cracked</i>	589	332
64	Image_228.jpg	normal	normal	6	8
65	Image_119.jpg	<i>cracked</i>	<i>cracked</i>	131	78
66	Image_058.jpg	<i>cracked</i>	<i>cracked</i>	543	279
67	Image_024.jpg	<i>cracked</i>	normal	31	24
68	Image_249.jpg	normal	normal	4	8
69	Image_215.jpg	normal	normal	2	4
70	Image_128.jpg	normal	normal	0	0
71	Image_244.jpg	normal	normal	12	18
72	Image_190.jpg	normal	normal	2	4
73	Image_030.jpg	<i>cracked</i>	<i>cracked</i>	249	146
74	Image_002.jpg	<i>cracked</i>	normal	2	2
75	Image_083.jpg	<i>cracked</i>	<i>cracked</i>	3261	1693

Lampiran 2: Hasil Klasifikasi Skenario-2

No	filename	kelas	prediksi	banyak pixel hitam	panjang tepi hitam
1	Image_130.jpg	normal	normal	19	15
2	Image_053.jpg	<i>cracked</i>	<i>cracked</i>	475	311
3	Image_125.jpg	<i>cracked</i>	<i>cracked</i>	1642	906
4	Image_100.jpg	<i>cracked</i>	<i>cracked</i>	335	269
5	Image_024.jpg	<i>cracked</i>	normal	31	24
6	Image_187.jpg	normal	normal	22	34
7	Image_172.jpg	normal	normal	36	27
8	Image_061.jpg	<i>cracked</i>	normal	0	0
9	Image_216.jpg	normal	normal	39	37
10	Image_087.jpg	<i>cracked</i>	<i>cracked</i>	135	108
11	Image_021.jpg	<i>cracked</i>	<i>cracked</i>	467	184
12	Image_072.jpg	<i>cracked</i>	<i>cracked</i>	439	259
13	Image_211.jpg	normal	normal	27	14
14	Image_003.jpg	<i>cracked</i>	normal	30	32
15	Image_169.jpg	normal	normal	23	20
16	Image_114.jpg	<i>cracked</i>	<i>cracked</i>	466	229
17	Image_103.jpg	<i>cracked</i>	<i>cracked</i>	284	162
18	Image_152.jpg	normal	normal	6	3

19	Image_052.jpg	<i>cracked</i>	<i>cracked</i>	140	90
20	Image_038.jpg	<i>cracked</i>	<i>cracked</i>	303	142
21	Image_160.jpg	normal	normal	18	10
22	Image_126.jpg	normal	normal	36	42
23	Image_229.jpg	normal	normal	32	28
24	Image_104.jpg	<i>cracked</i>	<i>cracked</i>	135	83
25	Image_093.jpg	<i>cracked</i>	<i>cracked</i>	1083	633
26	Image_226.jpg	normal	normal	8	6
27	Image_105.jpg	<i>cracked</i>	<i>cracked</i>	100	85
28	Image_112.jpg	<i>cracked</i>	normal	22	15
29	Image_022.jpg	<i>cracked</i>	<i>cracked</i>	587	348
30	Image_107.jpg	<i>cracked</i>	<i>cracked</i>	528	320
31	Image_203.jpg	normal	normal	0	0
32	Image_128.jpg	normal	normal	0	0
33	Image_162.jpg	normal	normal	11	3
34	Image_224.jpg	normal	normal	46	35
35	Image_213.jpg	normal	normal	18	17
36	Image_190.jpg	normal	normal	2	4
37	Image_215.jpg	normal	normal	2	4
38	Image_173.jpg	normal	<i>cracked</i>	255	113

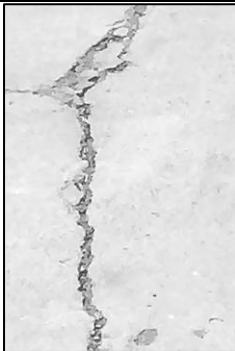
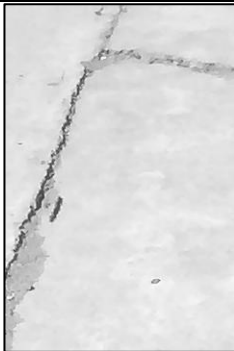
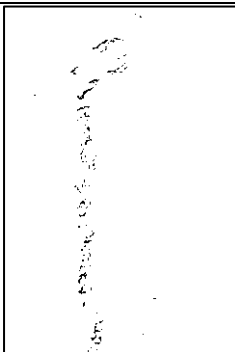
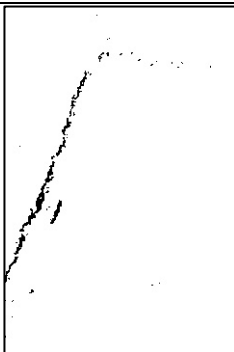
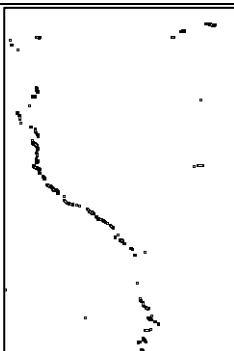
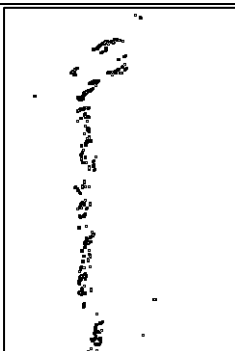
39	Image_228.jpg	normal	normal	6	8
40	Image_075.jpg	<i>cracked</i>	<i>cracked</i>	294	182
41	Image_148.jpg	normal	normal	21	24
42	Image_244.jpg	normal	normal	12	18
43	Image_249.jpg	normal	normal	4	8
44	Image_083.jpg	<i>cracked</i>	<i>cracked</i>	3261	1693
45	Image_134.jpg	normal	normal	0	0
46	Image_088.jpg	<i>cracked</i>	normal	16	31
47	Image_030.jpg	<i>cracked</i>	<i>cracked</i>	249	146
48	Image_139.jpg	normal	normal	2	4
49	Image_015.jpg	<i>cracked</i>	<i>cracked</i>	502	307
50	Image_002.jpg	<i>cracked</i>	normal	2	2

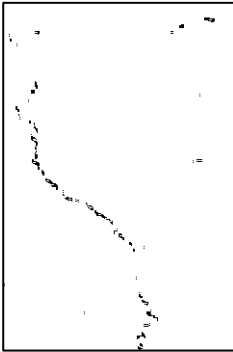
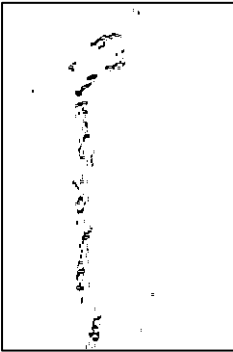
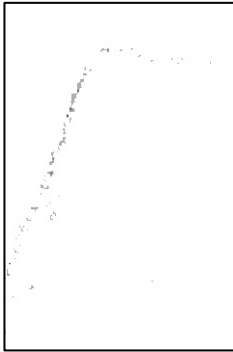
Lampiran 3: Hasil Klasifikasi Skenario-3

No	filename	kelas	prediksi	banyak pixel hitam	panjang tepi hitam
1	Image_015.jpg	<i>cracked</i>	<i>cracked</i>	502	307
2	Image_075.jpg	<i>cracked</i>	<i>cracked</i>	294	182
3	Image_072.jpg	<i>cracked</i>	<i>cracked</i>	439	259
4	Image_169.jpg	normal	normal	23	20
5	Image_148.jpg	normal	normal	21	24
6	Image_103.jpg	<i>cracked</i>	<i>cracked</i>	284	162
7	Image_061.jpg	<i>cracked</i>	normal	0	0
8	Image_115.jpg	<i>cracked</i>	<i>cracked</i>	289	167
9	Image_052.jpg	<i>cracked</i>	<i>cracked</i>	140	90
10	Image_213.jpg	normal	normal	18	17
11	Image_088.jpg	<i>cracked</i>	normal	16	31
12	Image_087.jpg	<i>cracked</i>	<i>cracked</i>	135	108
13	Image_021.jpg	<i>cracked</i>	<i>cracked</i>	467	184
14	Image_224.jpg	normal	normal	46	35
15	Image_187.jpg	normal	normal	22	34
16	Image_107.jpg	<i>cracked</i>	<i>cracked</i>	528	320
17	Image_211.jpg	normal	normal	27	14
18	Image_093.jpg	<i>cracked</i>	<i>cracked</i>	1083	633

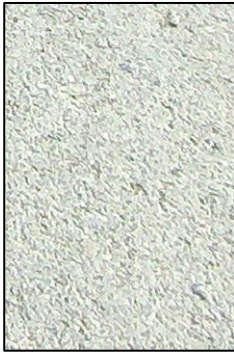
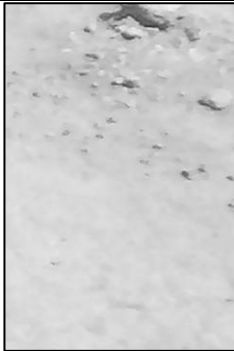
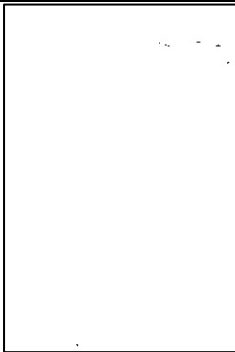
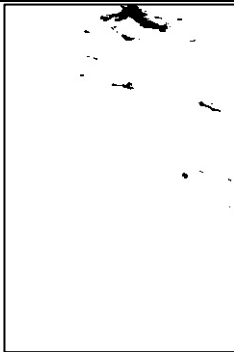
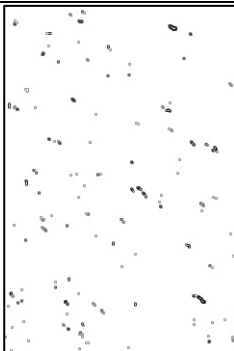
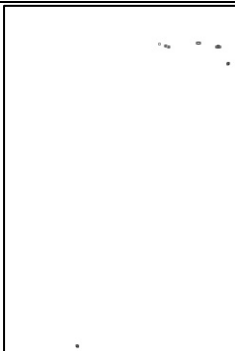
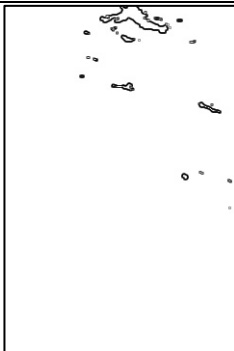
19	Image_244.jpg	normal	normal	12	18
20	Image_162.jpg	normal	normal	11	3
21	Image_173.jpg	normal	<i>cracked</i>	255	113
22	Image_229.jpg	normal	normal	32	28
23	Image_216.jpg	normal	normal	39	37
24	Image_083.jpg	<i>cracked</i>	<i>cracked</i>	3261	1693
25	Image_160.jpg	normal	normal	18	10

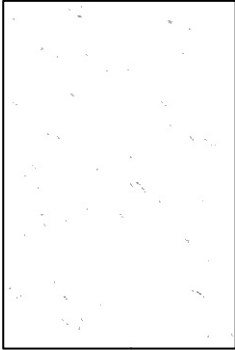
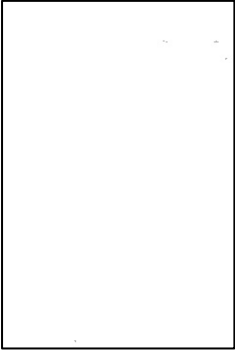
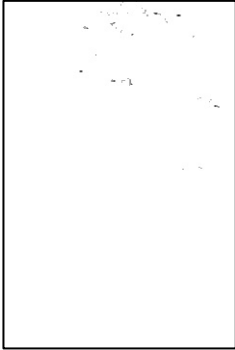
Lampiran 4: Visualisasi *Preprocessing* data cracked

No	<i>Preprocessing</i>	Gambar 1	Gambar 2	Gambar 3
1	Gambar Asli			
2	<i>Grayscale</i>			
3	<i>Thresholding</i>			
4	<i>Sobel</i>			

5	Erosi			
---	-------	---	--	---

Lampiran 5: Visualisasi *Preprocessing* data normal

No	<i>Preprocessing</i>	Gambar 1	Gambar 2	Gambar 3
1	Gambar Asli			
2	<i>Grayscale</i>			
3	<i>Thresholding</i>			
4	<i>Sobel</i>			

5	Erosi			
---	-------	---	--	---