



Analisis Perbandingan Model U-Net dan SegNet dalam Pemrosesan Citra Keretakan Bangunan Berbasis Deep Learning

Tugas Akhir

Oleh:

Muhammad Akmarizan (4212201077)

Laksamana Muda Jaya (4212201099)

**Program Studi Teknik Mekatronika
Jurusan Teknik Elektro
Politeknik Negeri Batam
2025**

Pernyataan Keaslian Tugas Akhir

Saya yang bertandatangan dibawah ini menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya yang berjudul : "Analisis Perbandingan Model U-Net dan SegNet dalam Pemrosesan Citra Keretakan Bangunan Berbasis Deep Learning" adalah **hasil karya sendiri**, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan, dan **bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri**. Semua referensi yang dikutip atau dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan saya ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.



Laksamana Muda Jaya
NIM: 4212201099

Batam, 20 Januari 2026



Muhammad Akmarizan
NIM: 4212201077

Lembar Pengesahan

Tugas Akhir disusun untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Terapan Teknik (S.Tr.T)
di
Politeknik Negeri Batam

Oleh:
Muhammad Akmarizan (4212201077)
Laksamana Muda Jaya (4212201099)

Tanggal Sidang: 20-01-2026

Disetujui oleh :



1. Eko Rudiawan Jamzuri, S.ST., M.S
NIK: 113117



1. Fadli Firdaus, S.Pd., M.Pd
NIK: 122271



2. Dr. Budi Sugandi, S.T., M.Eng
NIK: 100002

[Analisis Perbandingan Model U-Net dan SegNet dalam Pemrosesan Citra Keretakan Bangunan Berbasis Deep Learning]

Abstrak

Kerusakan berupa keretakan pada dinding bangunan merupakan indikator awal penurunan kualitas struktur yang perlu dideteksi secara akurat dan efisien. Proses inspeksi visual manual memiliki keterbatasan dari sisi subjektivitas, waktu, dan konsistensi, sehingga diperlukan pendekatan otomatis berbasis pengolahan citra. Penelitian ini bertujuan untuk mengembangkan dan mengevaluasi sistem deteksi keretakan dinding berbasis deep learning menggunakan arsitektur U-Net dan SegNet, serta menganalisis pengaruh variasi kondisi permukaan dinding dan pencahayaan terhadap kinerja kedua model. Metode yang digunakan meliputi pelatihan model segmentasi citra pada dataset keretakan dinding, evaluasi performa menggunakan metrik IoU, Accuracy, Precision, Recall, dan F1-Score, serta implementasi model ke dalam aplikasi berbasis Graphical User Interface (GUI) untuk pengujian secara interaktif. Hasil penelitian menunjukkan bahwa U-Net memiliki kualitas segmentasi yang lebih presisi dengan nilai IoU dan F1-Score yang lebih tinggi, khususnya pada permukaan dinding bercat dan deteksi keretakan kecil, sedangkan SegNet lebih unggul dari sisi efisiensi komputasi dengan waktu inferensi yang lebih cepat. Variasi permukaan dinding dan pencahayaan terbukti berpengaruh signifikan terhadap performa model, di mana dinding tanpa cat dan pencahayaan redup menyebabkan penurunan kinerja segmentasi. Secara keseluruhan, penelitian ini menegaskan adanya trade-off antara akurasi dan kecepatan, serta menunjukkan bahwa pemilihan model perlu disesuaikan dengan kebutuhan aplikasi deteksi kerusakan bangunan di lapangan.

Kata kunci: deteksi keretakan, segmentasi citra, U-Net, SegNet, deep learning

[Comparative Analysis of U-Net and SegNet Models in Deep Learning-Based Building Crack Image Processing]

Abstract

Cracks in building walls are early indicators of structural quality degradation that need to be detected accurately and efficiently. Manual visual inspection processes have limitations in terms of subjectivity, time consumption, and consistency; therefore, an automated image-processing-based approach is required. This study aims to develop and evaluate a deep learning-based wall crack detection system using U-Net and SegNet architectures, as well as to analyze the effects of variations in wall surface conditions and lighting on the performance of both models. The proposed method involves training image segmentation models on a wall crack dataset, evaluating performance using IoU, Accuracy, Precision, Recall, and F1-Score metrics, and implementing the models into a Graphical User Interface (GUI) application for interactive testing. The results show that U-Net provides more precise segmentation quality with higher IoU and F1-Score values, particularly on painted wall surfaces and for detecting small cracks, while SegNet outperforms in terms of computational efficiency with faster inference times. Variations in wall surface conditions and lighting are proven to have a significant impact on model performance, where unpainted walls and low-light conditions lead to a decline in segmentation accuracy. Overall, this study confirms the existence of a trade-off between accuracy and processing speed and demonstrates that model selection should be adapted to the specific requirements of building damage detection applications in real-world scenarios.

Keywords: crack detection, image segmentation, U-Net, SegNet, deep learning

Kata Pengantar

Puji dan syukur penulis panjatkan ke hadirat Allah SWT atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Laporan Tugas Akhir ini dengan baik. Laporan Tugas Akhir ini disusun sebagai salah satu syarat untuk menyelesaikan pendidikan jenjang Diploma IV pada Program Studi Teknik Mekatronika, Politeknik Negeri Batam.

Tugas Akhir ini membahas analisis keretakan bangunan berbasis *deep learning* dengan mengimplementasikan model U-Net dan SegNet dalam pemrosesan citra digital. Penelitian ini bertujuan untuk mengembangkan sistem segmentasi dan identifikasi keretakan bangunan secara otomatis guna mendukung proses inspeksi kondisi bangunan agar lebih efektif, akurat, dan efisien, serta membantu evaluasi dan pengambilan keputusan terkait pemeliharaan dan keselamatan struktur.

Dalam penyusunan Laporan Tugas Akhir ini, penulis menyadari bahwa keberhasilan penyelesaian laporan ini tidak terlepas dari bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT atas segala nikmat, kesehatan, dan kesempatan yang di berikan.
2. Kedua orang tua dan keluarga, atas doa dan dukungannya.
3. Bapak Diono, S.T., M.Eng, selaku Kepala Program Studi Teknik Mekatronika.
4. Bapak Indra Hardian Mulyadi, S.T., M.Eng, selaku Kepala Jurusan Teknik Elektro.
5. Bapak Fadli Firdaus, M.PD, selaku dosen pembimbing tugas akhir yang telah memberikan arahan dan bimbingan selama proses pengajaran laporan akhir ini.
6. Seluruh dosen Teknik Elektro, atas ilmu dan pengalaman yang telah diberikan selama proses perkuliahan.
7. Teman-teman Teknik Mekatronika, yang telah memberikan dukungan, bantuan, dan kebersamaan selama proses penyusunan buku laporan akhir ini.

Batam, 20 Januari 2026



Laksamana Muda Jaya
NIM: 4212201099



Muhammad Akmarizan
NIM: 4212201077

Daftar Isi

Pernyataan Keaslian Tugas Akhir	Error! Bookmark not defined.
Lembar Pengesahan	ii
Abstrak	iii
<i>Abstract</i>	iv
Kata Pengantar	v
Daftar Isi	v
Daftar Gambar	viii
Daftar Tabel	ix
Bab 1. Pendahuluan	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan	3
1.4. Manfaat	3
1.5. Batasan	3
1.6. <i>Work Breakdown Structure</i>	4
Bab 2. Tinjauan Pustaka	5
2.1. Deteksi Keretakan pada Bangunan	5
2.2. Deteksi Keretakan Berbasis Image Processing	6
2.3. Deep Learning dalam Segmentasi Citra	7
2.3.1. U-Net	8
2.3.2. SegNet	9
2.4. Penelitian Terkait	10
Bab 3. Metodologi Penelitian	13
3.1. Perancangan	13
3.1.1. Perancangan Penelitian	13
3.1.2. Perancangan Sistem	16
3.2. Alat dan Bahan	19
3.3. Pengujian	20

3.3.1. Pengujian Sistem dengan Algoritma U-Net	21
3.3.2. Pengujian Sistem dengan Algoritma SegNet	22
3.3.3. Klasifikasi Tingkat Keretakan	23
3.3.4. Analisis Komparatif Model U-Net dan SegNet.....	24
Bab 4. Hasil dan Pembahasan	30
4.1. Proses Alur Kerja Sistem.....	30
4.2. Hasil Pengujian	33
4.2.1. Hasil Pengujian Performa Utama Model U-Net.....	33
4.2.2. Hasil Pengujian Model U-Net dengan Variasi Permukaan Dinding	39
4.2.3. Hasil Pengujian Model U-Net dengan Variasi Pencahayaan	40
4.2.4. Hasil Pengujian Performa Model Utama SegNet.....	42
4.2.5. Hasil Pengujian Model SegNet dengan Variasi Permukaan Dinding	48
4.2.6. Hasil Pengujian Model SegNet dengan Variasi Pencahayaan	49
4.3. Implementasi GUI Model U-Net dan SegNet.....	50
4.3.1. Hasil Pengujian GUI Model U-Net dan SegNet	53
4.4. Pembahasan.....	56
4.4.1. Analisis Performa Utama Model U-Net dan SegNet	56
4.4.2. Analisis Variasi Permukaan Dinding Model U-Net dan SegNet... 59	
4.4.3. Analisis Variasi Pencahayaan Model U-Net dan SegNet.....	60
4.4.4. Analisis Implementasi GUI Model U-Net dan SegNet.....	61
Bab 5. Kesimpulan dan Saran	63
5.1. Kesimpulan	63
5.2. Saran	64
Daftar Pustaka	66
Biodata	68
Lampiran	69

Daftar Gambar

Gambar 2.1 Arsitektur U-Net [20]	8
Gambar 2.2 Arsitektur SegNet [8]	9
Gambar 3.1 Flowchart Perancangan Penelitian	13
Gambar 3.2 Flowchart Perancangan Sistem	16
Gambar 4.1 Model Summary Arsitektur Model U-Net.....	34
Gambar 4.2 Grafik Pelatihan Arsitektur U-Net.....	35
Gambar 4.3 Hasil Pengujian Arsitektur U-Net Pada Sampel.....	36
Gambar 4.4 Hasil Confusion Matrix U-Net Klasifikasi Tingkat Keretakan.....	37
Gambar 4.5 Model Summary Arsitektur Model SegNet.....	43
Gambar 4.6 Grafik Pelatihan Arsitektur SegNet	44
Gambar 4.7 Hasil Pengujian Arsitektur SegNet Pada Sampel.....	45
Gambar 4.8 Hasil Confusion Matrix SegNet Klasifikasi Tingkat Keretakan	46
Gambar 4.9 Tampilan GUI U-Net dan SegNet	51
Gambar 4.10 Hasil Pengujian Model U-Net dan SegNet pada sampel Data Uji Kelas Retak Ringan	53
Gambar 4.11 Hasil Pengujian Model U-Net dan SegNet pada sampel Data Uji Kelas Retak Sedang.....	54
Gambar 4.12 Hasil Pengujian Model U-Net dan SegNet pada sampel Data Uji Kelas Retak Berat.....	55
Gambar 4.13 Hasil Pengujian Model U-Net dan SegNet pada Citra Non Retak dan Non Relevan	56

Daftar Tabel

Tabel 1.1 <i>Work Breakdown Structure</i>	4
Tabel 2.1 Penelitian Terkait.....	10
Tabel 3.1 Alat dan Bahan.....	19
Tabel 3.2 Perbandingan U-Net dan SegNet (Pengujian Utama)	25
Tabel 3.3 Evaluasi Berdasarkan Variasi Permukaan Dinding	26
Tabel 3.4 Evaluasi Berdasarkan Variasi Pencahayaan	28
Tabel 4.1 Contoh dan Jumlah Citra Dataset	31
Tabel 4.2 Hyperparameter Training	32
Tabel 4.3 Classification Report Arsitektur U-Net pada Data Uji	37
Tabel 4.4 Hasil Metrik Evaluasi Segmentasi Arsitektur U-Net pada Data Uji.....	38
Tabel 4.5 Hasil Distribusi Data Prediksi vs Ground Truth pada Data Uji	39
Tabel 4.6 Evaluasi Berdasarkan Variasi Permukaan Dinding	39
Tabel 4.7 Evaluasi Berdasarkan Variasi Pencahayaan	40
Tabel 4.8 Classification Report Arsitektur SegNet pada Data Uji	46
Tabel 4.9 Hasil Metrik Evaluasi Segmentasi Arsitektur SegNet pada Data Uji.....	47
Tabel 4.10 Hasil Distribusi Data Prediksi vs Ground Truth pada Data Uji.....	48
Tabel 4.11 Evaluasi Berdasarkan Variasi Permukaan Dinding	48
Tabel 4.12 Evaluasi Berdasarkan Variasi Pencahayaan	49

Bab 1. Pendahuluan

1.1. Latar Belakang

Bangunan merupakan infrastruktur vital yang menopang berbagai aktivitas manusia, mulai dari tempat tinggal, perkantoran, hingga fasilitas publik. Kestabilan struktur bangunan adalah faktor mutlak yang menjamin keselamatan penghuninya. Namun, seiring berjalannya waktu, material bangunan mengalami degradasi yang sering kali diawali dengan munculnya keretakan.

Deteksi dini terhadap keretakan bangunan sangat krusial karena keretakan bukan sekadar masalah estetika, melainkan indikator awal dari kegagalan integritas struktural. Secara teknis, retakan yang tidak tertangani menjadi celah masuknya uap air dan oksigen yang memicu korosi pada tulangan beton serta penurunan kekakuan (*stiffness*) bangunan. Jika dibiarkan, kondisi ini dapat berujung pada keruntuhan mendadak yang fatal [1]. Badan Nasional Penanggulangan Bencana (BNPB) kerap menekankan bahwa penyebab utama korban jiwa saat gempa bumi bukanlah guncangan tanah itu sendiri, melainkan reruntuhan bangunan yang gagal mempertahankan integritasnya. Sebagai contoh kasus nyata, gempa Cianjur tahun 2022 mengakibatkan ribuan rumah runtuh akibat kualitas dinding yang buruk dan keretakan lama yang terabaikan [2].

Kondisi ini menjadi urgensi tersendiri di Indonesia. Banyak infrastruktur seperti sekolah, rumah sakit, dan gedung pemerintahan mengalami degradasi struktural karena usia bangunan yang sudah tua, kualitas konstruksi yang tidak seragam, serta minimnya anggaran pemeliharaan. Berdasarkan data Kementerian Pekerjaan Umum dan Perumahan Rakyat (PUPR) dan statistik pendidikan terkini, tercatat sekitar 60% bangunan sekolah dasar di daerah rawan bencana berada dalam kondisi rusak ringan hingga berat [1][3]. Selain itu, posisi Indonesia di cincin api Pasifik (*Ring of Fire*) menjadikan risiko keretakan struktural akibat aktivitas seismik sebagai ancaman permanen yang membutuhkan sistem pemantauan berkala.

Pentingnya pemeliharaan bangunan ini juga telah diatur secara tegas dalam regulasi pemerintah. Undang-Undang Republik Indonesia Nomor 28 Tahun 2002 tentang Bangunan Gedung serta Peraturan Menteri PUPR Nomor 27/PRT/M/2018 mewajibkan pemilik bangunan untuk melakukan pemeliharaan, perawatan, dan pemeriksaan berkala guna menjamin keandalan bangunan (Sertifikat Laik Fungsi) [4]. Dalam konteks regulasi ini, deteksi dini keretakan merupakan bagian fundamental dari standar keselamatan yang wajib dipenuhi agar bangunan tetap laik fungsi dan aman digunakan.

Sayangnya, pada praktiknya proses inspeksi keretakan bangunan umumnya masih dilakukan secara manual oleh tenaga ahli. Metode konvensional ini memiliki kelemahan mendasar: sangat bergantung pada subjektivitas pengamat, memakan waktu lama, serta berpotensi menimbulkan kesalahan (*human error*) akibat

kelelahan atau ketidakkonsistenan dalam penilaian visual [5]. Oleh karena itu, dibutuhkan solusi teknologi yang mampu membantu proses inspeksi secara otomatis, cepat, dan objektif.

Seiring dengan perkembangan teknologi, pendekatan berbasis *Deep Learning* telah menunjukkan potensi besar dan terbukti efektif di berbagai sektor kompleks. Di bidang medis, arsitektur seperti U-Net telah menjadi standar emas untuk segmentasi citra biomedis, misalnya dalam mendeteksi sel kanker atau tumor paru-paru dengan akurasi tinggi yang menyelamatkan nyawa pasien [6]. Sementara itu, di sektor otomotif dan robotika, model SegNet banyak diaplikasikan pada sistem kendaraan otonom (*self-driving cars*) untuk memetakan marka jalan dan rintangan secara *real-time* dengan efisiensi komputasi yang tinggi. Keberhasilan teknologi ini di bidang lain memberikan gambaran peluang besar penerapannya dalam inspeksi kesehatan bangunan [7][8].

Meskipun demikian, sebagian besar penelitian terdahulu mengenai deteksi retak bangunan masih menggunakan dataset luar negeri dengan karakteristik material dan lingkungan yang berbeda dari Indonesia. Belum banyak penelitian yang secara langsung membandingkan performa model U-Net dan SegNet secara *head-to-head* dalam konteks infrastruktur di Indonesia, khususnya yang mencakup klasifikasi tingkat kerusakan spesifik seperti ringan, sedang, dan berat. Inilah yang menjadi celah (*gap*) penting yang ingin diisi oleh penelitian ini [5][7].

Penelitian ini bertujuan untuk mengimplementasikan dan membandingkan performa model U-Net dan SegNet dalam mendeteksi dan menganalisis keretakan bangunan melalui citra digital. Diharapkan hasil dari penelitian ini dapat memberikan kontribusi nyata dalam pengembangan sistem monitoring otomatis guna mendukung deteksi dini dan mitigasi risiko keruntuhan bangunan.

1.2. Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah dalam penelitian ini adalah:

1. Bagaimana implementasi model U-Net dan SegNet dalam mendeteksi keretakan pada bangunan berbasis citra digital?
2. Bagaimana efektivitas kedua model tersebut dalam mendeteksi berbagai tingkat kerusakan (ringan, sedang, berat) pada struktur bangunan?
3. Model mana yang memberikan performa terbaik dalam hal akurasi, presisi, recall, F1-score, dan kemampuan untuk mendeteksi keretakan dalam kondisi permukaan dan pencahayaan yang bervariasi?

1.3. Tujuan

Tujuan dari penelitian ini adalah:

1. Mengimplementasikan model Deep Learning arsitektur U-Net dan SegNet untuk segmentasi citra retakan pada bangunan dengan memperhatikan variasi kondisi permukaan dan pencahayaan.
2. Menganalisis performa kedua model dalam mengklasifikasikan berdasarkan tingkat keretakan (ringan, sedang, berat) menggunakan metrik evaluasi seperti akurasi, presisi, recall, F1-score, Intersection over Union (IoU) dan waktu inferensi.
3. Mengevaluasi dan membandingkan performa kedua model berdasarkan metrik akurasi, presisi, recall, F1-score, IoU dan Waktu Inferensi serta menguji ketahanan model dalam mendeteksi keretakan pada kondisi permukaan dan pencahayaan yang bervariasi untuk menentukan model terbaik.

1.4. Manfaat

Manfaat dari penelitian ini antara lain:

1. Memberikan solusi otomatis dalam mendeteksi keretakan bangunan dengan pendekatan *Deep Learning*.
2. Menjadi acuan dalam pengembangan sistem monitoring bangunan berbasis citra digital.
3. Mendukung keselamatan bangunan melalui deteksi dini terhadap potensi kerusakan struktural.

1.5. Batasan

Penelitian ini memiliki batasan sebagai berikut:

1. Citra yang digunakan merupakan citra digital dari permukaan bangunan dengan jenis keretakan yang dapat diamati secara visual (*surface cracks*). Citra dengan retakan mikroskopis yang memerlukan alat bantu optik khusus atau retakan yang tertutup elemen arsitektural lain (seperti wallpaper atau panel kayu) tidak termasuk dalam kajian.
2. Evaluasi performa model difokuskan pada metrik segmentasi (Akurasi, Presisi, Recall, F1-score, IoU) dan pengukuran waktu inferensi (*inference time*). Penelitian ini tidak mencakup analisis efisiensi konsumsi daya baterai atau optimasi perangkat keras (*hardware acceleration*) secara mendalam.
3. Implementasi dan pengujian sistem dilakukan dalam lingkungan komputasi terkontrol menggunakan *Graphical User Interface* (GUI) berbasis desktop. Penelitian ini belum sampai pada tahap

- pengembangan aplikasi *mobile* (Android/iOS) atau sistem *embedded* yang siap pakai (*deployed*) secara komersial di lapangan.
4. Pengumpulan dataset dokumentasi lapangan terbatas pada area dinding yang dapat dijangkau secara fisik oleh surveyor menggunakan kamera *smartphone* (ketinggian mata manusia/ *eye-level* hingga jangkauan tangan). Area bangunan yang terletak pada ketinggian ekstrem (gedung bertingkat tinggi) atau area yang memerlukan alat bantu akses khusus (seperti *scaffolding* atau *drone*) tidak termasuk dalam lingkup dataset penelitian ini.
 5. Objek deteksi dibatasi pada permukaan dinding bangunan berbahan dasar semen/beton, baik yang memiliki lapisan cat (*painted*) maupun tanpa cat (*unpainted/exposed concrete*). Penelitian ini tidak mencakup deteksi keretakan pada material lain seperti keramik, kayu, kaca, atau logam.

1.6. Work Breakdown Structure

Tabel 1.1 Work Breakdown Structure

No	Nama	Tugas dan Tanggung Jawab dalam Tim
1	Muhammad Akmarizan	Pengujian dan Analisis Model SegNet
2	Laksamana Muda Jaya	Pengujian dan Analisis Model U-Net

Bab 2. Tinjauan Pustaka

2.1. Deteksi Keretakan pada Bangunan

Keretakan pada struktur bangunan merupakan indikator awal dari menurunnya integritas struktural yang tidak boleh diabaikan. Bangunan sebagai elemen vital dalam kehidupan manusia memiliki fungsi sebagai tempat tinggal, fasilitas publik, dan infrastruktur penunjang aktivitas sosial, ekonomi, dan budaya. Seiring berjalannya waktu, struktur bangunan berisiko mengalami degradasi akibat berbagai faktor, seperti beban berlebih (*overload*), pergeseran tanah, fluktuasi suhu, kualitas konstruksi yang kurang baik, serta bencana alam seperti gempa bumi [9].

Apabila tidak terdeteksi sejak dini, keretakan dapat berkembang menjadi kerusakan serius yang membahayakan keselamatan penghuni serta meningkatkan biaya perbaikan. Oleh karena itu, deteksi keretakan menjadi langkah krusial dalam upaya pemeliharaan preventif dan pelestarian bangunan, termasuk bangunan bersejarah seperti dinding batu maupun bata ekspos [10].

Namun, kondisi di lapangan menunjukkan urgensi yang tinggi. Data statistik terbaru menunjukkan bahwa mayoritas kerusakan akibat gempa di Indonesia terjadi pada bangunan *non-engineered* (seperti rumah tinggal dan sekolah) yang dibangun menggunakan dinding bata tanpa perkuatan memadai. Bahkan, pernyataan resmi pemerintah pada tahun 2025 menyoroti bahwa sekitar 80% rumah di Indonesia memiliki risiko tinggi jika terkena gempa besar karena belum memenuhi standar ketahanan struktur [11].

Dampak dari kondisi ini terlihat nyata pada fasilitas publik. Sebagai contoh, pada peristiwa Gempa Lombok 2018, tercatat ribuan gedung sekolah mengalami kerusakan, di mana lebih dari separuhnya adalah bangunan Sekolah Dasar (SD). Hal ini mengindikasikan banyaknya bangunan publik yang beroperasi dengan kerentanan struktural. Studi kasus lain yang relevan adalah Gempa Cianjur (2022), di mana ribuan rumah runtuh bukan semata karena guncangan tanah, melainkan akibat kegagalan dinding bata yang memiliki retakan lama dan kualitas konstruksi yang buruk [2].

Secara teknis, keretakan bangunan dapat diklasifikasikan berdasarkan karakteristik fisiknya untuk menentukan penanganan yang tepat:

- a. Lokasi : retakan dinding, lantai, balok, dan pondasi.
- b. Arah : *longitudinal* (sejajar sumbu struktur), *transversal* (tegak lurus sumbu), dan *diagonal*.
- c. Lebar dan kedalaman : *hairline cracks*, *medium cracks*, dan *wide cracks*
- d. Penyebab : retak susut, retak struktural, retak termal, dan retak akibat adanya Getaran atau beban dinamis.

Metode konvensional yang umum digunakan dalam inspeksi keretakan adalah observasi visual oleh tenaga ahli. Meskipun masih banyak digunakan, metode ini memiliki keterbatasan besar seperti subjektivitas penilaian, waktu pelaksanaan yang lama, dan risiko kesalahan manusia (*human error*), terutama pada bangunan tinggi atau area yang sulit dijangkau. Oleh sebab itu, dibutuhkan pendekatan teknologi yang lebih objektif dan efisien.

Seiring perkembangan teknologi, pendekatan berbasis *Deep Learning* menjadi solusi yang menjanjikan. Dengan memanfaatkan model segmentasi seperti U-Net dan SegNet, sistem dapat dilatih untuk mengenali pola keretakan dari citra digital secara otomatis, membedakannya dari noda atau kotoran dinding, serta mengklasifikasikan tingkat kerusakannya. Hal ini memungkinkan pemantauan kondisi bangunan dilakukan secara cepat, akurat, dan berkelanjutan [12].

2.2. Deteksi Keretakan Berbasis Image Processing

Sebelum berkembangnya metode berbasis *Deep Learning*, pendekatan pengolahan citra klasik seperti deteksi tepi (*edge detection* menggunakan operator Canny atau Sobel), transformasi morfologi, dan teknik *filtering* menjadi pilihan utama [13]. Namun, metode-metode ini memiliki keterbatasan signifikan karena sangat bergantung pada perbedaan intensitas piksel semata. Secara spesifik, metode ini sering kali gagal menangkap keretakan yang sangat halus (*hairline cracks*) karena gradien warnanya terlalu samar dibandingkan dengan tekstur kasar dinding. Selain itu, pendekatan visual murni ini sulit mendeteksi keretakan struktural yang tersamar atau tertutup oleh elemen arsitektural lain, seperti retakan pada kolom utama yang tertutup oleh lapisan plesteran tebal atau cat, sehingga sering kali menghasilkan negatif palsu (*false negatives*).

Pendekatan lain yang sempat digunakan adalah analisis fraktal, yang memanfaatkan parameter dimensi fraktal dan *lacunarity* untuk menangkap pola-pola keretakan yang tidak beraturan [14]. Meskipun efektif untuk tekstur kasar, metode ini memiliki kelemahan utama dalam hal kompleksitas komputasi matematis. Perhitungan dimensi fraktal (misalnya dengan metode *Box-Counting*) mengharuskan sistem melakukan iterasi perhitungan jumlah kotak pada berbagai skala grid secara berulang-ulang untuk mendapatkan nilai dimensi yang akurat. Proses iteratif ini membutuhkan sumber daya komputasi yang besar (*computationally expensive*) dan memakan waktu lama, sehingga kurang efisien jika diterapkan untuk sistem inspeksi *real-time*.

Oleh karena keterbatasan-keterbatasan tersebut—baik dari segi sensitivitas deteksi maupun efisiensi komputasi—pendekatan *Deep Learning* kini menjadi pilihan utama. Teknologi ini menawarkan kemampuan yang lebih adaptif dan akurat dalam mengekstraksi fitur keretakan dari berbagai kondisi permukaan yang kompleks.

2.3. Deep Learning dalam Segmentasi Citra

Convolutional Neural Network (CNN) merupakan arsitektur *Deep Learning* yang menjadi standar emas dalam analisis data visual. Berbeda dengan metode konvensional yang membutuhkan rekayasa fitur manual, CNN bekerja dengan mengekstraksi fitur spasial secara hierarkis dan otomatis. Struktur dasar CNN terdiri dari tiga komponen utama: Lapisan Konvolusi (*Convolutional Layer*) yang menggunakan filter untuk mendeteksi fitur visual seperti garis, tepi, dan tekstur; Lapisan Pooling (*Pooling Layer*) yang berfungsi mereduksi dimensi citra (*downsampling*) guna mengurangi beban komputasi serta menangkap informasi yang paling dominan; dan Lapisan Terhubung Penuh (*Fully Connected Layer*) yang biasanya digunakan untuk klasifikasi akhir [15]. Kemampuan CNN dalam mengekstraksi pola abstrak dari data latih menjadikannya sangat efektif untuk tugas visual yang kompleks [16].

Namun, untuk tugas segmentasi citra yang memerlukan prediksi pada tingkat piksel, arsitektur CNN standar dimodifikasi menjadi *Fully Convolutional Network (FCN)*. Dalam FCN, lapisan *fully connected* dihilangkan dan digantikan oleh mekanisme *Encoder-Decoder*. Bagian Encoder berfungsi mengadaptasi struktur CNN standar untuk mengekstraksi fitur dan menangkap konteks semantik citra (objek apa yang ada), sedangkan bagian Decoder bertugas mengembalikan resolusi spasial citra (*upsampling*) untuk memetakan lokasi objek secara presisi [17]. Segmentasi citra sendiri bertujuan memisahkan citra menjadi wilayah-wilayah homogen berdasarkan kesamaan karakteristik piksel, sehingga area penting seperti retakan dapat diisolasi dari latar belakang [18].

Dua arsitektur FCN yang paling dominan dalam penelitian segmentasi adalah U-Net dan SegNet, yang masing-masing memiliki pendekatan unik dalam mengadaptasi struktur Encoder-Decoder ini:

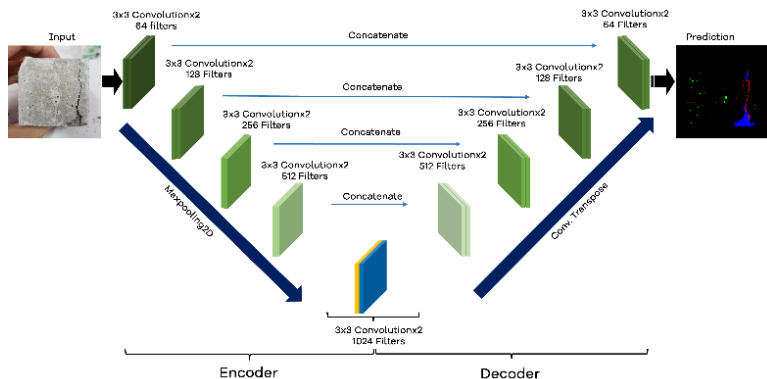
1. U-Net: Arsitektur ini menggunakan mekanisme Skip Connections yang menghubungkan langsung lapisan fitur pada Encoder ke lapisan Decoder yang bersesuaian. Mekanisme ini memungkinkan informasi spasial resolusi tinggi yang mungkin hilang saat proses *pooling* dapat ditransfer kembali ke output. Hal ini menjadikan U-Net lebih unggul dalam hal presisi segmentasi, terutama untuk mendeteksi objek halus seperti keretakan rambut (*hairline cracks*) atau struktur biomedis yang detail.
2. SegNet: Berbeda dengan U-Net, SegNet menggunakan teknik Pooling Indices (indeks max-pooling) untuk merekam lokasi fitur saat proses *downsampling*. Saat proses rekonstruksi (*upsampling*), SegNet hanya memanggil kembali indeks lokasi tersebut tanpa perlu menyimpan seluruh peta fitur (*feature maps*). Pendekatan ini membuat SegNet jauh lebih efisien dalam penggunaan memori dan komputasi, menjadikannya

pilihan ideal untuk aplikasi yang memiliki keterbatasan sumber daya perangkat keras.

Dengan memahami karakteristik kedua model tersebut, penelitian ini akan mengevaluasi efektivitasnya dalam memetakan distribusi retakan pada struktur bangunan, baik dari segi akurasi deteksi maupun efisiensi sistem.

2.3.1. U-Net

U-Net adalah arsitektur CNN berbasis encoder-decoder simetris yang dikembangkan untuk segmentasi citra medis, namun telah diadaptasi secara luas dalam berbagai aplikasi teknik, termasuk deteksi retakan pada permukaan beton [19]. Arsitektur ini memiliki keunggulan berupa *skip connection* yang menjaga informasi spasial resolusi tinggi dari encoder ke decoder. Dalam studi Hadinata et al, *U-Net* digunakan untuk mendeteksi retakan pada permukaan beton yang memiliki void. Model menunjukkan akurasi tinggi hingga 85,92% dan F1-score yang kuat. Selain itu, penelitian oleh Loverdos & Sarhosis menunjukkan bahwa *U-Net* mampu mendeteksi sekaligus mendokumentasikan keretakan pada dinding bata secara akurat [10].



Gambar 2.1 Arsitektur U-Net [20]

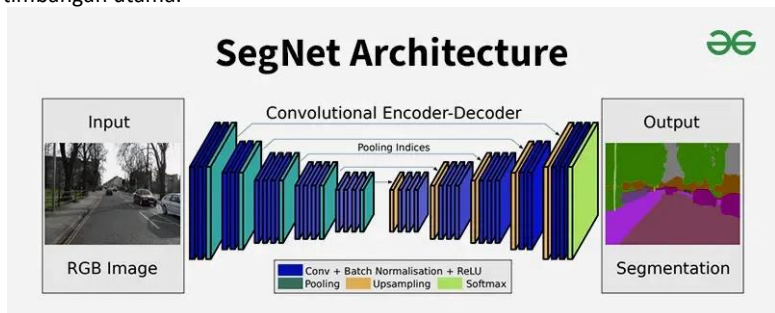
Gambar 2.1 Merupakan contoh bentuk arsitektur dari U-Net yang dirancang khusus untuk segmentasi citra, terdiri dari dua bagian utama yaitu encoder dan decoder yang membentuk struktur simetris menyerupai huruf “U”. Bagian encoder berfungsi mengekstraksi fitur penting dari citra melalui dua lapisan konvolusi 3x3 dan max-pooling, yang secara bertahap mengecilkan ukuran spasial citra namun meningkatkan kedalaman fitur. Setelah mencapai titik terdalam (bottleneck), peta fitur dikembalikan ke resolusi semula oleh decoder melalui proses upsampling menggunakan transposed convolution. Keunggulan utama U-Net terletak pada adanya skip connection, yaitu penggabungan fitur dari encoder

ke decoder pada level yang sama, sehingga informasi spasial detail tetap terjaga selama proses rekonstruksi. Hal ini membuat U-Net sangat efektif dalam mendeteksi pola halus dan bentuk kompleks seperti retakan bangunan. Lapisan akhir dari arsitektur ini menghasilkan peta segmentasi piksel demi piksel yang menandai area yang mengalami kerusakan secara akurat.

2.3.2. SegNet

SegNet merupakan arsitektur segmentasi semantik berbasis CNN yang menggunakan pendekatan encoder-decoder. Perbedaannya dengan U-Net terletak pada cara melakukan proses decoding: melainkan menggunakan *skip connection*, SegNet menyimpan indeks dari hasil max-pooling pada tahap encoder dan menggunakannya kembali saat decoding. Pendekatan ini memungkinkan efisiensi memori yang lebih tinggi serta mempercepat inferensi karena tidak perlu menyimpan seluruh fitur resolusi tinggi dari encoder.

Meskipun dalam studi ini arsitektur SegNet tidak diuji secara langsung, penelitian sebelumnya menunjukkan bahwa SegNet memberikan hasil segmentasi yang cukup halus dan tetap kompetitif dibandingkan dengan model seperti U-Net atau FCN-ResNet. Misalnya, dalam penelitian oleh Aprilyanto dan Yohannes, dilakukan evaluasi model CNN untuk segmentasi keretakan menggunakan VGG-UNet, namun disebutkan pula bahwa SegNet adalah alternatif populer karena efisiensi dan kesederhanaan strukturnya yang cocok untuk sistem dengan keterbatasan komputasi [16]. Dengan demikian, meskipun secara akurasi SegNet sering kali sedikit di bawah U-Net dalam deteksi retakan kompleks, ia tetap menjadi pilihan yang layak ketika sumber daya komputasi terbatas menjadi pertimbangan utama.



Gambar 2.2 Arsitektur SegNet [8]

Gambar 2.2 Merupakan contoh bentuk arsitektur dari SegNet yang dirancang untuk segmentasi citra semantik, dengan struktur encoder-decoder yang terdiri dari lapisan konvolusi, normalisasi, aktivasi ReLU, dan pooling. Pada bagian encoder, citra RGB input diproses melalui beberapa blok konvolusi dan max-

pooling untuk mengekstraksi fitur dan mengurangi resolusi spasial. Yang membedakan SegNet dari U-Net adalah mekanisme decoding-nya: SegNet tidak menggunakan skip connection, melainkan menyimpan indeks hasil max-pooling dari encoder untuk digunakan kembali saat upsampling di decoder. Proses ini menjaga efisiensi memori dan mempertahankan struktur spasial penting saat merekonstruksi citra. Di bagian decoder, setiap tahap upsampling dilakukan menggunakan indeks pooling yang telah disimpan, diikuti oleh konvolusi dan fungsi aktivasi untuk menghasilkan peta segmentasi. Hasil akhir berupa citra segmentasi yang menunjukkan klasifikasi setiap piksel pada area citra input, menjadikan SegNet efektif untuk segmentasi yang konsisten dan efisien dalam penggunaan sumber daya komputasi.

2.4. Penelitian Terkait

Tabel 2.1 Penelitian Terkait

No	Tahun	Penulis	Jurnal	Metode	Kelemahan
1	2020	Velumani	Analysis of Cracks in Structures and Buildings	M2GLD + Otsu Binarization	Kurang akurat untuk berbagai kondisi pencahayaan
2	2020	Pratama	Deteksi Kerusakan Jalan Menggunakan Convolutional Neural Network Berbasis SSD EfficientNet	CNN (SSD EfficientNet)	Kurang optimal untuk segmentasi detail
3	2021	Hadinata	Deep Convolutional Neural Network untuk Mendeteksi Retak pada Permukaan	Deep CNN	Tidak bisa membedakan oid dan cracks tanpa dataset tambahan

			Beton yang Memiliki Void		
4	2024	Muchtar	Penerapan Analisis Berbasis Fraktal dalam Klasifikasi Citra Retakan pada Permukaan Jembatan Beton	Analisis Fraktal + KNN	Memerlukan Pemrosesan matematis kompleks
5	2023	Aprilyanto & Yohannes	Implementasi Arsitektur VGG-UNET dalam melakukan Segmentasi Keretakan Pada Citra Bangunan	VGG-U-Net	Kurang efisien dalam memori

Berdasarkan tabel 2.1 di atas, dapat disimpulkan bahwa sebagian besar studi sebelumnya telah berkontribusi dalam pengembangan metode deteksi keretakan. Namun, masing-masing pendekatan memiliki keterbatasan, seperti ketidakstabilan terhadap pencahayaan pada metode klasik (Velumani, 2020), kurangnya detail segmentasi pada model deteksi objek (Pratama, 2020), serta kompleksitas komputasi yang tinggi pada metode fraktal (Muchtar, 2024). Selain itu, penelitian yang membandingkan performa arsitektur U-Net dan SegNet secara komparatif (*head-to-head*) pada karakteristik dinding bangunan di Indonesia masih terbatas. Kebanyakan studi juga tidak menguji performa model secara spesifik berdasarkan tingkat keparahan retak. Oleh karena itu, penelitian ini bertujuan mengisi kesenjangan tersebut dengan membandingkan akurasi dan efisiensi U-Net dan SegNet pada dataset yang diklasifikasikan menjadi kategori ringan, sedang, dan berat. Hal ini bertujuan untuk membuktikan model mana yang paling andal dalam mendeteksi berbagai variasi lebar retakan untuk mendukung sistem monitoring yang lebih presisi.

1. Komparasi Efisiensi vs Presisi: Belum ada studi yang secara *head-to-head* membandingkan U-Net (yang unggul dalam presisi detail melalui *skip connections*) dengan SegNet (yang unggul dalam efisiensi memori melalui *pooling indices*). Perbandingan ini krusial untuk menemukan keseimbangan terbaik antara akurasi segmentasi dan beban komputasi.

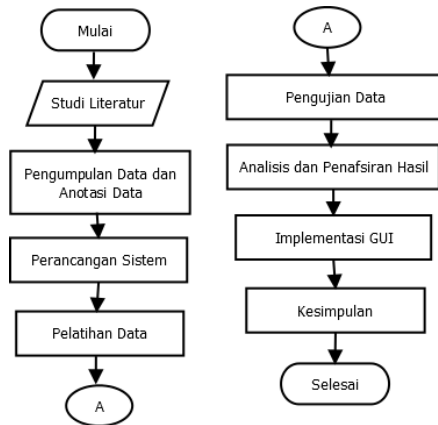
2. Objek Material Dinding (Masonry): Mayoritas penelitian terdahulu (seperti Hadinata, 2021 dan Muchtar, 2024) berfokus pada permukaan beton (*concrete*) jembatan atau jalan raya. Penelitian ini mengisi celah dengan berfokus pada dinding bata berplester, yang merupakan elemen konstruksi dominan pada bangunan gedung dan perumahan di Indonesia, yang memiliki karakteristik tekstur dan *noise* berbeda dari beton.
3. Evaluasi Sensitivitas pada Tingkat Keparahan Berbeda: Mayoritas penelitian terdahulu menyamaratakan semua jenis retakan dalam satu kategori umum (Retak vs Tidak Retak) tanpa membedakan dimensi kerusakannya. Penelitian ini menawarkan kebaruan dengan melakukan evaluasi terpisah pada tiga kategori dataset spesifik: Ringan, Sedang, dan Berat. Meskipun segmentasi dilakukan secara biner (0=Latar Belakang, 1=Retak), pengelompokan ini memungkinkan analisis mendalam untuk mengetahui sensitivitas model terhadap variasi lebar retakan—khususnya untuk menguji kemampuan model dalam menangkap retak rambut (*hairline cracks*) pada kategori ringan dibandingkan retak lebar pada kategori berat.

Oleh karena itu, penelitian ini bertujuan menjembatani kesenjangan tersebut dengan mengimplementasikan dan membandingkan performa U-Net dan SegNet pada citra dinding bangunan, guna menghasilkan sistem deteksi yang tidak hanya akurat secara visual tetapi juga informatif secara klasifikasi tingkat kerusakan.

Bab 3. Metodologi Penelitian

3.1. Perancangan

3.1.1. Perancangan Penelitian



Gambar 3.1 Flowchart Perancangan Penelitian

Gambar 3.1 Menunjukkan tahapan utama dalam pelaksanaan penelitian klasifikasi keretakan bangunan berbasis citra digital. Alur ini dimulai dari studi literatur dan pengumpulan data hingga tahap pengujian dan penarikan kesimpulan. Seluruh proses dirancang secara sistematis untuk memastikan bahwa metode yang digunakan, yaitu U-Net dan SegNet, dapat diuji secara menyeluruh dan dibandingkan performanya dalam mendeteksi serta mengklasifikasikan tingkat kerusakan bangunan.

1. Studi Literatur

Melakukan pencarian dan kajian terhadap referensi ilmiah terkait deteksi keretakan bangunan, segmentasi citra, serta arsitektur Deep Learning seperti U-Net dan SegNet untuk memperkuat dasar teori.

2. Pengumpulan dan Anotasi Data

Dataset yang digunakan dalam penelitian ini terdiri dari total 450 citra digital yang merepresentasikan berbagai kondisi keretakan pada permukaan bangunan. Untuk memastikan keberagaman data dan generalisasi model yang baik, citra diperoleh dari dua sumber utama, yaitu dataset publik dan hasil dokumentasi lapangan (*primary data*).

- a. Sumber Data dan Variasi Lingkungan Sebagian data diambil dari repositori terbuka *Building Crack Computer Vision Project* di platform Roboflow. Dataset ini dipilih karena validitasnya yang telah teruji dalam riset deteksi keretakan infrastruktur sipil, termasuk sebagai referensi *benchmark* CrackSeg9k [21][22]. Sementara itu, citra dokumentasi lapangan diambil secara langsung menggunakan kamera *smartphone* 12 MP di lokasi gedung tua, sekolah, dan area perumahan di Kota Batam. Pengambilan data lapangan ini krusial untuk menguji ketangguhan model pada kondisi dunia nyata (*real-world conditions*). Oleh karena itu, dataset dirancang mencakup variasi lingkungan yang signifikan:
- Variasi Material: Meliputi dinding dengan cat (*painted*) dan dinding semen ekspos/beton kasar (*unpainted*) untuk menguji tekstur yang berbeda.
 - Variasi Pencahayaan: Mencakup kondisi pencahayaan alami intensitas tinggi (*outdoor*) dan kondisi minim cahaya (*low-light/indoor*) guna memastikan model tidak bias terhadap tingkat kecerahan tertentu.
- b. Proses Anotasi Data Seluruh citra mentah kemudian dianotasi menggunakan *platform* Roboflow. Proses pelabelan dilakukan dengan fitur *Smart Select* untuk seleksi cepat dan diperhalus menggunakan *Polygon Tool* agar mask segmentasi mengikuti alur retakan secara presisi dan memisahkannya dari latar belakang (dinding). Dari total 450 citra, anotasi menghasilkan distribusi kelas yang seimbang (*balanced dataset*), yaitu:
- Retak Ringan: 150 citra (33.3%)
 - Retak Sedang: 150 citra (33.3%)
 - Retak Berat: 150 citra (33.3%)
- c. Pembagian Dataset dan Strategi Validasi Untuk menjamin validitas pengujian model, dataset dibagi menggunakan metode *Stratified Random Sampling* agar proporsi tiap kelas kerusakan tetap terjaga di setiap subset. Pembagian dilakukan dengan rasio 70:20:10, yaitu:
- Data Pelatihan (70% - 315 citra): Digunakan untuk melatih bobot model agar mengenali fitur retakan.
 - Data Validasi (20% - 90 citra): Digunakan untuk memantau kinerja model selama proses pelatihan (*training epochs*) dan menyetel *hyperparameter*. Subset ini berfungsi selayaknya mekanisme validasi silang (*cross-validation proxy*) untuk

mendeteksi dini gejala *overfitting* dan memastikan model mampu melakukan generalisasi dengan baik sebelum masuk tahap pengujian akhir.

- Data Pengujian (10% - 45 citra): Merupakan data yang benar-benar baru (*unseen data*) yang digunakan untuk evaluasi final guna mengukur akurasi model saat diterapkan pada kasus nyata. Secara spesifik, 45 citra ini telah dikurasi agar mencakup representasi variasi karakteristik (dinding bercat vs tanpa cat, dan pencahayaan alami vs redup). Sebagian sampel dari subset ini nantinya akan digunakan sebagai dataset pengujian spesifik untuk menganalisis ketangguhan model terhadap variasi tersebut, dengan alokasi masing-masing 3 citra representatif per kategori kerusakan.
- d. Pre-processing dan Augmentasi Seluruh citra dikonversi ke resolusi tetap 256×256 piksel untuk efisiensi komputasi. Selanjutnya, teknik augmentasi data diterapkan pada data pelatihan berupa rotasi ($\pm 15^\circ$), *flipping* horizontal, dan penyesuaian kecerahan (*brightness*) sebesar $\pm 20\%$. Langkah ini bertujuan memperkaya variasi data latih sehingga model menjadi lebih *robust* terhadap perubahan orientasi dan intensitas cahaya di lapangan.
 3. Perancangan Sistem
Merancang sistem deteksi otomatis berdasarkan model yang telah dilatih, termasuk integrasi preprocessing, segmentasi, dan klasifikasi ke dalam satu alur kerja.
 4. Pelatihan Data
Menggunakan dataset yang telah dianotasi untuk melatih model U-Net dan SegNet. Proses ini mencakup *training*, *validation*, dan penyimpanan model terbaik berdasarkan performa.
 5. Pengujian Data
Melakukan pengujian terhadap sistem menggunakan data uji untuk mengukur seberapa baik model mendeteksi dan mengklasifikasikan retakan bangunan.
 6. Analisis dan Penafsiran Hasil
Menganalisis hasil pengujian menggunakan metrik evaluasi seperti akurasi, presisi, recall, dan F1-score. Selain itu, membandingkan performa antara model U-Net dan SegNet.

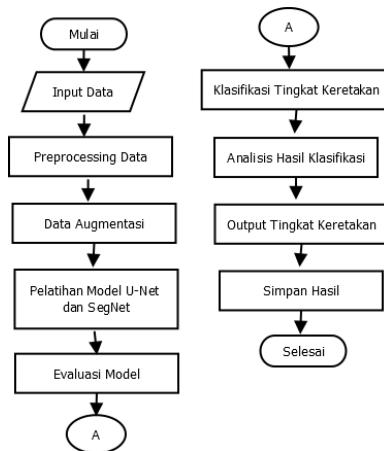
7. Implementasi GUI

Implementasi GUI Tahap terakhir dalam perancangan sistem ini adalah pengembangan antarmuka pengguna grafis atau *Graphical User Interface* (GUI). GUI dirancang sebagai media interaksi antara pengguna dengan sistem *Deep Learning* yang telah dibangun, sehingga proses deteksi dapat dilakukan secara praktis tanpa memerlukan pemahaman mendalam terkait kode pemrograman. Antarmuka ini dibangun berbasis desktop dan menyediakan fitur utama berupa input citra digital, pemilihan model (U-Net atau SegNet) untuk komparasi, serta tombol eksekusi prediksi. Hasil keluaran yang ditampilkan pada GUI mencakup visualisasi citra asli yang disandingkan dengan hasil segmentasi (*masking*) area keretakan, serta informasi tekstual mengenai kategori tingkat kerusakan (ringan, sedang, atau berat) dan parameter performa seperti waktu inferensi. Implementasi ini bertujuan untuk memudahkan pengguna dalam memvisualisasikan hasil analisis model secara langsung dan interaktif.

8. Kesimpulan

Merangkum hasil penelitian, menjawab rumusan masalah, dan mengevaluasi efektivitas pendekatan yang digunakan dalam konteks deteksi keretakan bangunan.

3.1.2. Perancangan Sistem



Gambar 3.2 Flowchart Perancangan Sistem

Gambar 3.2 menunjukkan tahapan penting dalam pengembangan sistem klasifikasi keretakan pada bangunan berbasis citra, yang dalam hal ini dikembangkan dengan pendekatan kombinasi metode deteksi objek dan

klasifikasi berbasis *Deep Learning*. Sistem ini dirancang untuk mendeteksi dan mengklasifikasikan jenis keretakan melalui *image processing* dengan memanfaatkan algoritma U-Net dan SegNet sebagai metode klasifikasi pembandingan.

1. Input Data

Pada tahap ini, sistem menerima data berupa citra bangunan yang diperoleh dari dokumentasi lapangan maupun dataset publik yang tersedia. Citra-citra tersebut merepresentasikan permukaan bangunan yang berpotensi mengalami keretakan dalam berbagai tingkat kerusakan (ringan, sedang, dan berat). Data ini menjadi pondasi awal untuk proses analisis visual menggunakan pendekatan *Deep Learning*. Seluruh proses pengelolaan dan pemrosesan data dilakukan dengan bantuan bahasa pemrograman Python, menggunakan pustaka dan framework seperti TensorFlow dan Keras, yang mendukung implementasi dan pelatihan arsitektur U-Net dan SegNet secara efisien.

2. Preprocessing Citra

Citra yang telah dikumpulkan pertama-tama melalui tahap pembuatan mask untuk menghasilkan label *ground truth*. Berdasarkan data anotasi berformat COCO JSON, sistem mendekode koordinat poligon retakan menjadi representasi biner, di mana area retakan dikonversi menjadi piksel bernilai 255 (putih) dan latar belakang menjadi 0 (hitam). Apabila terdapat beberapa anotasi retakan dalam satu citra, sistem akan menggabungkannya (*merge*) menjadi satu file mask utuh berformat PNG.

Setelah citra asli dan mask terbentuk, keduanya melalui proses *preprocessing* lanjutan agar siap digunakan dalam pelatihan model. Tahap ini mencakup normalisasi ukuran gambar menjadi 256 x 256, penyesuaian skala piksel (*rescaling* nilai piksel menjadi rentang 0 hingga 1), serta penghapusan *noise*.

3. Data Augmentasi

Pada tahap ini, dataset pelatihan diperkaya menggunakan teknik augmentasi data untuk meningkatkan kemampuan generalisasi model dan mencegah terjadinya *overfitting*. Mengingat kondisi pengambilan citra di lapangan sangat bervariasi, augmentasi bertujuan untuk mensimulasikan berbagai gangguan visual yang mungkin muncul di dunia nyata agar model menjadi lebih tangguh (*robust*). Teknik-teknik yang diterapkan meliputi:

- a. Transformasi Geometri (*Rotation & Flipping*): Dilakukan dengan memutar citra (rotasi $\pm 15^\circ$) serta membalik citra secara horizontal dan vertikal. Teknik ini bertujuan untuk

melatih model agar invarian terhadap orientasi objek, mengingat keretakan pada bangunan memiliki arah yang acak (vertikal, horizontal, atau diagonal). Hal ini memastikan model tetap mampu mendeteksi retakan tanpa mepedulikan sudut pengambilan gambar.

- b. Penyesuaian Fotometrik (*Brightness & Contrast Adjustment*): Dilakukan dengan mengubah tingkat kecerahan dan kontras citra sebesar $\pm 20\%$. Teknik ini sangat krusial untuk menyimulasikan kondisi pencahayaan yang tidak ideal, seperti area yang terlalu gelap (*underexposed*) atau terlalu terang. Penyesuaian kontras secara spesifik membantu mempertegas fitur retakan yang sangat tipis (*hairline cracks*) agar lebih mudah dibedakan dari tekstur dinding, sehingga meningkatkan sensitivitas model terhadap keretakan halus.

Dengan penerapan kombinasi teknik ini, variasi data latih menjadi lebih beragam, sehingga model diharapkan mampu memberikan prediksi yang konsisten meskipun dihadapkan pada citra uji dengan kondisi lingkungan yang berbeda dari data pelatihan awal.

4. Pelatihan Model U-Net dan SegNet

Citra yang telah diproses kemudian digunakan untuk melatih dua model segmentasi citra, yaitu U-Net dan SegNet. Kedua model ini bekerja untuk mempelajari pola keretakan pada citra bangunan dengan menghasilkan segmentasi berupa area yang menunjukkan lokasi keretakan. *Label ground truth* berupa mask keretakan digunakan sebagai acuan dalam proses pelatihan ini.

5. Evaluasi Model

Setelah pelatihan selesai, sistem melakukan evaluasi terhadap performa model U-Net dan SegNet. Evaluasi dilakukan menggunakan metrik seperti Intersection over Union (IoU), Dice Coefficient, dan akurasi segmentasi.

Metrik IoU digunakan karena sangat penting dalam konteks segmentasi citra, khususnya untuk mengukur sejauh mana prediksi model tumpang tindih dengan area keretakan sebenarnya. Nilai IoU yang tinggi menunjukkan bahwa model mampu memetakan bentuk dan lokasi keretakan secara presisi terhadap ground truth.

Dice Coefficient juga digunakan sebagai pelengkap, karena metrik ini sensitif terhadap segmentasi area kecil dan memberikan penilaian keseimbangan antara presisi dan recall, yang penting dalam mendeteksi keretakan yang tipis atau halus.

Sementara itu, akurasi segmentasi memberikan gambaran umum tentang proporsi piksel yang diklasifikasikan dengan benar, baik piksel retakan maupun latar belakang.

Dengan kombinasi metrik ini, evaluasi dilakukan secara menyeluruh untuk membandingkan performa kedua model dalam berbagai aspek, termasuk ketepatan segmentasi, konsistensi prediksi, dan kemampuan dalam mendeteksi pola kerusakan yang kompleks.

6. Klasifikasi Tingkat Keretakan

Berdasarkan hasil segmentasi yang dihasilkan oleh model, sistem mengukur luas dan karakteristik retakan. Informasi ini digunakan untuk mengelompokkan tingkat kerusakan bangunan ke dalam beberapa kategori, misalnya: ringan, sedang, dan berat. Klasifikasi ini bergantung pada proporsi area keretakan terhadap keseluruhan citra serta pola distribusinya.

7. Analisis Hasil Klasifikasi

Tahap ini bertujuan untuk menganalisis secara menyeluruh hasil klasifikasi dari setiap citra. Sistem meninjau akurasi dan konsistensi hasil klasifikasi serta mengevaluasi distribusi kelas keretakan pada seluruh dataset. Analisis ini penting untuk menilai stabilitas model dalam pengambilan keputusan.

8. Output Tingkat Keretakan Bangunan

Pada tahap akhir, sistem menyajikan hasil akhir berupa visualisasi area keretakan beserta klasifikasi tingkat keretakannya. Output ini dapat digunakan sebagai dasar pengambilan keputusan teknis dalam perawatan bangunan, dokumentasi kerusakan, atau tindak lanjut profesional di bidang teknik sipil dan konstruksi.

9. Simpan Hasil

Pada tahap ini, hasil segmentasi, klasifikasi tingkat keretakan, serta nilai metrik evaluasi disimpan secara terstruktur. Data yang disimpan mencakup mask hasil prediksi, visualisasi keretakan, dan informasi tingkat keretakan bangunan. Penyimpanan ini bertujuan untuk dokumentasi, analisis lanjutan, dan pelaporan hasil penelitian.

3.2. Alat dan Bahan

Tabel 3.1 Alat dan Bahan

No	Alat/bahan	Spesifikasi/Software	Harga Satuan	Jumlah	Total	Keterangan
----	------------	----------------------	--------------	--------	-------	------------

			n (Rp.)		(Rp.)	
1	Laptop	Lenovo IdeaPad Gaming 3 15ARH7 (Ryzen 5 6600H, RAM 16GB, GPU RTX 3050, SSD 512GB)		1		Milik Pribadi
2	SmartPhone	iPhone 13 Pro Max, Kamera 12 MP		1		Milik Pribadi
3	Software	Python 3.11.9 TensorFlow, Keras, OpenCV, Roboflow, Google Colab				
4	Sistem Operasi	Windows 11 64-bit				
Total						-

Pada Tabel 3.1 Dalam penelitian ini, perangkat keras yang digunakan terdiri dari satu unit laptop Lenovo IdeaPad Gaming 3 15ARH7 yang memiliki spesifikasi cukup untuk melakukan pelatihan model deep learning, dan satu unit iPhone 13 Pro Max yang digunakan untuk dokumentasi lapangan dengan kamera 12 MP.

Adapun perangkat lunak yang digunakan meliputi Python 3.13, dengan pustaka TensorFlow, Keras, OpenCV, Google Colab serta website Roboflow untuk proses anotasi citra. Semua software yang digunakan bersifat open source dan berjalan di atas sistem operasi Windows 11.

3.3. Pengujian

Pengujian dilakukan untuk menilai performa sistem deteksi keretakan pada bangunan berbasis citra digital menggunakan dua model deep learning, yaitu *U-Net* dan *SegNet*. Fokus pengujian adalah menilai kemampuan model dalam melakukan segmentasi area keretakan serta mengklasifikasikan tingkat keparahannya berdasarkan luas area yang terdampak.

Dataset citra bangunan yang mengalami keretakan digunakan sebagai input untuk masing-masing model. Hasil segmentasi dari model kemudian dibandingkan dengan ground truth menggunakan beberapa metrik evaluasi, seperti Akurasi, Presisi, Recall, F1-score, dan Intersection over Union (IoU).

3.3.1. Pengujian Sistem dengan Algoritma U-Net

Model U-Net digunakan untuk segmentasi citra dengan mempertahankan informasi spasial melalui arsitektur *skip connection* antara bagian encoder dan decoder. Proses pengujian terdiri dari tahapan sebagai berikut:

1. Input Data: Dataset citra bangunan yang telah melalui proses normalisasi dan augmentasi.
2. Segmentasi: Model U-Net melakukan prediksi area keretakan dalam bentuk peta segmentasi.
3. Evaluasi: Hasil segmentasi dibandingkan dengan data ground truth.

➤ Rumus Evaluasi Segmentasi

1. Accuracy (Akurasi)
$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Menunjukkan proporsi piksel yang diprediksi dengan benar (baik retak maupun tidak retak setiap kategori).

2. Precision (Presisi)
$$Precision = \frac{TP}{TP + FP}$$

Mengukur seberapa akurat model dalam memprediksi piksel retak (True Positive dibanding semua prediksi positif).

3. Recall (Sensitivity)
$$Recall = \frac{TP}{TP + FN}$$

Mengukur seberapa baik model dalam menemukan semua area retak yang sebenarnya ada.

4. F1-Score
$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Merupakan rata-rata harmonik antara presisi dan recall.

5. Intersection over Union (IoU)
$$IoU = \frac{Area_{Overlap}}{Area_{Union}} = \frac{TP}{TP + FP + FN}$$

Digunakan untuk mengevaluasi tumpang tindih antara prediksi model dan ground truth dalam segmentasi area keretakan.

$$6. \quad \begin{array}{l} \text{Macro} \\ \text{Avera} \\ \text{ge} \end{array} \quad MA = \frac{Metric_{Kelas1} + Metric_{Kelas2} + Metric_{Kelas3}}{n}$$

Macro Average menghitung rata-rata metrik (Precision, Recall, atau F1-Score) dengan memberikan bobot yang sama untuk setiap kelas, tanpa mempertimbangkan berapa banyak jumlah data (*support*) di masing-masing kelas tersebut. Dimana n yaitu jumlah kelas, Ini berguna untuk melihat performa model secara keseluruhan di semua kategori secara adil.

$$7. \quad \begin{array}{l} \text{Weighte} \\ \text{d} \\ \text{Average} \end{array} \quad WA = \frac{\sum (Metric_i \times Support_i)}{Total\ Support}$$

Weighted Average menghitung rata-rata dengan mempertimbangkan jumlah sampel (*support*) di setiap kelas. Kelas dengan jumlah data yang lebih banyak akan lebih memengaruhi hasil akhir.

$$8. \quad \begin{array}{l} \text{Mean} \\ \text{Pixel} \\ \text{Accuracy} \end{array} \quad MPA = \frac{1}{k+1} \sum_{i=0}^k \frac{p_{ii}}{\sum_{j=0}^k p_{ij}}$$

Mean Pixel Accuracy adalah metrik evaluasi pada segmentasi citra yang menghitung rata-rata dari akurasi klasifikasi piksel untuk setiap kelas secara individu. Metrik ini memberikan gambaran seberapa tepat model dalam mengklasifikasikan piksel untuk setiap kategori (misalnya: retak vs. bukan retak) sebelum dirata-ratakan.

$$9. \quad \begin{array}{l} \text{Accuracy} \\ \text{Confusio} \\ \text{n Matrix} \end{array} \quad Akurasi = \frac{\text{jumlah prediksi benar}}{\text{Jumlah seluruh data}}$$

Accuracy Confusion Matrix untuk mengukur evaluasi keseluruhan performa model.

3.3.2. Pengujian Sistem dengan Algoritma SegNet

Model SegNet memiliki arsitektur encoder-decoder yang lebih efisien dalam penggunaan memori, dengan proses *upsampling* berdasarkan indeks dari *max-pooling*. Ini membuat SegNet cocok untuk segmentasi *real-time* dan citra beresolusi tinggi.

Langkah pengujian dilakukan sebagai berikut:

1. Input Data: Dataset citra bangunan yang sama digunakan untuk menjamin konsistensi pengujian.
 2. Segmentasi: Model SegNet memproses citra dan menghasilkan peta segmentasi area keretakan.
 3. Evaluasi: Hasil segmentasi dibandingkan dengan ground truth menggunakan metrik yang sama seperti pada model U-Net.
- Evaluasi dan Klasifikasi
- Semua metrik evaluasi (Akurasi, Presisi, Recall, F1-Score, IoU) dihitung menggunakan rumus yang sama seperti pada model U-Net.
 - Tingkat keretakan juga dikategorikan berdasarkan luas area retakan.
 - Waktu inferensi dan efisiensi komputasi juga dicatat untuk membandingkan performa runtime antara U-Net dan SegNet.

3.3.3. Klasifikasi Tingkat Keretakan

Untuk menentukan kategori keretakan secara objektif, penelitian ini menggunakan metrik **Crack Area Index (CAI)**. CAI menghitung proporsi piksel retak terhadap total piksel citra sesuai dengan metode yang diadaptasi dari König et al. [23]. Rumus CAI yang digunakan adalah sebagai berikut:

$$P = \frac{\Sigma_{\text{PikselPutih}(\text{Mask})}}{\Sigma_{\text{TotalPiksel}(\text{Resolusi})}} \times 100\%$$

Nilai P yang dihasilkan kemudian diklasifikasikan ke dalam tiga tingkat kerusakan (Ringan, Sedang, Berat) sebagai dasar pengambilan keputusan mitigasi struktural.

1. Retak Ringan: Luas < 5% dari total area citra
Citra dikategorikan sebagai retak ringan apabila area retak mencakup kurang dari 5% dari total area citra. Secara visual, retakan tampak sebagai garis tipis, terisolasi, dan belum menyebar secara luas. Kategori ini merepresentasikan kondisi bangunan yang umumnya masih stabil secara struktural, di mana retakan hanya bersifat kosmetik atau permukaan. Dalam banyak standar teknik sipil, seperti ACI (American Concrete Institute), retakan minor seperti ini dianggap tidak berbahaya secara struktural, namun tetap perlu dipantau.
2. Retak Sedang: Luas 5–20%
Retakan dikategorikan sedang apabila area retaknya berkisar antara 5% hingga 20% dari total area citra. Karakteristik visualnya mencakup retak yang mulai menyebar secara horizontal atau vertikal,

dengan pola yang lebih kompleks dibandingkan kategori ringan. Kategori ini merepresentasikan potensi kerusakan fungsional yang mulai muncul, seperti penurunan daya tahan material terhadap beban atau air. Dari sudut pandang teknis, kondisi ini harus masuk dalam prioritas pemeriksaan lanjutan, khususnya jika ditemukan pada elemen struktural utama.

3. Retak Berat: Luas > 20%

Jika area retakan melebihi 20% dari total area citra, maka citra diklasifikasikan sebagai retak berat. Pada kategori ini, retakan tampak menyebar luas, membentuk jaringan retak besar, bercabang, atau saling terhubung. Kondisi ini menunjukkan kemungkinan adanya kerusakan struktural signifikan yang dapat mengancam keselamatan dan integritas bangunan. Dalam konteks sistem monitoring otomatis, kategori ini memiliki prioritas deteksi tertinggi, dan biasanya memerlukan tindakan teknis segera seperti penguatan atau evakuasi area

Pendekatan ini mengacu pada metode pengklasifikasian yang digunakan dalam studi Loverdos & Sarhosis serta diadopsi dalam penelitian König et al. yang menilai tingkat keparahan kerusakan berdasarkan proporsi segmentasi area retak terhadap citra asli [10][23]. Pendekatan berbasis area ini dinilai praktis karena bersifat kuantitatif, objektif, dan mudah diintegrasikan ke dalam sistem klasifikasi berbasis *Deep Learning*.

3.3.4. Analisis Komparatif Model U-Net dan SegNet

Untuk mengevaluasi efektivitas model secara menyeluruh, penelitian ini merancang analisis komparatif yang tidak hanya berbasis pada akurasi segmentasi, tetapi juga pada efisiensi komputasi dan ketangguhan model terhadap variasi kondisi lapangan. Evaluasi dilakukan dengan membandingkan hasil prediksi model terhadap *ground truth* pada data uji menggunakan enam metrik performa utama: *Accuracy*, *Precision*, *Recall*, *F1-Score*, *Intersection over Union* (IoU), dan waktu inferensi (*inference time*). Analisis ini difokuskan pada tiga aspek utama:

1. Pengujian Efisiensi Komputasi (Waktu Inferensi) Salah satu parameter kunci yang ditambahkan dalam penelitian ini adalah pengukuran waktu inferensi (*inference time*) per citra. Pengujian ini bertujuan untuk menilai seberapa efisien kedua model dalam memproses satu gambar input. SegNet, dengan arsitektur *decoder* yang hanya menggunakan indeks *max-pooling*, secara teori memiliki beban komputasi yang lebih ringan dibandingkan U-Net yang memproses *feature maps* lengkap. Data waktu inferensi ini akan menjadi indikator penting untuk menentukan model

mana yang lebih efisien jika nantinya dikembangkan lebih lanjut ke perangkat dengan spesifikasi rendah (*low-resource devices*), meskipun pada penelitian ini implementasi masih dilakukan dalam bentuk antarmuka berbasis desktop (GUI).

- 2. Analisis Karakteristik Keretakan Selain metrik kuantitatif, analisis mendalam dilakukan untuk memahami bagaimana perbedaan arsitektur mempengaruhi kemampuan deteksi pada jenis keretakan yang berbeda:
 - Keretakan Ringan (*Hairline Cracks*): Model U-Net diprediksi lebih unggul dalam mendeteksi retakan tipis karena adanya mekanisme *skip connections* yang mentransfer informasi detail spasial dari *encoder* ke *decoder*, meminimalisir hilangnya fitur halus saat *downsampling*.
 - Keretakan Berat & Latar Belakang Kompleks: Pada retakan lebar atau area dengan latar belakang dinding kasar (*unpainted*), performa SegNet akan dibandingkan untuk melihat apakah efisiensi memorinya tetap mampu menghasilkan batas segmentasi yang cukup akurat tanpa terganggu oleh *noise* tekstur dinding.
- 3. Evaluasi Berbasis Variasi Data Evaluasi akan dilakukan dalam tiga tahap pengujian terpisah untuk memastikan validitas dan ketangguhan model pada berbagai kondisi. Berikut adalah rincian tabel pengujian beserta penjelasannya:
 - a. Pengujian Utama
Pengujian tahap pertama merupakan evaluasi tingkat tinggi untuk melihat performa model secara umum pada data uji acak yang merepresentasikan kondisi standar.

Tabel 3.2 Perbandingan U-Net dan SegNet (Pengujian Utama)

No	Kategori Keretakan	Model	Jumlah data	Accuracy	Precision	Recall	F1-Score	IoU	Waktu Inferensi(s)
1	Ringan	U-Net	15						
		SegNet	15						

2	Sedan g	U-Net	15						
		SegNet	15						
3	Berat	U-Net	15						
		SegNet	15						

Tabel 3.2 menyajikan rekapitulasi performa model U-Net dan SegNet berdasarkan tiga tingkat keparahan retak (Ringan, Sedang, dan Berat) dengan jumlah sampel uji masing-masing sebanyak 15 citra. Data pada tabel ini akan menjawab pertanyaan mendasar mengenai model mana yang memiliki akurasi rata-rata terbaik. Analisis dari tabel ini akan menyoroti *trade-off* antara kualitas segmentasi (diwakili oleh nilai IoU dan F1-Score) dengan kecepatan pemrosesan (Waktu Inferensi). Diharapkan tabel ini dapat membuktikan hipotesis bahwa U-Net lebih unggul dalam metrik presisi, sementara SegNet lebih unggul dalam efisiensi waktu.

- b. Pengujian Berdasarkan Variasi Permukaan Dinding
 Pengujian tahap kedua dilakukan secara spesifik untuk menguji sensitivitas model terhadap tekstur latar belakang (*background noise*).

Tabel 3.3 Evaluasi Berdasarkan Variasi Permukaan Dinding

No	Kategori Keretakan	Kondisi Permukaan	Model	Accuracy	Precision	Recall	IoU (%)	F1-Score (%)	Waktu Inferensi (s)
1	Ringan	Dinding Bercat	U-Net						
			SegNet						
		Tanpa Cat (Semen)	U-Net						
			SegNet						

2	Sedang	Dinding Bercat	U-Net						
			SegNet						
		Tanpa Cat (Semen)	U-Net						
			SegNet						
3	Berat	Dinding Bercat	U-Net						
			SegNet						
		Tanpa Cat (Semen)	U-Net						
			SegNet						

Tabel 3.3 dirancang untuk menganalisis ketangguhan (*robustness*) model terhadap jenis material dinding. Sampel dipisahkan menjadi dinding bercat (permukaan halus/homogen) dan dinding tanpa cat atau semen ekspos (permukaan kasar/*noisy*). Permukaan dinding semen yang kasar seringkali memiliki tekstur garis tak beraturan yang menyerupai pola retakan, sehingga sangat rentan memicu kesalahan deteksi (*False Positive*). Oleh karena itu, metrik Precision menjadi indikator kunci dalam tabel ini. Nilai Precision akan menunjukkan kemampuan model dalam membedakan secara spesifik antara retakan struktural yang asli dengan sekadar *noise* atau tekstur permukaan dinding, memastikan model tidak "terlalu sensitif" pada dinding kasar.

c. Pengujian Berdasarkan Variasi Pencahayaan

Pengujian tahap ketiga bertujuan menguji kemampuan adaptasi model terhadap intensitas cahaya yang berbeda, menyimulasikan kondisi inspeksi di lapangan.

Tabel 3.4 Evaluasi Berdasarkan Variasi Pencahayaan

N o	Kategori Keretakan	Kondisi Permukaan	Model	Accuracy	Precision	Recall	IoU (%)	F1-Score (%)	Waktu Inferensi (s)
1	Ringan	Alami (Terang)	U-Net						
			SegNet						
		Redup (Indoor)	U-Net						
			SegNet						
2	Sedang	Alami (Terang)	U-Net						
			SegNet						
		Redup (Indoor)	U-Net						
			SegNet						
3	Berat	Alami (Terang)	U-Net						
			SegNet						
		Redup (Indoor)	U-Net						
			SegNet						

Tabel 3.4 berfokus pada evaluasi performa model di bawah kondisi pencahayaan alami (terang/*outdoor*) dibandingkan dengan pencahayaan

redup/buatan (*indoor/low-light*). Kondisi minim cahaya seringkali menurunkan kontras antara retakan dan dinding, membuat retakan tipis sulit terdeteksi. Oleh karena itu, metrik Recall menjadi indikator utama dalam tabel ini. Nilai Recall akan menunjukkan sensitivitas model dalam menemukan seluruh retakan asli tanpa ada yang terlewat (*False Negative*), meskipun dalam kondisi gelap. Hasil dari tabel ini akan memvalidasi efektivitas teknik augmentasi data (penyesuaian *brightness/contrast*) yang telah dilakukan.

Bab 4. Hasil dan Pembahasan

4.1. Proses Alur Kerja Sistem

Tahap pertama dalam pengembangan sistem ini adalah pengumpulan dataset yang dilakukan melalui dua sumber utama, yaitu pengambilan citra dokumentasi lapangan menggunakan kamera *smartphone* pada gedung tua, sekolah, dan area perumahan di Kota Batam, serta pengunduhan dataset publik dari repositori Roboflow. Citra yang terkumpul mencakup berbagai kondisi keretakan dinding dengan variasi pencahayaan dan tekstur permukaan guna memastikan model memiliki cakupan data yang luas. Dataset yang berjumlah 450 citra ini kemudian dibagi menjadi tiga bagian independen menggunakan rasio 70:20:10. Alasan utama pemilihan rasio ini adalah untuk menyeimbangkan kebutuhan model akan data pelatihan yang cukup, sekaligus menyediakan data evaluasi yang memadai guna menjamin validitas hasil pengujian. Porsi 70% (315 citra) dialokasikan sebagai sumber pembelajaran utama agar model dapat mencapai titik konvergensi tanpa mengalami *underfitting*. Data validasi sebesar 20% (90 citra) digunakan untuk melakukan *tuning hyperparameter* dan memantau indikasi *overfitting* agar menghasilkan sinyal akurasi yang stabil. Sementara itu, 10% sisanya (45 citra) berfungsi sebagai data pengujian (*unseen data*) untuk mengukur kemampuan generalisasi model secara objektif. Penerapan skema ini secara signifikan meningkatkan validitas penelitian dengan mencegah terjadinya *data leakage*, sehingga hasil evaluasi akhir benar-benar murni dan imparial.

Proses selanjutnya setelah pembagian data adalah *preprocessing* yang meliputi *resize* ke ukuran 256x256 piksel dan augmentasi data untuk memperkaya variasi pada data pelatihan. Teknik augmentasi seperti rotasi, pembalikan citra (*flipping*), serta penyesuaian kecerahan dan kontras diterapkan sebagai upaya strategis untuk meningkatkan kemampuan generalisasi model. Hal ini sangat krusial agar model tetap konsisten dan akurat dalam mendeteksi keretakan meskipun menghadapi kondisi lingkungan yang berbeda di lapangan, seperti sudut pengambilan gambar yang beragam atau intensitas cahaya yang minim. Dengan menciptakan keseimbangan antara *bias* dan *variance* melalui alur kerja yang sistematis ini, setiap langkah berkontribusi langsung pada pencapaian tujuan penelitian, yaitu menghasilkan segmentasi yang presisi serta model yang memiliki kemampuan generalisasi tinggi dan hasil akurasi yang dapat dipertanggung jawabkan secara ilmiah.

Setiap citra asli (*original image*) yang dikumpulkan telah melalui proses anotasi untuk menghasilkan label (*ground truth*) dalam bentuk citra biner. Pada citra label ini, piksel bernilai 1 (putih) merepresentasikan area keretakan, sedangkan piksel bernilai 0 (hitam) merepresentasikan latar belakang dinding. Dengan demikian, total keseluruhan *file* dataset (gabungan citra asli dan label) adalah sebanyak 900 *file*.

Tabel 4.1 berikut menampilkan rincian jumlah dan contoh visual dari dataset yang digunakan.

Tabel 4.1 Contoh dan Jumlah Citra Dataset

No	Nama	Contoh Citra	Jumlah Citra
1	Citra Asli		450
2	Citra Label		450

Tabel 4.1 merupakan contoh representasi pasangan citra yang digunakan sebagai input untuk model U-Net dan SegNet. Proses selanjutnya setelah data terkumpul adalah *preprocessing* yang meliputi *resize* ke ukuran 256x256 piksel dan augmentasi data untuk memperkaya variasi pada *data train*.

Preprocessing dan Pembagian Data Setelah proses pengumpulan data selesai, tahapan selanjutnya sesuai dengan diagram alir sistem adalah *preprocessing*. Pada tahap ini, seluruh citra asli dan citra label (*ground truth*) dimuat dan diubah ukurannya (*resize*) menjadi dimensi tetap 256×256 piksel. Langkah ini dilakukan untuk menyeragamkan resolusi input agar sesuai dengan arsitektur model U-Net dan SegNet, serta untuk mengefisiensikan beban komputasi. Selain itu, nilai piksel pada citra dinormalisasi ke rentang 0 hingga 1 untuk mempercepat konvergensi model saat pelatihan.

Dataset yang digunakan dalam penelitian ini berjumlah 450 citra, yang dibagi ke dalam tiga subset, yaitu data latih (training), data validasi (validation), dan data uji (testing) dengan rasio 70:20:10, sehingga diperoleh masing-masing 315 citra untuk pelatihan, 90 citra untuk validasi, dan 45 citra sebagai data pengujian akhir. Data uji tersebut merupakan data yang benar-benar baru (unseen data) dan tidak digunakan pada tahap pelatihan maupun validasi, sehingga berfungsi untuk mengevaluasi performa akhir model dalam kondisi yang mendekati penerapan nyata. Ke-45 citra ini telah dikurasi untuk merepresentasikan variasi karakteristik

citra, meliputi kondisi dinding bercat dan tanpa cat serta pencahayaan alami dan redup. Selanjutnya, sebagian sampel dari subset data uji digunakan sebagai dataset pengujian spesifik untuk menganalisis ketangguhan model terhadap variasi tersebut, dengan alokasi masing-masing 3 citra representatif pada setiap kategori tingkat keretakan (ringan, sedang dan berat).

Augmentasi Data Untuk meningkatkan keragaman data latih dan mencegah *overfitting*, diterapkan teknik augmentasi data pada 315 citra *training*. Teknik yang digunakan meliputi rotasi (*rotation*) sebesar $\pm 15^\circ$, pembalikan citra (*horizontal/vertical flip*), serta penyesuaian kecerahan (*brightness adjustment*) sebesar $\pm 20\%$. Proses ini bertujuan agar model mampu mengenali pola keretakan dalam berbagai orientasi dan kondisi pencahayaan.

Konfigurasi Pelatihan (Hyperparameter) Sebelum memasuki tahap pelatihan model U-Net dan SegNet, parameter pelatihan atau *hyperparameters* harus ditentukan terlebih dahulu. Konfigurasi ini sangat krusial karena akan mempengaruhi kecepatan konvergensi model dan akurasi yang dihasilkan. Parameter yang digunakan dalam penelitian ini disajikan pada Tabel 4.2 berikut.

Tabel 4.2 Hyperparameter Training

Hyperparameter training	Nilai konfigurasi
Split data	70% training, 20% validation, 10% testing
Input size	256 x 256 piksel
Batch size	8
epoch	50
optimizer	adam
Learning rate	0.0001

Tabel 4.2 merupakan tabel *hyperparameter* yang digunakan dalam proses pelatihan model. *Batch size* yang digunakan sebanyak 8. Hal ini berarti, sebanyak 8 sampel dari citra asli dan citra label akan diproses sekaligus dalam satu iterasi komputasi. Kemudian dengan menggunakan 50 epoch, artinya model akan melakukan 50 kali siklus pelatihan menggunakan seluruh dataset *train* dan *validation*. Model menggunakan *optimizer Adam (Adaptive Moment Estimation)* dengan kecepatan pembelajaran (*learning rate*) *default* sebesar 0.0001, yang dipilih karena efisiensinya dalam menangani data citra yang kompleks.

4.2. Hasil Pengujian

4.2.1. Hasil Pengujian Performa Utama Model U-Net

1. Model Summary Arsitektur U-Net

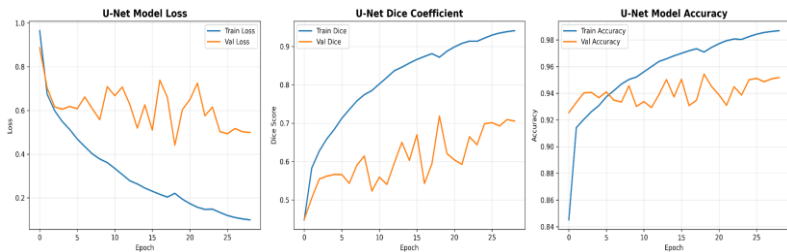
Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 256, 256, 3)	0	-
conv2d (Conv2D)	(None, 256, 256, 64)	1,792	input_layer[0][0]
batch_normalization (BatchNormalization)	(None, 256, 256, 64)	256	conv2d[0][0]
activation (Activation)	(None, 256, 256, 64)	0	batch_normalizat...
conv2d_1 (Conv2D)	(None, 256, 256, 64)	96,928	activation[0][0]
batch_normalization_1 (BatchNormalization)	(None, 256, 256, 64)	256	conv2d_1[0][0]
activation_1 (Activation)	(None, 256, 256, 64)	0	batch_normalizat...
dropout (Dropout)	(None, 256, 256, 64)	0	activation_1[0][0]
max_pooling2d (MaxPooling2D)	(None, 128, 128, 64)	0	dropout[0][0]
conv2d_2 (Conv2D)	(None, 128, 128, 128)	73,856	max_pooling2d[0][0]
batch_normalization_2 (BatchNormalization)	(None, 128, 128, 128)	512	conv2d_2[0][0]
activation_2 (Activation)	(None, 128, 128, 128)	0	batch_normalizat...
conv2d_3 (Conv2D)	(None, 128, 128, 128)	147,504	activation_2[0][0]
batch_normalization_3 (BatchNormalization)	(None, 128, 128, 128)	512	conv2d_3[0][0]
activation_3 (Activation)	(None, 128, 128, 128)	0	batch_normalizat...
dropout_1 (Dropout)	(None, 128, 128, 128)	0	activation_3[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 64, 64, 128)	0	dropout_1[0][0]
conv2d_4 (Conv2D)	(None, 64, 64, 256)	205,184	max_pooling2d_1[0][0]
batch_normalization_4 (BatchNormalization)	(None, 64, 64, 256)	1,024	conv2d_4[0][0]
activation_4 (Activation)	(None, 64, 64, 256)	0	batch_normalizat...
conv2d_5 (Conv2D)	(None, 64, 64, 256)	396,000	activation_4[0][0]
batch_normalization_5 (BatchNormalization)	(None, 64, 64, 256)	1,024	conv2d_5[0][0]
activation_5 (Activation)	(None, 64, 64, 256)	0	batch_normalizat...
dropout_2 (Dropout)	(None, 64, 64, 256)	0	activation_5[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 32, 32, 256)	0	dropout_2[0][0]
conv2d_6 (Conv2D)	(None, 32, 32, 512)	1,100,160	max_pooling2d_2[0][0]
batch_normalization_6 (BatchNormalization)	(None, 32, 32, 512)	2,048	conv2d_6[0][0]
activation_6 (Activation)	(None, 32, 32, 512)	0	batch_normalizat...
conv2d_7 (Conv2D)	(None, 32, 32, 512)	2,350,000	activation_6[0][0]
batch_normalization_7 (BatchNormalization)	(None, 32, 32, 512)	2,048	conv2d_7[0][0]
activation_7 (Activation)	(None, 32, 32, 512)	0	batch_normalizat...
dropout_3 (Dropout)	(None, 32, 32, 512)	0	activation_7[0][0]
max_pooling2d_3 (MaxPooling2D)	(None, 16, 16, 512)	0	dropout_3[0][0]
conv2d_8 (Conv2D)	(None, 16, 16, 1024)	4,710,016	max_pooling2d_3[0][0]
batch_normalization_8 (BatchNormalization)	(None, 16, 16, 1024)	4,096	conv2d_8[0][0]
activation_8 (Activation)	(None, 16, 16, 1024)	0	batch_normalizat...
conv2d_9 (Conv2D)	(None, 16, 16, 1024)	9,430,000	activation_8[0][0]
batch_normalization_9 (BatchNormalization)	(None, 16, 16, 1024)	4,096	conv2d_9[0][0]
activation_9 (Activation)	(None, 16, 16, 1024)	0	batch_normalizat...
dropout_4 (Dropout)	(None, 16, 16, 1024)	0	activation_9[0][0]

dropout_4 (Dropout)	(None, 16, 16, 16)	0	activation_9[0] [-]
conv2d_transpose (Conv2DTranspose)	(None, 32, 32, 2)	2,097,664	dropout_4[0] [0]
concatenate (Concatenate)	(None, 32, 32, 2)	0	conv2d_transpose_dropout_3[0] [0]
conv2d_10 (Conv2D)	(None, 32, 32, 2)	8,719,394	concatenate[0] [0]
batch_normalization (Batch Normalization)	(None, 32, 32, 2)	2,098	conv2d_10[0] [0]
activation_10 (Activation)	(None, 32, 32, 2)	0	batch_normalizat...
conv2d_11 (Conv2D)	(None, 32, 32, 2)	2,359,880	activation_10[0] [-]
batch_normalization (Batch Normalization)	(None, 32, 32, 2)	2,098	conv2d_11[0] [0]
activation_11 (Activation)	(None, 32, 32, 2)	0	batch_normalizat...
dropout_5 (Dropout)	(None, 32, 32, 2)	0	activation_11[0] [-]
conv2d_transpose_1 (Conv2DTranspose)	(None, 64, 64, 256)	524,594	dropout_5[0] [0]
concatenate_1 (Concatenate)	(None, 64, 64, 256)	0	conv2d_transpose_dropout_2[0] [0]
conv2d_12 (Conv2D)	(None, 64, 64, 256)	1,179,984	concatenate_1[0] [-]
batch_normalization (Batch Normalization)	(None, 64, 64, 256)	1,024	conv2d_12[0] [0]
activation_12 (Activation)	(None, 64, 64, 256)	0	batch_normalizat...
conv2d_13 (Conv2D)	(None, 64, 64, 256)	590,000	activation_12[0] [-]
batch_normalization (Batch Normalization)	(None, 64, 64, 256)	1,024	conv2d_13[0] [0]
activation_13 (Activation)	(None, 64, 64, 256)	0	batch_normalizat...
dropout_6 (Dropout)	(None, 64, 64, 256)	0	activation_13[0] [-]
conv2d_transpose_2 (Conv2DTranspose)	(None, 128, 128, 2)	131,200	dropout_6[0] [0]
concatenate_2 (Concatenate)	(None, 128, 128, 256)	0	conv2d_transpose_dropout_1[0] [0]
conv2d_14 (Conv2D)	(None, 128, 128, 256)	205,040	concatenate_2[0] [-]
batch_normalization (Batch Normalization)	(None, 128, 128, 256)	512	conv2d_14[0] [0]
activation_14 (Activation)	(None, 128, 128, 256)	0	batch_normalizat...
conv2d_15 (Conv2D)	(None, 128, 128, 256)	147,584	activation_14[0] [-]
batch_normalization (Batch Normalization)	(None, 128, 128, 256)	512	conv2d_15[0] [0]
activation_15 (Activation)	(None, 128, 128, 256)	0	batch_normalizat...
dropout_7 (Dropout)	(None, 128, 128, 256)	0	activation_15[0] [-]
conv2d_transpose_3 (Conv2DTranspose)	(None, 256, 256, 64)	32,832	dropout_7[0] [0]
concatenate_3 (Concatenate)	(None, 256, 256, 128)	0	conv2d_transpose_dropout[0] [0]
conv2d_16 (Conv2D)	(None, 256, 256, 128)	73,292	concatenate_3[0] [-]
batch_normalization (Batch Normalization)	(None, 256, 256, 128)	256	conv2d_16[0] [0]
activation_16 (Activation)	(None, 256, 256, 128)	0	batch_normalizat...
conv2d_17 (Conv2D)	(None, 256, 256, 128)	36,928	activation_16[0] [-]
batch_normalization (Batch Normalization)	(None, 256, 256, 128)	256	conv2d_17[0] [0]
activation_17 (Activation)	(None, 256, 256, 128)	0	batch_normalizat...
dropout_8 (Dropout)	(None, 256, 256, 128)	0	activation_17[0] [-]
conv2d_18 (Conv2D)	(None, 256, 256, 128)	65	dropout_8[0] [0]
Total params: 31,055,292 (118.47 MB)			
Trainable params: 31,043,521 (118.42 MB)			
Non-trainable params: 11,776 (46.00 KB)			
Total Parameters: 31,055,292			
Trainable Parameters: 31,043,521			
Non-trainable Parameters: 11,776			

Gambar 4.1 Model Summary Arsitektur Model U-Net

Gambar 4.1 menampilkan ringkasan arsitektur model U-Net yang telah dibangun dan diuji dalam penelitian ini. Model diawali dengan *input layer* berdimensi 256×256 piksel dengan 3 *channel* warna (RGB), yang kemudian diproses melalui serangkaian lapisan *encoder* menggunakan kombinasi *Conv2D*, *BatchNormalization*, *Activation*, *Dropout*, dan *MaxPooling2D* hingga mencapai kedalaman fitur maksimal sebesar 1.024 filter. Fitur-fitur tersebut selanjutnya direkonstruksi kembali melalui jalur *decoder* menggunakan *Conv2DTranspose* dan *Concatenate* (*skip connection*), sebelum akhirnya menghasilkan *output* citra segmentasi berukuran 256×256 piksel dengan 1 *channel* (biner) yang memiliki total parameter latih (*trainable parameters*) sebanyak 31.043.521.

2. Grafik Pelatihan Arsitektur U-Net



Gambar 4.2 Grafik Pelatihan Arsitektur U-Net

Gambar 4.2 merupakan grafik hasil *training* model U-Net yang mencakup metrik *Loss*, *Dice Coefficient*, dan *Accuracy*. Berdasarkan grafik tersebut, dapat dilihat bahwa *training loss* (garis biru) mengalami penurunan yang signifikan dan konsisten, sementara *training accuracy* mengalami kenaikan hingga mendekati nilai sempurna. Namun, untuk hasil pada data validasi (garis oranye), terlihat grafik mengalami fluktuasi yang cukup sering naik-turun baik pada *loss* maupun akurasi. Hal ini mengindikasikan bahwa meskipun model sangat baik dalam mempelajari data latih, model masih belum cukup stabil dalam memvalidasi data baru yang diberikan, ditandai dengan adanya celah (*gap*) kinerja antara data *training* dan *validation*. Meskipun demikian, fluktuasi yang terjadi pada kurva validasi tidak hanya menunjukkan perbedaan kinerja antara data *training* dan *validation*, tetapi juga mengindikasikan keterbatasan kemampuan generalisasi model U-Net terhadap data yang tidak dilatih secara langsung. Pola naik-turun pada nilai *loss* dan akurasi validasi merupakan indikasi awal adanya kecenderungan *overfitting*, yang dipengaruhi oleh kompleksitas tekstur permukaan dinding, variasi pencahayaan, serta keterbatasan jumlah dataset. Meskipun demikian, mengingat performa model pada data uji tetap menunjukkan nilai *IoU* dan *F1-Score* yang tinggi, indikasi *overfitting* tersebut masih berada dalam batas yang dapat diterima untuk skala penelitian ini.

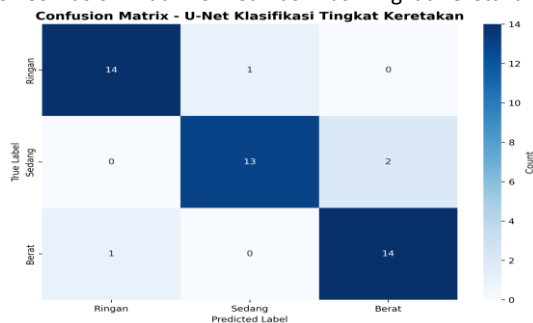
3. Hasil Pengujian Arsitektur U-Net Pada Sampel Data Uji



Gambar 4.3 Hasil Pengujian Arsitektur U-Net Pada Sampel

Gambar 4.3 merupakan perbandingan antara citra asli di sebelah kiri dengan citra *ground truth* dan hasil prediksi model di sebelah kanan. Citra asli menampilkan objek keretakan dinding kategori berat dalam bentuk dan warna aslinya, sedangkan citra prediksi menunjukkan hasil pemrosesan segmentasi oleh arsitektur U-Net. Pada gambar prediksi, area yang terdeteksi sebagai keretakan ditampilkan dalam warna putih, sedangkan area yang tidak terdeteksi diklasifikasikan sebagai latar belakang yang ditampilkan dalam warna hitam. Hasil tersebut menunjukkan performa yang sangat baik dengan nilai IoU mencapai 0.963, di mana proporsi area retak hasil prediksi (30.17%) sangat mendekati proporsi area retak yang sebenarnya (31.03%). Nilai Intersection over Union (IoU) sebesar 0,963 menunjukkan tingkat kesesuaian yang sangat tinggi antara area keretakan hasil prediksi model U-Net dengan ground truth. Secara praktis, capaian ini mengindikasikan bahwa sistem mampu memetakan lokasi dan bentuk retakan secara presisi, sehingga dapat dimanfaatkan sebagai alat bantu inspeksi visual awal dalam mendeteksi area kerusakan bangunan. Dengan tingkat tumpang tindih prediksi yang tinggi, sistem berpotensi mengurangi subjektivitas inspeksi manual serta mempercepat proses identifikasi keretakan sebelum dilakukan evaluasi struktural lanjutan oleh tenaga ahli.

4. Hasil Confusion Matrix U-Net Klasifikasi Tingkat Keretakan



Gambar 4.4 Hasil Confusion Matrix U-Net Klasifikasi Tingkat Keretakan

Gambar 4.4 menampilkan *Confusion Matrix* dari hasil klasifikasi tingkat keretakan menggunakan model U-Net pada data uji. Matriks ini memetakan perbandingan antara kategori sebenarnya (*True Label*) pada sumbu vertikal dengan kategori hasil prediksi model (*Predicted Label*) pada sumbu horizontal untuk tiga kelas: Ringan, Sedang, dan Berat. Berdasarkan visualisasi tersebut, terlihat bahwa model memiliki akurasi klasifikasi yang tinggi, ditandai dengan konsentrasi nilai terbesar yang berada pada garis diagonal utama. Secara spesifik, model berhasil memprediksi dengan tepat 14 dari 15 citra pada kategori Ringan dan Berat, serta 13 dari 15 citra pada kategori Sedang, dengan hanya sedikit kesalahan klasifikasi (*misclassification*) yang terjadi pada kelas yang berdekatan.

Tabel 4.3 Classification Report Arsitektur U-Net pada Data Uji

Kelas	Precision	Recall	F1-Score	Accuracy%	Support
Ringan	93%	93%	93%	91%	15
Sedang	93%	87%	90%		15
Berat	88%	96%	90%		15
Macro Average	91%	91%	91%		45
Weighted Average	91%	91%	91%		45

Tabel 4.3 menyajikan rekapitulasi hasil evaluasi performa model U-Net pada data uji yang diklasifikasikan ke dalam tiga kategori tingkat keretakan, yaitu ringan, sedang, dan berat. Berdasarkan tabel tersebut, model U-Net menunjukkan kinerja yang sangat baik dengan nilai *accuracy* sebesar 91% pada total 45 data uji. Untuk masing-masing kelas, diperoleh nilai *F1-Score* sebesar 93% pada kelas ringan, 90% pada kelas sedang, dan 90% pada kelas berat. Nilai *macro average* dan *weighted average* untuk *precision*, *recall*, dan *F1-Score* masing-masing berada pada angka 91%, yang menandakan bahwa performa model relatif seimbang antar kelas.

Secara keseluruhan, hasil ini menunjukkan bahwa model U-Net mampu melakukan klasifikasi tingkat keretakan dengan sangat baik dan konsisten pada data uji.

Guna memverifikasi validitas nilai persentase di atas sesuai dengan catatan pengujian, dilakukan perhitungan manual berdasarkan data *confusion matrix* menggunakan persamaan yang telah dijabarkan pada Bab 3. Sebagai contoh representatif, berikut adalah rincian perhitungan nilai Akurasi *confusion matrix* dan Presisi untuk kategori Retak Ringan:

1. Perhitungan Akurasi: Berdasarkan hasil pengujian, jumlah data uji yang digunakan sebanyak 45 citra, dengan total prediksi benar sebanyak 41 citra. Nilai akurasi dihitung sebagai perbandingan antara jumlah prediksi benar terhadap jumlah keseluruhan data uji, sehingga diperoleh nilai akurasi sebesar:

$$\text{Akurasi} = \frac{41}{45} = 0,9111 = 91\%$$

2. Perhitungan Presisi: Berdasarkan hasil pengujian, jumlah prediksi benar (*True Positive*) yang diperoleh adalah sebanyak 14 citra, dengan jumlah kesalahan prediksi positif (*False Positive*) sebanyak 1 citra. Nilai presisi dihitung sebagai perbandingan antara jumlah prediksi benar terhadap total data yang diprediksi sebagai kategori tersebut (penjumlahan *True Positive* dan *False Positive*), sehingga diperoleh nilai presisi Retak Ringan sebesar:

$$\text{Presisi} = \frac{14}{14+1} = 0,9333 = 93\%$$

Hasil perhitungan manual ini membuktikan bahwa nilai performa yang dihasilkan oleh sistem konsisten dengan prinsip perhitungan evaluasi model klasifikasi.

Tabel 4.4 Hasil Metrik Evaluasi Segmentasi Arsitektur U-Net pada Data Uji

Mean IoU	Mean Pixel Accuracy	Mean Precision	Mean Recall	Mean F1-Score	Mean Inference Time
63.56%	95.38%	84.12%	72.36%	73.42%	271.62 ms

Tabel 4.4 memperlihatkan rincian data hasil pengukuran performa model U-Net terhadap data uji. Tabel ini menyajikan nilai capaian untuk lima parameter evaluasi utama, yaitu *Accuracy*, *Precision*, *Recall*, *F1-Score*, dan *IoU* yang dikelompokkan berdasarkan tiga kategori keretakan (Ringan, Sedang, dan Berat) dengan jumlah sampel masing-masing sebanyak 15 data. Selain metrik performa segmentasi, tabel ini juga mencantumkan informasi mengenai waktu inferensi

untuk setiap kategori serta nilai rata-rata keseluruhan (*average*) dari seluruh parameter pengujian yang dilakukan.

Tabel 4.5 Hasil Distribusi Data Prediksi vs Ground Truth pada Data Uji

Kategori Kelas	Ground Truth	Prediksi
Ringan	15 (33.33%)	15 (33.33%)
Sedang	15 (33.33%)	14 (31.1%)
Berat	15 (33.33%)	16 (35.6%)

Tabel 4.5 menunjukkan perbandingan distribusi kelas antara Ground Truth dan hasil prediksi model pada data uji dengan tiga kategori, yaitu ringan, sedang, dan berat. Pada Ground Truth, ketiga kelas memiliki distribusi yang seimbang sebesar 33,33% untuk masing-masing kelas. Hasil prediksi menunjukkan bahwa kelas ringan memiliki proporsi 33,3% yang hampir sama dengan Ground Truth, kelas sedang sedikit menurun menjadi 31,1%, dan kelas berat meningkat menjadi 35,6%. Perbedaan ini mengindikasikan adanya kecenderungan kecil model dalam memprediksi kelas berat, namun secara keseluruhan distribusi prediksi masih mendekati Ground Truth, sehingga model dapat dikatakan mampu menjaga keseimbangan klasifikasi antar kelas pada data uji.

4.2.2. Hasil Pengujian Model U-Net dengan Variasi Permukaan Dinding

Tabel 4.6 Evaluasi Berdasarkan Variasi Permukaan Dinding

N o	Kategori Keretakan	Kondisi Permukaan	Accuracy	Precision	Recall	IoU (%)	F1- Score (%)	Waktu Inferensi (s)
1	Ringan	Dinding Bercat	98.98 %	88.13 %	66.18 %	60.96 %	75.49 %	0.86 11s
		Tanpa Cat (Semen)	98.05 %	68.31 %	8.22 %	6.82 %	12.71 %	0.73 72s
2	Sedang	Dinding Bercat	98.83 %	95.10 %	94.60 %	90.01 %	94.73 %	0.86 60s
		Tanpa Cat (Semen)	93.06 %	85.04 %	75.23 %	65.86 %	79.34 %	0.73 15s
3	Berat	Dinding Bercat	94.63 %	97.04 %	88.33 %	85.97 %	92.36 %	0.78 15s

		Tanpa Cat (Semen)	78.28 %	93.27 %	16.04 %	15.17 %	24.39 %	0.72 06s
--	--	-------------------	---------	---------	---------	---------	---------	----------

Berdasarkan Tabel 4.6, model U-Net secara konsisten menunjukkan performa yang lebih baik pada permukaan dinding bercat dibandingkan dinding tanpa cat, terutama pada metrik Recall, F1-Score, dan IoU di seluruh kategori tingkat keretakan. Hal ini menunjukkan bahwa model lebih mampu mendeteksi dan memetakan area retakan secara presisi pada permukaan yang bercat. Secara visual, dinding bercat memiliki tekstur yang lebih halus dan homogen, sehingga retakan tampil dengan kontras yang lebih jelas terhadap latar belakang, yang memudahkan model dalam membedakan pola retakan dari noise permukaan. Sebaliknya, pada dinding tanpa cat atau semen ekspos yang memiliki tekstur kasar dan tidak seragam, performa model cenderung menurun signifikan, khususnya pada kategori retak ringan dan berat, karena pola tekstur permukaan sering menyerupai bentuk retakan sehingga meningkatkan noise visual dan menyebabkan rendahnya nilai Recall dan IoU. Dalam konteks aplikasi dunia nyata, temuan ini mengindikasikan bahwa sistem deteksi keretakan berbasis U-Net lebih andal pada bangunan dengan permukaan dinding halus, sementara pada kondisi dinding kasar diperlukan adaptasi tambahan, seperti memperkaya data latih pada permukaan semen ekspos, menerapkan augmentasi tekstur yang lebih agresif, atau preprocessing untuk menekan noise visual, agar sistem tetap memberikan hasil yang optimal di lapangan.

4.2.3. Hasil Pengujian Model U-Net dengan Variasi Pencahayaan

Tabel 4.7 Evaluasi Berdasarkan Variasi Pencahayaan

No	Kategori Keretakan	Kondisi Permukaan	Accuracy	Precision	Recall	IoU (%)	F1-Score (%)	Waktu Inferensi (s)
1	Ringan	Alami (Terang)	98.19 %	90.89 %	4.51 %	4.47 %	8.53 %	0.68 19s
		Redup (Indoor)	98.72 %	58.97 %	84.51 %	53.21 %	69.46 %	0.83 19s
2	Sedang	Alami (Terang)	98.00 %	89.90 %	85.84 %	79.98 %	87.81 %	0.73 06s
		Redup (Indoor)	98.65 %	92.50 %	92.09 %	85.42 %	92.04 %	0.98 13s
3	Berat	Alami (Terang)	89.97 %	88.07 %	78.19 %	73.13 %	82.69 %	0.80 92s

		Redup (Indoor)	81.04 %	94.04 %	37.08 %	36.64 %	52.32 %	0.87 05s
--	--	-------------------	------------	------------	------------	------------	------------	-------------

Berdasarkan Tabel 4.7, terlihat bahwa kondisi pencahayaan memiliki pengaruh signifikan terhadap performa model U-Net, khususnya pada metrik Recall. Pada beberapa kategori, terutama retak ringan dan berat, kondisi pencahayaan redup menunjukkan penurunan nilai Recall yang cukup drastis dibandingkan kondisi pencahayaan alami, yang mengindikasikan bahwa model berpotensi melewatkan sebagian area retakan kecil ketika intensitas cahaya rendah. Hal ini terjadi karena pada kondisi low-light, kontras antara retakan dan permukaan dinding menurun, sehingga pola retakan menjadi kurang jelas dan sulit dibedakan dari latar belakang. Recall menjadi metrik yang sangat krusial dalam konteks ini karena secara langsung merepresentasikan kemampuan model dalam menemukan seluruh area keretakan yang ada; nilai Recall yang rendah menunjukkan risiko terjadinya false negative, yang berbahaya dalam aplikasi inspeksi bangunan karena keretakan dapat terlewat. Untuk meningkatkan sensitivitas model pada kondisi pencahayaan redup, diperlukan adaptasi seperti penambahan augmentasi data yang mensimulasikan kondisi low-light secara lebih agresif, peningkatan kontras citra pada tahap preprocessing, serta penyesuaian strategi pelatihan agar model lebih robust terhadap variasi intensitas cahaya. Dengan pendekatan tersebut, performa model diharapkan dapat lebih stabil dan andal ketika diterapkan pada kondisi lapangan dengan pencahayaan yang tidak ideal.

4.2.4. Hasil Pengujian Performa Model Utama SegNet

1. Model Summary Arsitektur SegNet

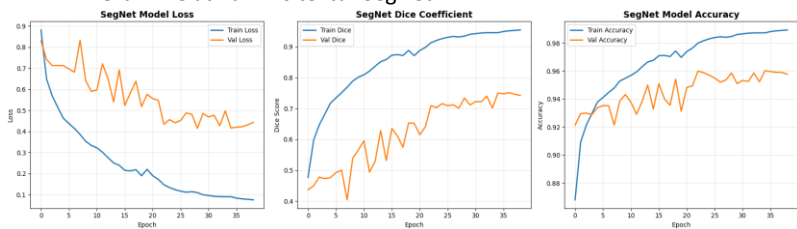
Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 256, 256, 3)	0
conv2d (Conv2D)	(None, 256, 256, 64)	1,792
batch_normalization (BatchNormalization)	(None, 256, 256, 64)	256
activation (Activation)	(None, 256, 256, 64)	0
conv2d_1 (Conv2D)	(None, 256, 256, 64)	36,928
batch_normalization_1 (BatchNormalization)	(None, 256, 256, 64)	256
activation_1 (Activation)	(None, 256, 256, 64)	0
max_pooling2d (MaxPooling2D)	(None, 128, 128, 64)	0
conv2d_2 (Conv2D)	(None, 128, 128, 128)	73,856
batch_normalization_2 (BatchNormalization)	(None, 128, 128, 128)	512
activation_2 (Activation)	(None, 128, 128, 128)	0
conv2d_3 (Conv2D)	(None, 128, 128, 128)	147,584
batch_normalization_3 (BatchNormalization)	(None, 128, 128, 128)	512
activation_3 (Activation)	(None, 128, 128, 128)	0
max_pooling2d_1 (MaxPooling2D)	(None, 64, 64, 128)	0
conv2d_4 (Conv2D)	(None, 64, 64, 256)	295,168
batch_normalization_4 (BatchNormalization)	(None, 64, 64, 256)	1,024
activation_4 (Activation)	(None, 64, 64, 256)	0
conv2d_5 (Conv2D)	(None, 64, 64, 256)	590,080
batch_normalization_5 (BatchNormalization)	(None, 64, 64, 256)	1,024
activation_5 (Activation)	(None, 64, 64, 256)	0
conv2d_6 (Conv2D)	(None, 64, 64, 256)	590,080
batch_normalization_6 (BatchNormalization)	(None, 64, 64, 256)	1,024
activation_6 (Activation)	(None, 64, 64, 256)	0
max_pooling2d_2 (MaxPooling2D)	(None, 32, 32, 256)	0
conv2d_7 (Conv2D)	(None, 32, 32, 512)	1,180,160
batch_normalization_7 (BatchNormalization)	(None, 32, 32, 512)	2,048
activation_7 (Activation)	(None, 32, 32, 512)	0
conv2d_8 (Conv2D)	(None, 32, 32, 512)	2,359,008
batch_normalization_8 (BatchNormalization)	(None, 32, 32, 512)	2,048
activation_8 (Activation)	(None, 32, 32, 512)	0
conv2d_9 (Conv2D)	(None, 32, 32, 512)	2,359,008
batch_normalization_9 (BatchNormalization)	(None, 32, 32, 512)	2,048
activation_9 (Activation)	(None, 32, 32, 512)	0
dropout (Dropout)	(None, 32, 32, 512)	0
max_pooling2d_3 (MaxPooling2D)	(None, 16, 16, 512)	0
conv2d_10 (Conv2D)	(None, 16, 16, 512)	2,359,008
batch_normalization_10 (BatchNormalization)	(None, 16, 16, 512)	2,048
activation_10 (Activation)	(None, 16, 16, 512)	0
conv2d_11 (Conv2D)	(None, 16, 16, 512)	2,359,008
batch_normalization_11 (BatchNormalization)	(None, 16, 16, 512)	2,048
activation_11 (Activation)	(None, 16, 16, 512)	0
dropout_1 (Dropout)	(None, 16, 16, 512)	0
up_sampling2d (UpSampling2D)	(None, 32, 32, 512)	0
conv2d_12 (Conv2D)	(None, 32, 32, 512)	2,359,008
batch_normalization_12 (BatchNormalization)	(None, 32, 32, 512)	2,048

activation_12 (Activation)	(None, 32, 32, 512)	0
conv2d_13 (Conv2D)	(None, 32, 32, 512)	2,359,808
batch_normalization_13 (BatchNormalization)	(None, 32, 32, 512)	2,048
activation_13 (Activation)	(None, 32, 32, 512)	0
conv2d_14 (Conv2D)	(None, 32, 32, 512)	2,359,808
batch_normalization_14 (BatchNormalization)	(None, 32, 32, 512)	2,048
activation_14 (Activation)	(None, 32, 32, 512)	0
dropout_2 (Dropout)	(None, 32, 32, 512)	0
up_sampling2d_1 (UpSampling2D)	(None, 64, 64, 512)	0
conv2d_15 (Conv2D)	(None, 64, 64, 256)	1,179,904
batch_normalization_15 (BatchNormalization)	(None, 64, 64, 256)	1,024
activation_15 (Activation)	(None, 64, 64, 256)	0
conv2d_16 (Conv2D)	(None, 64, 64, 256)	590,080
batch_normalization_16 (BatchNormalization)	(None, 64, 64, 256)	1,024
activation_16 (Activation)	(None, 64, 64, 256)	0
conv2d_17 (Conv2D)	(None, 64, 64, 256)	590,080
batch_normalization_17 (BatchNormalization)	(None, 64, 64, 256)	1,024
activation_17 (Activation)	(None, 64, 64, 256)	0
up_sampling2d_2 (UpSampling2D)	(None, 128, 128, 256)	0
conv2d_18 (Conv2D)	(None, 128, 128, 128)	295,040
batch_normalization_18 (BatchNormalization)	(None, 128, 128, 128)	512
activation_18 (Activation)	(None, 128, 128, 128)	0
conv2d_19 (Conv2D)	(None, 128, 128, 128)	147,584
batch_normalization_19 (BatchNormalization)	(None, 128, 128, 128)	512
activation_19 (Activation)	(None, 128, 128, 128)	0
up_sampling2d_3 (UpSampling2D)	(None, 256, 256, 128)	0
conv2d_20 (Conv2D)	(None, 256, 256, 64)	73,792
batch_normalization_20 (BatchNormalization)	(None, 256, 256, 64)	256
activation_20 (Activation)	(None, 256, 256, 64)	0
conv2d_21 (Conv2D)	(None, 256, 256, 64)	36,928
batch_normalization_21 (BatchNormalization)	(None, 256, 256, 64)	256
activation_21 (Activation)	(None, 256, 256, 64)	0
conv2d_22 (Conv2D)	(None, 256, 256, 1)	65
Total params: 22,373,377 (85.35 MB)		
Trainable params: 22,360,577 (85.30 MB)		
Non-trainable params: 12,800 (50.00 KB)		

Gambar 4.5 Model Summary Arsitektur Model SegNet

Gambar 4.5 menampilkan ringkasan arsitektur model SegNet yang digunakan dalam penelitian. Arsitektur ini diawali dengan *input layer* yang menerima citra berukuran 256×256 piksel dengan 3 *channel* warna (RGB). Proses pengolahan data melewati serangkaian lapisan *encoder* yang terdiri dari *Conv2D*, *BatchNormalization*, *Activation*, dan *MaxPooling2D* untuk mengekstraksi fitur hingga mencapai kedalaman filter 512. Selanjutnya, pada tahap *decoder*, dimensi citra dipulihkan kembali menggunakan lapisan *UpSampling2D* yang dikombinasikan dengan lapisan konvolusi untuk merekonstruksi detail spasial. Output akhir dari model ini adalah lapisan konvolusi tunggal yang menghasilkan citra segmentasi biner berukuran 256×256 piksel, dengan total parameter keseluruhan sebanyak 22.373.377

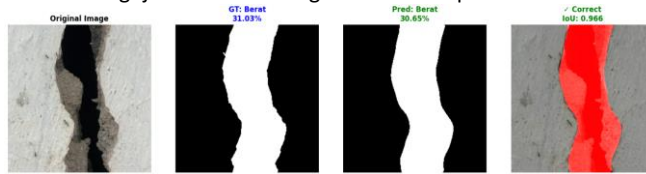
2. Grafik Pelatihan Arsitektur SegNet



Gambar 4.6 Grafik Pelatihan Arsitektur SegNet

Gambar 4.6 merupakan grafik hasil *training* model SegNet yang mencakup metrik *Loss*, *Dice Coefficient*, dan *Accuracy*. Berdasarkan grafik tersebut, terlihat bahwa pada data *training* (garis biru), kurva menunjukkan tren yang sangat baik dengan penurunan *loss* yang mulus dan peningkatan akurasi yang stabil. Sebaliknya, pada data validasi (garis oranye), grafik memperlihatkan pola fluktuasi yang sangat ekstrem dan tidak stabil sepanjang proses *epoch*. Meskipun model SegNet menunjukkan proses pembelajaran yang sangat baik pada data *training*, ditandai dengan penurunan nilai *loss* yang mulus serta peningkatan akurasi yang stabil, grafik pada data validasi memperlihatkan fluktuasi yang tajam dan tidak stabil sepanjang *epoch*. Pola osilasi pada nilai *loss* dan akurasi validasi ini mengindikasikan bahwa model mengalami kesulitan dalam mempertahankan konsistensi generalisasi terhadap data baru. Kondisi tersebut menunjukkan adanya kecenderungan *overfitting*, yang kemungkinan dipengaruhi oleh kompleksitas tekstur permukaan dinding, variasi pencahayaan, serta keterbatasan jumlah dataset. Meskipun demikian, hasil evaluasi pada data uji tetap menunjukkan performa yang layak, sehingga indikasi *overfitting* yang terjadi masih berada dalam batas yang dapat diterima untuk lingkup penelitian ini.

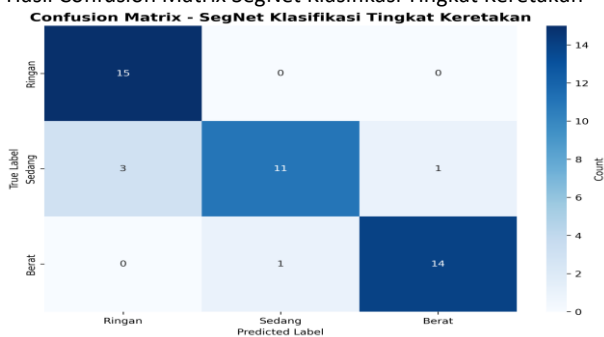
3. Hasil Pengujian Arsitektur SegNet Pada Sampel



Gambar 4.7 Hasil Pengujian Arsitektur SegNet Pada Sampel

Gambar 4.7 merupakan perbandingan antara citra asli di sebelah kiri dengan citra *ground truth* dan hasil prediksi model SegNet di sebelah kanan. Citra asli menampilkan objek keretakan dinding kategori berat, sedangkan citra prediksi menunjukkan hasil pemrosesan segmentasi oleh arsitektur SegNet. Pada gambar prediksi, area yang terdeteksi sebagai keretakan ditampilkan dalam warna putih, sedangkan area lainnya diklasifikasikan sebagai latar belakang berwarna hitam. Hasil tersebut menunjukkan performa dengan nilai IoU sebesar 0.966, di mana proporsi area retak hasil prediksi (30.65%) terukur lebih besar dibandingkan proporsi area retak yang sebenarnya (31.03%), mengindikasikan bahwa model SegNet cenderung melakukan prediksi berlebih (*over-segmentation*) dibandingkan area aslinya. Nilai Intersection over Union (IoU) sebesar 0,966 pada model SegNet menunjukkan tingkat kesesuaian yang cukup baik antara hasil segmentasi dan *ground truth*, namun masih terdapat perbedaan luasan area retakan yang terdeteksi. Berdasarkan perbandingan proporsi area retak, model SegNet cenderung menghasilkan prediksi yang lebih luas dibandingkan kondisi sebenarnya, yang mengindikasikan terjadinya *over-segmentation*. Dalam konteks implementasi praktis, karakteristik ini menunjukkan bahwa SegNet lebih sensitif dalam mendeteksi area keretakan, sehingga berpotensi sesuai digunakan sebagai sistem penyaring awal (*screening*) untuk mengidentifikasi area yang dicurigai mengalami kerusakan, meskipun masih memerlukan verifikasi lanjutan untuk meningkatkan presisi batas retakan.

4. Hasil Confusion Matrix SegNet Klasifikasi Tingkat Keretakan



Gambar 4.8 Hasil Confusion Matrix SegNet Klasifikasi Tingkat Keretakan

Gambar 4.8 menampilkan *Confusion Matrix* dari hasil klasifikasi tingkat keretakan menggunakan model SegNet pada data uji. Matriks ini memetakan perbandingan antara kategori sebenarnya (*True Label*) dengan kategori hasil prediksi model (*Predicted Label*). Berdasarkan visualisasi tersebut, terlihat bahwa model memiliki performa yang cukup baik dengan konsentrasi data tertinggi berada pada garis diagonal utama. Secara spesifik, model berhasil mengklasifikasikan dengan tepat seluruh 15 citra pada kategori Ringan, 11 citra pada kategori Sedang, dan 14 citra pada kategori Berat. Meskipun demikian, masih terdapat kesalahan klasifikasi (*misclassification*) yang terlihat di luar diagonal utama, terutama pada kategori Sedang di mana 3 citra terdeteksi sebagai Ringan dan 1 citra terdeteksi sebagai Berat.

Tabel 4.8 Classification Report Arsitektur SegNet pada Data Uji

Kelas	Precision	Recall	F1-Score	Accuracy%	Support
Ringan	83%	100%	91%	89%	15
Sedang	92%	73%	81%		15
Berat	93%	93%	93%		15
Macro Average	89%	89%	89%		45
Weighted Average	89%	89%	89%		45

Tabel 4.8 menyajikan rekapitulasi hasil evaluasi performa model SegNet pada data uji yang diklasifikasikan ke dalam tiga kategori tingkat keretakan, yaitu ringan, sedang, dan berat. Berdasarkan tabel tersebut, model SegNet menunjukkan kinerja yang baik dengan nilai accuracy sebesar 89% pada total 45 data uji. Untuk masing-masing kelas, diperoleh nilai F1-Score sebesar 91% pada kelas ringan, 81%

pada kelas sedang, dan 93% pada kelas berat. Nilai macro average dan weighted average untuk precision, recall, dan F1-Score masing-masing berada pada angka 89%, yang menandakan bahwa performa model relatif seimbang antar kelas. Secara keseluruhan, hasil ini menunjukkan bahwa model SegNet mampu melakukan klasifikasi tingkat keretakan dengan baik dan konsisten pada data uji.

Guna memverifikasi validitas nilai persentase di atas sesuai dengan catatan pengujian, dilakukan perhitungan manual berdasarkan data *confusion matrix* menggunakan persamaan yang telah dijabarkan pada Bab 3. Sebagai contoh representatif, berikut adalah rincian perhitungan nilai Akurasi *confusion matrix* dan Presisi untuk kategori Retak Ringan:

1. Perhitungan Akurasi: Berdasarkan hasil pengujian, jumlah data uji yang digunakan sebanyak 45 citra, dengan total prediksi benar sebanyak 40 citra. Nilai akurasi dihitung sebagai perbandingan antara jumlah prediksi benar terhadap jumlah keseluruhan data uji, sehingga diperoleh nilai akurasi sebesar:

$$Akurasi = \frac{40}{45} = 0,8888 = 89\%$$

2. Perhitungan Presisi: Berdasarkan hasil pengujian, jumlah prediksi benar (*True Positive*) yang diperoleh adalah sebanyak 15 citra, dengan jumlah kesalahan prediksi positif (*False Positive*) sebanyak 3 citra. Nilai presisi dihitung sebagai perbandingan antara jumlah prediksi benar terhadap total data yang diprediksi sebagai kategori tersebut (penjumlahan *True Positive* dan *False Positive*), sehingga diperoleh nilai presisi Retak Ringan sebesar:

$$Presisi = \frac{15}{15+3} = 0,8333 = 83\%$$

Hasil perhitungan manual ini membuktikan bahwa nilai performa yang dihasilkan oleh sistem konsisten dengan prinsip perhitungan evaluasi model klasifikasi.

Tabel 4.9 Hasil Metrik Evaluasi Segmentasi Arsitektur SegNet pada Data Uji

Mean IoU	Mean Pixel Accuracy	Mean Precision	Mean Recall	Mean F1-Score	Mean Inference Time
60.05%	94.35%	79.20%	72.23%	72.52%	216.88 ms

Tabel 4.9 memperlihatkan rincian data hasil pengukuran performa model SegNet terhadap data uji. Tabel ini menyajikan nilai capaian untuk lima parameter evaluasi utama, yaitu *Accuracy*, *Precision*, *Recall*, *F1-Score*, dan *IoU* yang dikelompokkan berdasarkan tiga kategori keretakan (Ringan, Sedang, dan Berat)

dengan jumlah sampel masing-masing sebanyak 15 data. Selain metrik performa segmentasi, tabel ini juga mencantumkan informasi mengenai waktu inferensi untuk setiap kategori serta nilai rata-rata keseluruhan (*average*) dari seluruh parameter pengujian yang dilakukan.

Tabel 4.10 Hasil Distribusi Data Prediksi vs Ground Truth pada Data Uji

Kategori Kelas	Ground Truth	Prediksi
Ringan	15 (33.33%)	18 (40.0%)
Sedang	15 (33.33%)	12 (26.7%)
Berat	15 (33.33%)	15 (33.3%)

Tabel 4.10 menunjukkan perbandingan distribusi kelas antara Ground Truth dan hasil prediksi model pada data uji dengan tiga kategori, yaitu ringan, sedang, dan berat. Pada Ground Truth, ketiga kelas memiliki distribusi yang seimbang sebesar 33,33% untuk masing-masing kelas. Hasil prediksi menunjukkan bahwa kelas ringan memiliki proporsi 40,0% sedangkan kelas sedang sedikit menurun menjadi 26,7%, dan kelas berat tetap di 33,33%. Perbedaan ini menunjukkan bahwa model cenderung lebih sering memprediksi kelas ringan, yang terlihat dari peningkatan proporsi menjadi 40,0%, sementara kelas sedang mengalami penurunan proporsi menjadi 26,7%. Adapun kelas berat diprediksi dengan proporsi yang sama seperti Ground Truth, yaitu 33,33%. Meskipun terdapat pergeseran kecil pada kelas ringan dan sedang, secara umum distribusi hasil prediksi masih relatif mendekati distribusi Ground Truth, sehingga model dapat dikatakan cukup mampu mempertahankan keseimbangan klasifikasi antar kelas pada data uji.

4.2.5. Hasil Pengujian Model SegNet dengan Variasi Permukaan Dinding

Tabel 4.11 Evaluasi Berdasarkan Variasi Permukaan Dinding

No	Kategori Kereta	Kondisi Permukaan	Accuracy	Precision	Recall	IoU (%)	F1-Score (%)	Waktu Inferensi (s)
1	Ringan	Dinding Bercat	98.48 %	68.62 %	65.0 4%	49.6 7%	66.1 8%	0.702 2s
		Tanpa Cat (Semen)	98.14 %	47.82 %	45.2 4%	30.5 6%	45.8 6%	0.762 2s
2	Sedang	Dinding Bercat	97.89 %	91.99 %	89.7 0%	83.1 2%	90.7 3%	0.745 7s

		Tanpa Cat (Semen)	91.87 %	81.86 %	71.0 0%	61.3 0%	75.7 8%	0.702 2s
3	Berat	Dinding Bercat	92.37 %	96.07 %	83.1 5%	80.3 1%	88.8 8%	0.748 5s
		Tanpa Cat (Semen)	86.20 %	99.18 %	44.9 4%	44.8 0%	61.7 0%	0.702 8s

Berdasarkan Tabel 4.11, model SegNet menunjukkan performa yang lebih baik pada permukaan dinding bercat dibandingkan dinding tanpa cat di seluruh kategori tingkat keretakan, khususnya pada metrik Recall, F1-Score, dan IoU. Pada permukaan dinding bercat yang lebih halus dan homogen, pola retakan tampil lebih jelas sehingga memudahkan SegNet dalam memetakan area kerusakan secara konsisten. Sebaliknya, pada permukaan dinding tanpa cat atau semen ekspos yang memiliki tekstur kasar dan tidak seragam, performa model mengalami penurunan yang cukup signifikan, terutama pada nilai Recall dan IoU, yang mengindikasikan bahwa model cenderung kesulitan membedakan retakan struktural dari tekstur permukaan yang menyerupai pola retak. Kondisi ini menyebabkan sebagian area retakan tidak terdeteksi atau tersegmentasi secara tidak optimal. Dalam konteks aplikasi dunia nyata, hasil ini menunjukkan bahwa SegNet lebih andal digunakan pada bangunan dengan permukaan dinding yang relatif halus, sementara pada dinding kasar diperlukan adaptasi tambahan seperti peningkatan variasi data latih pada permukaan semen ekspos, augmentasi tekstur yang lebih intensif, atau preprocessing untuk menekan noise visual agar performa deteksi tetap optimal di lapangan.

4.2.6. Hasil Pengujian Model SegNet dengan Variasi Pencahayaan

Tabel 4.12 Evaluasi Berdasarkan Variasi Pencahayaan

N o	Kategori Keretakan	Kondisi Permukaan	Accuracy	Precision	Recall	IoU (%)	F1-Score (%)	Waktu Inferensi (s)
1	Ringan	Alami (Terang)	98.40 %	58.82 %	50.95 %	37.80 %	54.13 %	0.99 07s
		Redup (Indoor)	98.05 %	46.97 %	97.20 %	46.35 %	63.29 %	0.95 77s
2	Sedang	Alami (Terang)	96.87 %	93.19 %	72.71 %	68.05 %	79.66 %	0.93 56s

		Redup (Indoor)	97.39 %	89.69 %	78.33 %	71.53 %	82.85 %	0.88 73s
3	Berat	Alami (Terang)	92.11 %	97.14 %	80.24 %	78.31 %	87.64 %	0.88 36s
		Redup (Indoor)	75.81 %	93.20 %	18.75 %	18.57 %	30.17 %	0.88 33s

Berdasarkan Tabel 4.12, kondisi pencahayaan terbukti memberikan pengaruh signifikan terhadap performa model SegNet, terutama pada metrik Recall dan IoU. Pada kategori retak ringan dan sedang, kondisi pencahayaan redup justru menunjukkan peningkatan nilai Recall dibandingkan kondisi terang, yang mengindikasikan bahwa SegNet cenderung lebih sensitif dalam mendeteksi area retakan pada intensitas cahaya rendah. Namun demikian, pada kategori retak berat, performa model mengalami penurunan yang sangat signifikan pada kondisi pencahayaan redup, ditandai dengan turunnya nilai Recall hingga 18,75% dan IoU sebesar 18,57%. Kondisi ini menunjukkan bahwa meskipun SegNet mampu mendeteksi retakan kecil secara agresif pada kondisi low-light, model mengalami kesulitan dalam mempertahankan konsistensi deteksi pada retakan besar akibat menurunnya kontras visual dan meningkatnya noise citra. Dalam konteks aplikasi dunia nyata, hasil ini mengindikasikan bahwa SegNet memiliki karakteristik sensitivitas tinggi terhadap keberadaan retakan pada kondisi minim cahaya, namun berisiko menghasilkan false negative pada keretakan berat. Oleh karena itu, untuk meningkatkan kinerja SegNet pada kondisi low-light, diperlukan adaptasi seperti augmentasi data pencahayaan redup yang lebih terkontrol, peningkatan kontras pada tahap preprocessing, serta penyesuaian strategi pelatihan agar sensitivitas model tetap seimbang tanpa mengorbankan stabilitas deteksi pada retakan berskala besar.

4.3. Implementasi GUI Model U-Net dan SegNet

Pada tahap ini, model *Deep Learning* U-Net dan SegNet yang telah dilatih sebelumnya diimplementasikan ke dalam sebuah aplikasi berbasis *Graphical User Interface* (GUI). Pembuatan GUI bertujuan untuk mempermudah pengguna dalam melakukan proses segmentasi dan klasifikasi keretakan dinding secara interaktif. Dengan adanya antarmuka ini, proses analisis citra dapat dilakukan cukup dengan mengunggah gambar melalui menu yang tersedia, sehingga pengujian data dapat dilakukan secara praktis tanpa harus melakukan konfigurasi kode program secara manual untuk setiap citra baru yang ingin di uji.

Dalam penelitian ini, pembuatan GUI menggunakan *framework* Tkinter, yaitu *library* bawaan Python yang umum digunakan untuk mengembangkan antarmuka aplikasi desktop. Pemilihan Tkinter didasarkan pada beberapa pertimbangan utama, antara lain kemudahan integrasi dengan model *deep learning* yang telah

dibangun, ukuran aplikasi yang ringan saat dijalankan (*running*), serta kemudahan dalam kustomisasi tampilan untuk kebutuhan visualisasi hasil segmentasi.

Proses implementasi GUI dilakukan dengan mengintegrasikan model U-Net dan SegNet ke dalam satu alur sistem. Aplikasi dirancang untuk memproses citra masukan, mendeteksi area piksel retakan, dan secara otomatis menghitung persentase kerusakan guna menentukan klasifikasi tingkat keparahan sesuai *threshold* penelitian, yaitu: Ringan (<5%), Sedang (5-20%), dan Berat (>20%). Selain menampilkan visualisasi hasil segmentasi bersanding dengan citra asli, antarmuka juga menyajikan informasi waktu inferensi (*inference time*). Data ini ditampilkan untuk memberikan gambaran kuantitatif mengenai kecepatan komputasi yang dibutuhkan oleh masing-masing model dalam memproses satu citra hingga menghasilkan prediksi.



Gambar 4.9 Tampilan GUI U-Net dan SegNet

Gambar 4.9 menunjukkan tampilan antarmuka grafis (Graphical User Interface/GUI) yang dirancang untuk memudahkan pengguna dalam memuat model, memasukkan citra uji, menampilkan hasil segmentasi, serta merekam riwayat pengujian secara otomatis.

Pada bagian Model Config, pengguna dapat memuat *best* model U-Net dan SegNet yang telah dilatih sebelumnya. Sistem menyediakan indikator status untuk memastikan bahwa model telah berhasil dimuat sebelum proses pengujian dilakukan. Selain itu, ditampilkan pula informasi *threshold* klasifikasi tingkat kerusakan retak yang digunakan sebagai dasar penentuan kategori ringan, sedang dan berat.

Bagian Image Input berfungsi untuk memasukkan citra uji ke dalam sistem. Pengguna dapat mengunggah citra, menjalankan proses segmentasi, serta memilih mode visualisasi hasil berupa *masks*, *overlay*, atau *binary*. Fitur ini memungkinkan pengguna untuk melihat hasil segmentasi dalam berbagai

representasi sesuai kebutuhan analisis. Secara lebih spesifik, ketiga mode visualisasi tersebut menyediakan perspektif berbeda dalam menganalisis hasil deteksi retakan:

- **Mask:** Memvisualisasikan area retakan menggunakan warna solid yang menutupi area objek tanpa menampilkan detail tekstur dari citra asli. Fokus utama dari mode ini adalah untuk mengisolasi bentuk, pola, dan geometri retakan sehingga pengguna dapat melihat struktur kerusakan secara jelas tanpa gangguan visual dari warna atau bayangan pada dinding.
- **Overlay:** Menyajikan hasil segmentasi dalam bentuk lapisan warna transparan yang diletakkan tepat di atas citra asli (*original image*). Visualisasi ini sangat krusial untuk proses validasi, karena memungkinkan pengguna untuk memastikan apakah posisi retakan yang diprediksi oleh model U-Net maupun SegNet sudah presisi dan sesuai dengan lokasi fisik retakan yang sebenarnya pada permukaan dinding.
- **Binary:** Menampilkan hasil prediksi dalam format citra hitam-putih murni, di mana piksel putih (nilai 1) merepresentasikan area yang terdeteksi sebagai retakan dan piksel hitam (nilai 0) merepresentasikan latar belakang atau dinding bersih. Mode ini merupakan representasi data mentah yang digunakan sistem untuk melakukan kalkulasi matematis terhadap persentase luas retakan secara objektif.

Hasil pengujian ditampilkan pada bagian Results, yang terdiri dari tiga panel utama, yaitu citra asli (*original image*), hasil segmentasi menggunakan model U-Net, dan hasil segmentasi menggunakan model SegNet. Pada setiap panel hasil, sistem menampilkan informasi kategori tingkat kerusakan, persentase luas area retak, serta waktu inferensi yang dibutuhkan oleh masing-masing model.

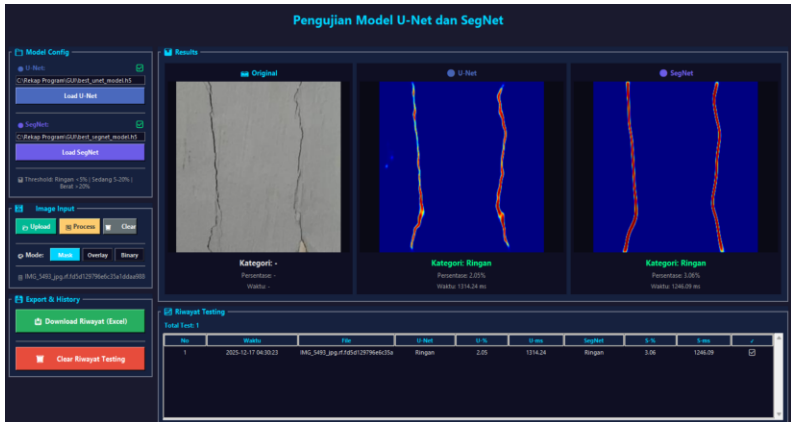
Selain itu, GUI ini dilengkapi dengan fitur Export & History dan Riwayat Testing yang berfungsi untuk menyimpan serta menampilkan rekam jejak pengujian. Riwayat tersebut mencakup waktu pengujian, nama berkas citra, hasil klasifikasi dari masing-masing model, persentase area retak, dan waktu proses. Data riwayat dapat diekspor ke dalam format *Excel* untuk keperluan dokumentasi dan analisis lebih lanjut.

Secara keseluruhan, antarmuka GUI ini mendukung proses pengujian model segmentasi retak secara terstruktur, interaktif, dan informatif, sehingga memudahkan pengguna dalam melakukan evaluasi performa model U-Net dan SegNet.

Selain berfungsi sebagai antarmuka visual, implementasi GUI ini juga memungkinkan evaluasi perbedaan performa antara model U-Net dan SegNet secara langsung, khususnya dari sisi ketelitian hasil segmentasi dan waktu inferensi. Informasi waktu inferensi yang ditampilkan menunjukkan bahwa SegNet cenderung memiliki proses komputasi yang lebih efisien, sedangkan U-Net

membutuhkan waktu inferensi sedikit lebih lama akibat kompleksitas arsitektur dan penggunaan skip connection. Perbedaan ini memberikan gambaran awal mengenai trade-off antara presisi segmentasi dan efisiensi komputasi dalam penerapan sistem deteksi keretakan secara praktis.

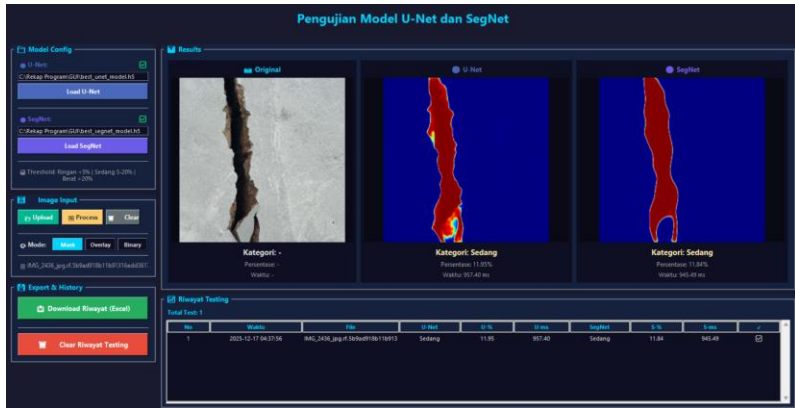
4.3.1. Hasil Pengujian GUI Model U-Net dan SegNet



Gambar 4.10 Hasil Pengujian Model U-Net dan SegNet pada sampel Data Uji Kelas Retak Ringan

Gambar 4.10 menunjukkan hasil pengujian model U-Net dan SegNet pada sampel data uji dengan kategori retak ringan. Pada citra asli terlihat retakan tipis dengan lebar yang relatif kecil. Hasil segmentasi dari kedua model mampu mendeteksi pola retakan dengan cukup baik, meskipun masih terdapat beberapa bagian retak yang tidak tersegmentasi secara sempurna. Berdasarkan hasil perhitungan luas area retak, baik model U-Net maupun SegNet mengklasifikasikan sampel ini ke dalam kategori ringan sesuai dengan ambang batas yang telah ditentukan.

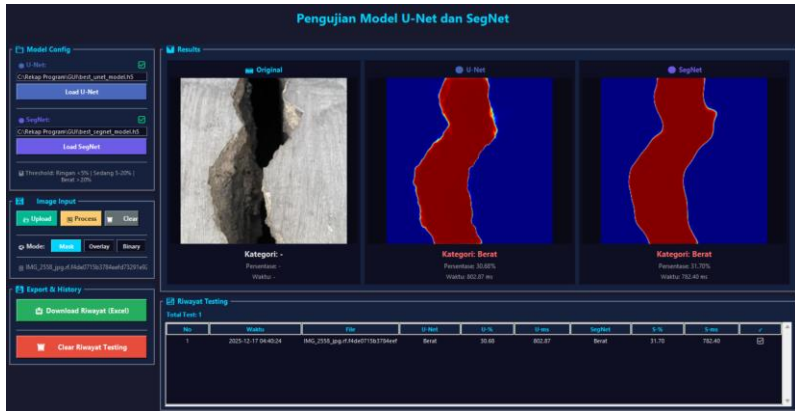
Pada kategori retak ringan, perbedaan karakteristik model U-Net dan SegNet mulai terlihat pada sensitivitas terhadap retakan berukuran kecil. U-Net cenderung menghasilkan segmentasi yang lebih mengikuti kontur retakan secara detail, yang berkaitan dengan kemampuannya mempertahankan informasi spasial resolusi tinggi melalui mekanisme skip connection. Sebaliknya, SegNet menunjukkan hasil segmentasi yang lebih konservatif, di mana sebagian retakan tipis tidak terdeteksi secara optimal. Hal ini menunjukkan bahwa arsitektur SegNet yang lebih sederhana memiliki keterbatasan dalam menangkap detail halus pada citra dengan tingkat kerusakan rendah.



Gambar 4.11 Hasil Pengujian Model U-Net dan SegNet pada sampel Data Uji Kelas Retak Sedang

Gambar 4.11 memperlihatkan hasil pengujian pada sampel dengan kategori retak sedang. Retakan pada citra asli tampak lebih lebar dan memanjang dibandingkan kelas ringan. Hasil segmentasi menunjukkan bahwa kedua model mampu mengikuti kontur retakan dengan lebih jelas, terutama pada bagian utama retak. Persentase area retak yang dihasilkan berada pada rentang 5–20%, sehingga sistem mengklasifikasikan sampel ini sebagai retak sedang. Perbedaan tipis pada hasil segmentasi menunjukkan adanya variasi sensitivitas model terhadap area tepi retakan.

Pada kategori retak sedang, kedua model menunjukkan performa yang relatif stabil dalam mendeteksi area keretakan utama. Namun demikian, U-Net masih memperlihatkan keunggulan dalam mengikuti batas retakan secara lebih presisi, terutama pada area tepi retakan yang kompleks. SegNet, di sisi lain, menghasilkan segmentasi yang lebih merata dan sederhana, mencerminkan pendekatan arsitektur yang lebih efisien namun kurang detail. Perbedaan ini menunjukkan bahwa pada tingkat kerusakan menengah, kedua model sama-sama efektif, tetapi memiliki karakteristik segmentasi yang berbeda.



Gambar 4.12 Hasil Pengujian Model U-Net dan SegNet pada sampel Data Uji Kelas Retak Berat

Gambar 4.12 menunjukkan hasil pengujian model pada sampel retak berat. Citra asli menampilkan retakan dengan lebar besar dan area kerusakan yang signifikan. Hasil segmentasi dari model U-Net dan SegNet mampu mengidentifikasi hampir seluruh area retak secara konsisten. Persentase area retak yang dihasilkan melebihi 20%, sehingga sistem secara otomatis mengklasifikasikan sampel ke dalam kategori retak berat. Hal ini menunjukkan bahwa kedua model memiliki performa yang baik dalam mendeteksi kerusakan dengan tingkat keparahan tinggi.

Pada pengujian retak berat, perbedaan performa antara model U-Net dan SegNet menjadi kurang signifikan karena karakteristik retakan yang dominan secara visual. Retakan dengan lebar besar dan area kerusakan yang luas memudahkan kedua model dalam melakukan segmentasi. Meskipun demikian, U-Net tetap menghasilkan batas segmentasi yang lebih presisi, sedangkan SegNet menunjukkan hasil yang lebih kasar namun stabil. Hal ini menunjukkan bahwa pada kondisi kerusakan berat, kompleksitas arsitektur model tidak terlalu mempengaruhi keberhasilan deteksi secara keseluruhan.



Gambar 4.13 Hasil Pengujian Model U-Net dan SegNet pada Citra Non Retak dan Non Relevan

Gambar 4.13 Menunjukkan hasil pengujian sampel citra dinding bersih atau tanpa retak (*background only*), ditemukan kasus di mana sistem tetap memberikan indikasi kerusakan, meskipun secara visual dinding tidak memiliki keretakan fisik, sistem mendeteksi sejumlah kecil piksel sebagai area retak. Hal ini menyebabkan algoritma menghitung persentase kerusakan di atas 0% dan secara otomatis menampilkan hasil klasifikasi sebagai kategori Retak Ringan.

4.4. Pembahasan

4.4.1. Analisis Performa Utama Model U-Net dan SegNet

Pada tahap ini, dilakukan analisis komparatif antara model U-Net dan SegNet untuk mengevaluasi efektivitas kedua arsitektur secara menyeluruh. Evaluasi didasarkan pada triangulasi data antara metrik segmentasi (*Mean IoU*), bukti visual hasil prediksi, laporan klasifikasi (*Classification Report*), dan pola distribusi kesalahan (*Confusion Matrix*).

Nilai performa segmentasi yang diperoleh, khususnya nilai IoU yang tinggi, memiliki implikasi langsung terhadap penerapan sistem deteksi keretakan pada kondisi lapangan. Intersection over Union (IoU) merepresentasikan tingkat kesesuaian antara area retakan hasil prediksi model dengan kondisi aktual (*ground truth*), sehingga nilai IoU yang tinggi menunjukkan bahwa model mampu memetakan lokasi dan luasan retakan secara presisi. Dalam konteks inspeksi bangunan, kemampuan ini sangat penting karena ketepatan pemetaan area kerusakan akan memengaruhi akurasi perhitungan tingkat keparahan retakan, yang selanjutnya menjadi dasar pengambilan keputusan teknis terkait tindakan perbaikan atau pemeliharaan struktur.

1. Confusion Matrix Klasifikasi Tingkat Keretakan

Temuan visual dan statistik tersebut dikuatkan oleh pola kesalahan pada *Confusion Matrix*, di mana rendahnya nilai *Recall* (0.73) pada kategori Sedang di model SegNet berkorelasi langsung dengan kesalahan prediksi yang mencatatkan 3 citra kategori Sedang salah diklasifikasikan menjadi Ringan dan 1 citra menjadi Berat. Kesalahan ini terjadi secara logis akibat hasil segmentasi visual SegNet yang terputus-putus, yang menyebabkan area retakan tidak terdeteksi secara utuh dan persentase piksel kerusakan yang dihitung sistem menyusut menjadi lebih kecil dari kondisi aslinya, sehingga data yang seharusnya masuk kategori Sedang (5-20%) jatuh ke rentang kategori Ringan (<5%). Sebaliknya, U-Net menunjukkan stabilitas klasifikasi yang jauh lebih baik dengan tingkat kesalahan yang minim dan terdistribusi merata—hanya 4 kesalahan dari total 45 data—tanpa adanya bias yang signifikan terhadap satu kelas tertentu.

2. Classification Report

Analisis mendalam terhadap *Classification Report* (Tabel 4.3 dan Tabel 4.8) mengungkapkan perbedaan signifikan pada performa per kelas akibat perbedaan kualitas segmentasi tersebut. Model U-Net menunjukkan performa yang sangat seimbang (*balanced*) antar kategori dengan nilai *Macro Average F1-Score* mencapai 91%, didukung oleh rincian *F1-Score* untuk kelas Ringan (0.93), Sedang (0.90), dan Berat (0.90) yang relatif setara, menandakan kemampuan U-Net mengenali ciri visual ketiga kategori keretakan dengan sama baiknya. Berbeda dengan itu, Model SegNet, meskipun memiliki *Macro Average F1-Score* 89%, memperlihatkan ketimpangan performa yang mencolok pada kategori Sedang, di mana nilai *Recall* anjlok ke angka 0.73 (73%)—jauh di bawah kategori Ringan (1.00) dan Berat (0.93)—yang secara kuat mengindikasikan bahwa SegNet sering kali gagal mengenali kembali data kategori Sedang secara akurat dibandingkan kategori lainnya.

3. Metrik Evaluasi Segmentasi Arsitektur

Secara garis besar, model U-Net menunjukkan superioritas dalam kualitas segmentasi citra, di mana data pengujian mencatatkan *Mean IoU* sebesar 0.6356 yang mengungguli SegNet dengan angka 0.6005, sebuah selisih yang terkonfirmasi secara jelas melalui perbandingan visual hasil prediksi. Pada hasil visualisasi tersebut, terlihat bahwa prediksi model U-Net mampu menghasilkan area retakan yang padat, utuh, dan berkelanjutan yang sangat presisi menyerupai *Ground Truth*, membuktikan bahwa fitur *skip connections* pada U-Net efektif dalam memulihkan informasi spasial yang hilang sehingga tepian retakan dapat terdeteksi dengan tajam. Sebaliknya, hasil prediksi

SegNet terlihat lebih kasar, terputus-putus (*fragmented*), dan mengandung *noise* atau bercak piksel yang tidak relevan di sekitar objek utama, di mana fenomena "bolong-bolong" ini terjadi akibat penggunaan indeks *max-pooling* saat *upsampling* yang gagal merekonstruksi detail halus pada objek tipis seperti retakan; ketidakmampuan SegNet mengisi piksel retakan secara penuh inilah yang secara langsung menyebabkan penurunan akurasi piksel dan perolehan nilai *IoU* yang lebih rendah dibandingkan U-Net.

Secara praktis, nilai *IoU* tertinggi yang dicapai oleh model U-Net, yakni sebesar 0.963 pada skenario pengujian tertentu, menunjukkan bahwa hampir seluruh area retakan yang terdeteksi oleh sistem memiliki tumpang tindih yang sangat tinggi dengan kondisi aktual di lapangan. Hal ini berarti bahwa sistem berbasis U-Net mampu meminimalkan kesalahan segmentasi, baik berupa area retakan yang terlewat (*false negative*) maupun area non-retakan yang salah terdeteksi (*false positive*). Dalam implementasi nyata, keunggulan ini memungkinkan proses inspeksi visual bangunan dilakukan secara lebih objektif dan konsisten dibandingkan inspeksi manual yang sangat bergantung pada subjektivitas pengamat.

Jika dibandingkan secara langsung dengan SegNet, keunggulan U-Net menjadi lebih konkret pada aspek akurasi segmentasi dan stabilitas deteksi. U-Net secara konsisten menghasilkan nilai *IoU* dan *Recall* yang lebih tinggi, terutama pada retakan dengan bentuk tipis dan tidak beraturan, karena arsitekturnya mampu mempertahankan informasi spasial resolusi tinggi melalui mekanisme *skip connection*. Sebaliknya, SegNet meskipun lebih efisien secara komputasi, cenderung menghasilkan segmentasi yang terfragmentasi pada objek tipis seperti retakan, yang berdampak pada menurunnya akurasi piksel dan ketepatan klasifikasi tingkat kerusakan.

4. Efisiensi Komputasi Model

Dari aspek kecepatan, SegNet terbukti lebih efisien dengan rata-rata waktu inferensi 216.88 ms, lebih cepat sekitar 20% dibandingkan U-Net yang membutuhkan 271.62 ms per citra. Efisiensi ini didapat karena SegNet tidak menyimpan *feature maps* lengkap pada memori melainkan hanya indeks lokasi pikselnya. Namun, menimbang tujuan utama sistem sebagai alat inspeksi keselamatan infrastruktur, keunggulan akurasi U-Net—terutama kemampuannya mendeteksi retakan secara utuh dan menjaga nilai *Recall* yang tinggi—dinilai lebih krusial dibandingkan keunggulan kecepatan SegNet. Kegagalan mendeteksi tingkat keparahan retak secara akurat berisiko menimbulkan kesalahan diagnosis penanganan perbaikan dinding. Dengan demikian, meskipun SegNet menawarkan keunggulan dari sisi kecepatan inferensi, hasil penelitian ini menunjukkan bahwa U-Net memberikan keseimbangan yang lebih optimal antara akurasi, presisi segmentasi, dan reliabilitas deteksi keretakan, sehingga lebih sesuai untuk aplikasi inspeksi

keselamatan bangunan yang mengutamakan ketepatan diagnosis dibandingkan efisiensi waktu semata.

4.4.2. Analisis Variasi Permukaan Dinding Model U-Net dan SegNet

Berdasarkan hasil pengujian pada Tabel 4.6 dan Tabel 4.11, variasi permukaan dinding terbukti memberikan pengaruh yang signifikan terhadap kinerja model U-Net dan SegNet pada seluruh kategori tingkat keretakan. Kedua model secara konsisten menunjukkan performa yang lebih baik pada dinding bercat dibandingkan dinding tanpa cat (semen), yang berkaitan dengan karakteristik visual permukaan dinding bercat yang lebih halus dan homogen sehingga retakan memiliki kontras yang lebih jelas terhadap latar belakang. Kondisi ini tercermin dari nilai IoU dan F1-Score yang tinggi pada dinding bercat, khususnya pada model U-Net yang mampu mempertahankan kualitas segmentasi secara konsisten di seluruh kategori keretakan. Jika dibandingkan secara langsung, U-Net menunjukkan penurunan performa yang lebih ekstrem pada permukaan tanpa cat dibandingkan SegNet, terutama pada kelas retak ringan dan berat, di mana nilai IoU U-Net turun hingga di bawah 10% pada retak ringan, sedangkan SegNet masih mempertahankan IoU di kisaran 30%. Hal ini menunjukkan bahwa meskipun U-Net unggul pada kondisi visual yang jelas, model ini lebih sensitif terhadap degradasi kualitas citra akibat tekstur kasar dan noise permukaan. Sebaliknya, SegNet menunjukkan degradasi performa yang relatif lebih moderat pada permukaan kasar, meskipun kualitas segmentasi yang dihasilkan secara umum tetap lebih rendah dibandingkan U-Net pada dinding bercat.

Perbedaan karakteristik ini menunjukkan adanya trade-off yang menarik antara kedua model dalam menghadapi variasi permukaan dinding. U-Net unggul dalam menghasilkan segmentasi yang presisi dan utuh pada permukaan yang mendukung (dinding bercat), tetapi cenderung kehilangan sensitivitas secara drastis ketika kontras visual menurun. SegNet, meskipun menghasilkan segmentasi yang lebih kasar, menunjukkan ketahanan relatif yang lebih baik terhadap tekstur permukaan kasar karena pendekatan arsitekturnya yang lebih sederhana dan tidak terlalu bergantung pada detail spasial resolusi tinggi. Dalam konteks implementasi lapangan, temuan ini mengindikasikan bahwa pemilihan model perlu disesuaikan dengan karakteristik bangunan yang akan dianalisis. Untuk bangunan dengan dinding bercat atau permukaan halus, U-Net lebih direkomendasikan karena memberikan akurasi dan presisi segmentasi yang tinggi. Namun, pada bangunan dengan dinding tanpa cat atau semen ekspos, diperlukan strategi adaptasi tambahan—seperti augmentasi data berbasis tekstur kasar, peningkatan preprocessing kontras, atau bahkan pendekatan hybrid—agar sistem dapat meminimalkan penurunan performa dan tetap memberikan hasil deteksi yang andal di lapangan.

4.4.3. Analisis Variasi Pencahayaan Model U-Net dan SegNet

Berdasarkan hasil pengujian pada Tabel 4.7 dan Tabel 4.12, variasi pencahayaan terbukti memberikan pengaruh yang signifikan terhadap performa model U-Net dan SegNet dalam melakukan segmentasi keretakan bangunan pada seluruh kategori tingkat keretakan. Performa yang menurun pada kondisi pencahayaan redup, khususnya pada kategori retak berat, menunjukkan bahwa intensitas cahaya yang rendah menyebabkan berkurangnya kontras visual antara area retakan dan latar belakang permukaan dinding. Pada kondisi low-light, detail tepi retakan menjadi kurang tajam dan cenderung menyatu dengan tekstur sekitar, sehingga fitur visual yang diekstraksi oleh model menjadi kurang representatif. Kondisi ini berdampak langsung pada penurunan nilai Recall dan IoU, yang mengindikasikan bahwa sebagian area retakan gagal terdeteksi secara utuh. Tantangan utama pada pencahayaan redup terletak pada hilangnya informasi gradien intensitas yang penting untuk membedakan batas kerusakan secara akurat, terutama pada retakan berukuran besar yang membutuhkan kejelasan tepi untuk dapat tersegmentasi dengan baik.

Jika dibandingkan secara langsung, U-Net dan SegNet menunjukkan respons yang berbeda terhadap variasi pencahayaan. U-Net cenderung mengalami penurunan performa yang lebih signifikan pada kondisi pencahayaan redup untuk kategori retak berat, yang tercermin dari turunnya nilai Recall dan IoU secara drastis, menunjukkan bahwa model ini sangat bergantung pada kejelasan fitur spasial dan kontras visual. Sebaliknya, SegNet pada beberapa kasus, khususnya retak ringan dan sedang, justru menunjukkan peningkatan nilai Recall pada pencahayaan redup, yang mengindikasikan bahwa SegNet bersifat lebih sensitif terhadap keberadaan pola retakan pada kondisi low-light, meskipun sering kali diiringi dengan penurunan presisi batas segmentasi. Perbedaan ini menegaskan bahwa U-Net unggul dalam ketelitian segmentasi ketika kondisi pencahayaan mendukung, sedangkan SegNet lebih toleran terhadap variasi pencahayaan namun menghasilkan segmentasi yang cenderung lebih kasar dan kurang presisi.

Untuk mengatasi keterbatasan performa model pada kondisi pencahayaan rendah dalam implementasi lapangan, diperlukan strategi adaptasi yang lebih sistematis, seperti penerapan augmentasi data berbasis variasi pencahayaan rendah (low-light augmentation), penggunaan teknik peningkatan kontras citra seperti histogram equalization atau adaptive contrast enhancement pada tahap preprocessing, serta penyesuaian strategi pelatihan agar model lebih robust terhadap fluktuasi intensitas cahaya. Dengan pendekatan tersebut, sistem deteksi keretakan diharapkan mampu mengurangi kesenjangan performa antara kondisi pencahayaan terang dan redup, sekaligus memaksimalkan keunggulan masing-masing model sesuai dengan karakteristik arsitekturnya.

4.4.4. Analisis Implementasi GUI Model U-Net dan SegNet

Berdasarkan hasil pengujian pada Gambar 4.10, Gambar 4.11, dan Gambar 4.12, dapat diketahui bahwa model SegNet memiliki kinerja komputasi yang lebih cepat dibandingkan U-Net pada implementasi sistem berbasis antarmuka grafis (GUI), meskipun kedua model menunjukkan hasil segmentasi dan klasifikasi tingkat keretakan yang relatif konsisten. Hal ini terlihat dari waktu inferensi SegNet yang secara umum lebih rendah pada seluruh kategori keretakan, di mana pada keretakan berat SegNet membutuhkan waktu 782,40 ms dibandingkan 802,87 ms pada U-Net, pada keretakan sedang SegNet mencatat waktu 945,49 ms sedangkan U-Net 957,40 ms, serta pada keretakan ringan SegNet membutuhkan 1246,09 ms dan U-Net 1314,24 ms. Perbedaan waktu inferensi ini menunjukkan bahwa arsitektur SegNet yang menggunakan proses upsampling berbasis indeks pooling lebih efisien secara komputasi dibandingkan U-Net yang memanfaatkan skip connection dengan penggabungan feature map, sehingga membutuhkan proses komputasi yang lebih besar.

Dalam konteks implementasi dunia nyata, perbedaan waktu inferensi ini memiliki implikasi praktis yang penting, terutama pada skenario inspeksi bangunan yang membutuhkan pemrosesan cepat atau mendekati real-time. Waktu inferensi SegNet yang lebih singkat memungkinkan sistem melakukan analisis keretakan secara lebih responsif, sehingga cocok digunakan pada inspeksi lapangan dengan jumlah citra yang besar atau pada sistem monitoring yang menuntut kecepatan pemrosesan tinggi. Sebaliknya, meskipun U-Net memiliki waktu inferensi yang sedikit lebih lama, hasil segmentasi yang dihasilkan cenderung lebih detail dan mengikuti kontur retakan dengan lebih presisi, sebagaimana terlihat pada hasil visual GUI, khususnya pada keretakan ringan dan retakan dengan resolusi tinggi. Hal ini menunjukkan bahwa U-Net lebih unggul dalam mendeteksi keretakan kecil atau retakan dengan batas yang kompleks, meskipun konsekuensinya adalah waktu komputasi yang lebih besar.

Analisis ini menegaskan adanya trade-off yang jelas antara kecepatan dan akurasi pada kedua model. SegNet lebih unggul dari sisi efisiensi komputasi karena penggunaan indeks pooling yang lebih hemat memori dan komputasi, sehingga sesuai untuk aplikasi yang memprioritaskan kecepatan dan respons sistem. Sebaliknya, U-Net menawarkan akurasi segmentasi yang lebih tinggi berkat mekanisme skip connection yang mampu mempertahankan informasi spasial resolusi tinggi, sehingga lebih tepat digunakan pada aplikasi yang menuntut ketelitian segmentasi, seperti analisis kerusakan detail atau penilaian tingkat keparahan retakan secara presisi. Dengan demikian, pemilihan model dalam implementasi GUI ini dapat disesuaikan dengan kebutuhan lapangan, apakah lebih mengutamakan kecepatan pemrosesan atau akurasi hasil segmentasi.

Namun, terlepas dari perbandingan performa komputasi di atas, evaluasi fungsional terhadap sistem GUI secara keseluruhan turut mengungkap batasan

signifikan terkait validitas input. Analisis mendalam terhadap hasil pengujian mengungkapkan kelemahan signifikan terkait ketangguhan sistem dalam menangani citra non-retak (*negative samples*) dan input yang tidak relevan. Pertama, ditemukan terjadinya kesalahan deteksi positif palsu (*false positive*) pada citra dinding bersih, sebagaimana ditunjukkan pada Gambar 4.13. Variasi tekstur permukaan, noda semen, atau gradasi bayangan halus sering kali diinterpretasikan secara keliru sebagai piksel retak (*crack pixels*), yang menyebabkan akumulasi persentase kerusakan di atas 0% dan memicu klasifikasi otomatis 'Retak Ringan'. Kelemahan ini diperparah oleh ketiadaan mekanisme pra-pemrosesan untuk memvalidasi relevansi citra (*image relevance check*) pada implementasi GUI. Ketika sistem menerima input citra acak di luar domain pelatihan (*out-of-distribution inputs*), seperti logo atau wajah manusia, model tetap memaksakan proses segmentasi dan pencocokan fitur (*feature matching*). Garis tegas atau perbedaan kontras pada objek non-dinding tersebut dianggap sebagai pola retakan, yang kembali menghasilkan vonis 'Retak Ringan'. Hal ini mengindikasikan bahwa model *Deep Learning* yang dikembangkan terpaku pada pengenalan pola retakan tanpa memiliki kemampuan untuk mengenali kelas 'Bukan Dinding' (background/negative class) atau menolak input yang tidak valid secara otomatis.

Bab 5. Kesimpulan dan Saran

5.1. Kesimpulan

Berdasarkan hasil pengujian dan pembahasan yang telah dilakukan pada Bab 4, dapat disimpulkan bahwa model U-Net dan SegNet sama-sama mampu digunakan untuk mendeteksi dan mengklasifikasikan tingkat keretakan dinding ke dalam kategori ringan, sedang, dan berat. Namun demikian, kedua model menunjukkan karakteristik performa yang berbeda baik dari sisi akurasi segmentasi maupun efisiensi komputasi.

Pembeda utama penelitian ini dibandingkan dengan penelitian terdahulu seperti penelitian oleh König et al., 2021 atau Kulkarni et al., 2023 terletak pada spesifikasi objek dan variasi kondisi lingkungan yang diuji. Jika penelitian sebelumnya mayoritas berfokus pada segmentasi retakan pada struktur beton masif atau jalan raya dengan tekstur seragam, penelitian ini melakukan analisis mendalam pada dinding bangunan yang memiliki variasi lapisan (bercat vs tanpa cat) serta pengaruh pencahayaan ruang. Temuan spesifik yang membedakan penelitian ini adalah identifikasi karakteristik *robustness* model; di mana ditemukan bahwa meskipun U-Net unggul dalam presisi piksel pada permukaan halus, SegNet justru menunjukkan stabilitas yang lebih baik terhadap *noise* tekstur pada permukaan dinding semen yang kasar, suatu fenomena yang jarang dibahas secara komparatif dalam studi literatur segmentasi retakan umum.

Secara keseluruhan, U-Net menunjukkan kualitas segmentasi yang lebih unggul dengan nilai IoU dan F1-Score yang lebih tinggi, serta hasil visual segmentasi yang lebih utuh dan presisi, terutama pada keretakan ringan dan keretakan dengan batas yang kompleks. Sebaliknya, SegNet menunjukkan keunggulan pada aspek efisiensi komputasi dengan waktu inferensi yang lebih cepat secara konsisten pada seluruh skenario pengujian.

Variasi kondisi permukaan dinding terbukti memberikan pengaruh signifikan terhadap performa kedua model. Pada permukaan dinding bercat, baik U-Net maupun SegNet menunjukkan performa yang lebih baik dibandingkan pada dinding tanpa cat (semen), yang disebabkan oleh tekstur permukaan yang lebih halus dan kontras visual retakan yang lebih jelas. U-Net unggul secara signifikan pada kondisi ini, namun menunjukkan penurunan performa yang lebih drastis pada permukaan kasar dibandingkan SegNet. Sementara itu, SegNet cenderung lebih toleran terhadap tekstur kasar, meskipun kualitas segmentasinya secara umum tetap berada di bawah U-Net.

Pada variasi pencahayaan, hasil pengujian menunjukkan bahwa kondisi pencahayaan sangat mempengaruhi kemampuan model dalam mendeteksi keretakan. Pencahayaan redup menyebabkan penurunan performa, khususnya pada keretakan berat, akibat berkurangnya kontras visual dan hilangnya detail tepi retakan. Namun, pada beberapa kasus keretakan ringan dan sedang, pencahayaan

redup justru meningkatkan sensitivitas model terhadap retakan halus. Perbandingan antara U-Net dan SegNet menunjukkan bahwa U-Net lebih bergantung pada kualitas fitur spasial yang jelas, sedangkan SegNet relatif lebih toleran terhadap variasi pencahayaan meskipun menghasilkan segmentasi yang lebih kasar.

Dari sisi implementasi sistem berbasis GUI, SegNet terbukti lebih efisien secara komputasi karena penggunaan indeks pooling yang lebih hemat memori, sehingga menghasilkan waktu inferensi yang lebih cepat dibandingkan U-Net. Sebaliknya, U-Net membutuhkan waktu inferensi yang sedikit lebih lama akibat penggunaan *skip connection*, namun mampu menghasilkan segmentasi yang lebih detail dan presisi. Dengan demikian, terdapat *trade-off* yang jelas antara kecepatan dan akurasi pada kedua model, yang perlu dipertimbangkan sesuai dengan kebutuhan aplikasi.

Di samping keberhasilan fungsi deteksi, ditemukan batasan pada sistem terkait validasi input. Sistem cenderung menghasilkan deteksi positif palsu (*false positive*) pada citra dinding bersih, di mana variasi tekstur mikro atau noda terdeteksi sebagai retak ringan. Selain itu, implementasi GUI saat ini belum memiliki mekanisme filter untuk menolak citra non-relevan (*out-of-distribution*) seperti citra acak atau objek selain dinding. Hal ini menyebabkan model tetap memproses input tersebut dan secara keliru mengklasifikasikannya ke dalam kategori ringan akibat ketiadaan kelas *negative sample* (bukan dinding) dalam pelatihan model.

Secara keseluruhan, hasil penelitian ini menegaskan bahwa pemilihan model deteksi keretakan tidak dapat ditentukan secara mutlak, melainkan harus disesuaikan dengan tujuan penggunaan. SegNet lebih sesuai untuk aplikasi yang memprioritaskan kecepatan pemrosesan dan kebutuhan *real-time*, sedangkan U-Net lebih tepat digunakan pada aplikasi yang menuntut ketelitian tinggi dalam pemetaan dan klasifikasi keretakan. Temuan ini memberikan kontribusi dalam pengembangan sistem deteksi kerusakan bangunan otomatis yang lebih efisien, adaptif dan aplikatif pada kondisi lapangan yang beragam.

5.2. Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat diberikan untuk pengembangan penelitian selanjutnya. Pertama, performa model pada permukaan dinding tanpa cat dan kondisi pencahayaan rendah masih dapat ditingkatkan dengan memperkaya dataset pelatihan, khususnya dengan menambahkan variasi tekstur permukaan kasar dan kondisi low-light agar model menjadi lebih robust terhadap kondisi lapangan yang tidak ideal. Kedua, penerapan teknik preprocessing tambahan, seperti peningkatan kontras citra atau normalisasi pencahayaan, dapat dipertimbangkan untuk meningkatkan kualitas fitur visual sebelum proses segmentasi.

Selain itu, penelitian selanjutnya dapat mengkaji pendekatan hybrid atau pengembangan arsitektur model yang mengombinasikan keunggulan U-Net dalam presisi segmentasi dan SegNet dalam efisiensi komputasi. Pengujian pada perangkat dengan spesifikasi terbatas atau sistem tertanam (embedded system) juga dapat dilakukan untuk mengevaluasi kesiapan model dalam implementasi nyata di lapangan. Terakhir, pengembangan antarmuka GUI dapat diperluas dengan fitur analisis lanjutan, seperti pelaporan otomatis tingkat kerusakan atau integrasi dengan sistem monitoring bangunan, sehingga sistem deteksi keretakan dapat memberikan manfaat yang lebih komprehensif dalam mendukung inspeksi dan pemeliharaan infrastruktur bangunan.

Daftar Pustaka

- [1] Y. Chen, Z. Zhu, Z. Lin, and Y. Zhou, "Building Surface Crack Detection Using Deep Learning Technology," *Buildings*, vol. 13, no. 7, 2023, doi: 10.3390/buildings13071814.
- [2] E. Science, "Case study : Estimating the occurrence of sea water intrusion using geoelectrical method in Pangandaran district Case study : Estimating the occurrence of sea water intrusion using geoelectrical method in Pangandaran district", doi: 10.1088/1755-1315/708/1/012002.
- [3] S. Sekolah and D. Statistik, "No Title".
- [4] K. PUPR, "PermenPUPR27-2018," no. 1, pp. 1–63, 2018.
- [5] G. Li, B. Ma, S. He, X. Ren, and Q. Liu, "Automatic tunnel crack detection based on u-net and a convolutional neural network with alternately updated clique," *Sensors (Switzerland)*, vol. 20, no. 3, 2020, doi: 10.3390/s20030717.
- [6] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," pp. 1–8.
- [7] S. Shomal Zadeh, S. Aalipour birgani, M. Khorshidi, and F. Kooban, "Concrete Surface Crack Detection with Convolutional-based Deep Learning Models," *SSRN Electron. J.*, vol. 10, no. 3, pp. 25–35, 2024, doi: 10.2139/ssrn.4661249.
- [8] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, 2017.
- [9] V. N. Vyrovoy, V. G. Sukhanov, and O. A. Korobko, "Analysis of cracks in structures and buildings," 2020, doi: 10.1088/1742-6596/1706/1/012116.
- [10] D. Loverdos and V. Sarhosis, "Automation in Construction Automatic image-based brick segmentation and crack detection of masonry walls using machine learning," *Autom. Constr.*, vol. 140, no. June, p. 104389, 2022, doi: 10.1016/j.autcon.2022.104389.
- [11] R. Indonesia, "Presiden republik indonesia," 2002.
- [12] Y. Cha and W. Choi, "Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks," no. March, 2017, doi: 10.1111/mice.12263.
- [13] P. N. Hadinata, L. Angeles, and L. Eddy, "Deep Convolutional Neural Network untuk Mendeteksi Retak pada Permukaan Beton yang Memiliki Void Deep Convolutional Neural Network untuk Mendeteksi Retak pada Permukaan Beton yang Memiliki Void," no. October 2021, 2023, doi:

- 10.26593/josc.v1i1.5151.
- [14] I. Bagus, L. Mahadya, M. Sudarma, I. N. S. Kumara, and A. Optimizer, "Resonance Imaging Dengan Menggunakan Metode U-NET," no. December 2020, 2021, doi: 10.24843/MITE.2020.v19i02.P05.
 - [15] F. T. Informasi and E. Mailoa, "Deteksi Kerusakan Jalan Menggunakan Convolutional Neural Network Berbasis SSD EfficientNet Artikel Ilmiah Program Studi S1 Teknik Informatika Fakultas Teknologi Informasi Universitas Kristen Satya Wacana Salatiga," no. 672016033, 2020.
 - [16] Y. Jimmy Aprilyanto, "View of Implementasi Arsitektur VGG-Unet Dalam Melakukan Segmentasi Keretakan pada Citra Bangunan".
 - [17] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," pp. 3431–3440, 2015.
 - [18] P. J. Curran, *Digital Image Processing*. 2016.
 - [19] M. Muchtar *et al.*, "Penerapan analisis berbasis fraktal dalam klasifikasi citra retakan pada permukaan jembatan beton," vol. 9, no. 1, pp. 21–29, 2024.
 - [20] P. N. Hadinata, D. Simanta, L. Eddy, and K. Nagai, "Multiclass Segmentation of Concrete Surface Damages Using U-Net and DeepLabV3+," *Appl. Sci.*, vol. 13, no. 4, 2023, doi: 10.3390/app13042398.
 - [21] W. Z. Taffese, R. Sharma, M. H. Afsharmovahed, G. Manogaran, and G. Chen, "Benchmarking YOLOv8 for Optimal Crack Detection in Civil Infrastructure," Jan. 2025, Accessed: Jul. 25, 2025. [Online]. Available: <https://arxiv.org/abs/2501.06922v2>
 - [22] S. Kulkarni, S. Singh, D. Balakrishnan, S. Sharma, S. Devunuri, and S. C. R. Korlapati, "CrackSeg9k: A Collection and Benchmark for Crack Segmentation Datasets and Frameworks," *Lect. Notes Comput. Sci.*, vol. 13807 LNCS, pp. 179–195, 2023, doi: 10.1007/978-3-031-25082-8_12.
 - [23] J. König, M. D. Jenkins, M. Mannion, P. Barrie, and G. Morison, "Optimized deep encoder-decoder methods for crack segmentation," *Digit. Signal Process. A Rev. J.*, vol. 108, pp. 1–27, 2021, doi: 10.1016/j.dsp.2020.102907.

Biodata



Nama : Muhammad Akmarizan
TTL : Kundur Karimun, 06-08-2004
Agama : Islam
Alamat : Jalan Parit Gantung, RT. 001 / RW. 001,
Kel. Sei Sebesi, Kec. Kundur.
Email : rizan88switch@gmail.com
Riwayat Pendidikan : SMA : SMA NEGERI 1 KUNDUR
SMP : MTs NEGERI KARIMUN



Nama : Laksamana Muda Jaya
TTL : Perbaungan, 23-03-2003
Agama : Islam
Alamat : Jalan Tiban Koperasi, Perum Tiban
Palem, RT. 005 / RW. 008 Blok C4 No.9,
Tiban Baru, Kec. Sekupang, Kota Batam.
Email : mudajaya065@gmail.com
Riwayat Pendidikan : SMK : SMK NEGERI 3 SIBOLGA
SMP : MTs NEGERI SIBOLGA

Lampiran

https://drive.google.com/drive/folders/159OIS39nkcXL21ZEUC3H8YPaL551dFOH?usp=drive_link