

**IMPLEMENTASI DENSENET-121 UNTUK EKSTRAKSI FITUR TINGKAT
KERUSAKAN BANGUNAN PASCA BENCANA ALAM**

SKRIPSI

Oleh :

IQLIMA RAHMAFITRI KARINDAH

NIM. 210605110146



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

**IMPLEMENTASI DENSENET-121 UNTUK EKSTRAKSI FITUR
TINGKAT KERUSAKAN BANGUNAN
PASCA BENCANA ALAM**

SKRIPSI

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

Oleh :
IQLIMA RAHMAFITRI KARINDAH
NIM. 210605110146

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

HALAMAN PERSETUJUAN

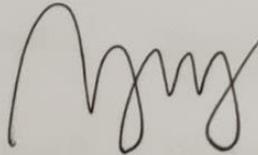
**IMPLEMENTASI DENSENET-121 UNTUK EKSTRAKSI
FITUR TINGKAT KERUSAKAN BANGUNAN
PASCA BENCANA ALAM**

SKRIPSI

Oleh :
IQLIMA RAHMAFITRI KARINDAH
NIM. 21060510146

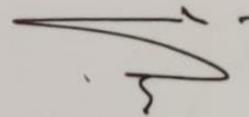
Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 05 Juni 2025

Pembimbing I,



Dr. Agung Teguh Wibowo Almais, M.T
NIP. 19860301 202321 1 016

Pembimbing II,

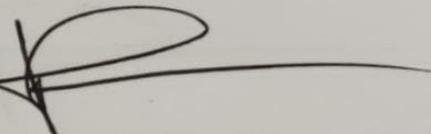


Syahiduz Zaman, M. Kom
NIP. 19700502 200501 1 005

Mengetahui,

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




Dr. M. Fachrul Kurniawan, M.MT., IPU
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

IMPLEMENTASI DENSENET-121 UNTUK EKSTRAKSI FITUR TINGKAT KERUSAKAN BANGUNAN PASCA BENCANA ALAM

SKRIPSI

Oleh :

IQLIMA RAHMAFITRI KARINDAH
NIM. 210605110146

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 23 Juni 2025

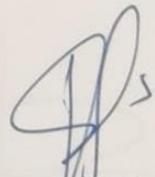
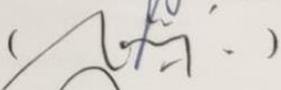
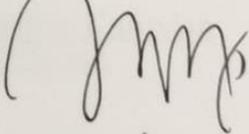
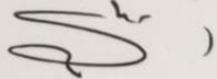
Susunan Dewan Penguji

Ketua Penguji : Dr. Totok Chamidy, M. Kom
NIP. 19691222 200604 1 001

Anggota Penguji I : Ashri Shabrina Afrah, M.T
NIP. 19900430 202012 2 003

Anggota Penguji II : Dr. Agung Teguh Wibowo Almais, M.T
NIP. 19860301 202321 1 016

Anggota Penguji III : Syahiduz Zaman, M.Kom
NIP. 19700502 200511 005

()
()
()
()

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



Dr. Ir. Fachrul Kurniawan, M.MT., IPU
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Iqlima Rahmafutri Karindah
NIM : 210605110146
Fakultas / Program Studi : Sains dan Teknologi/Teknik Informatika
Judul Skripsi : Implementasi DenseNet-121 untuk Ekstraksi Fitur
Tingkat Kerusakan Bangunan Pasca Bencana Alam

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

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

Malang, 05 Juni 2025
Yang membuat pernyataan,



Iqlima Rahmafutri Karindah
NIM.210605110146

MOTTO

So, surely with hardship comes ease.

[Al-Qur'an, 94:5]

HALAMAN PERSEMBAHAN

Karya ini penulis persembahkan untuk Mama, yang mimpinya telah menjadi langkah-langkah penulis hingga skripsi ini ditulis. Sejak kecil, Mama sangat ingin menempuh pendidikan di Universitas Islam Negeri Maulana Malik Ibrahim yang merupakan tempat penulis menempuh jenjang pendidikan S1 Teknik Informatika, namun takdir membawa Mama menuju jalan yang berbeda. Penulis mempersembahkan skripsi ini sebagai bukti bahwa mimpi itu tidak pernah benar-benar padam—ia menemukan caranya sendiri untuk terwujud. Penulis ingin mengucapkan terima kasih kepada Mama untuk segala doa, ketulusan, dan cinta tanpa syarat yang tak lekang oleh waktu. Keberhasilan penulis adalah bayangan dari pengorbanan Mama dan penulis bangga bisa berjalan di jejak mimpi yang Mama titipkan.

Untuk Ayah, yang selama ini menjadi definisi nyata dari kata kuat dan kokoh. Dari Ayah, penulis belajar arti ketegaran dan keberanian untuk terus melangkah, apapun rintangan yang akan menghadang jalan. Skripsi ini penulis persembahkan sebagai ungkapan terima kasih atas semua perjuangan dan kasih sayang yang mungkin tak selalu terucap, tetapi bisa dirasakan. Berkat Ayah—yang tidak pernah sekalipun mengeluh akan kerasnya kehidupan, selalu melihat kehidupan dengan bijak, dan menghadapi segala tantangan dengan sikap positif—penulis bisa menjadi sosok yang mandiri serta tumbuh dengan kuat dan tegar.

KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang telah melimpahkan segala nikmat dan kasih sayang-Nya sehingga penulis dapat menyelesaikan skripsi dengan lancar dan tepat waktu. Semoga shalawat serta salam selalu tercurahkan kepada Nabi Muhammad SAW. Dalam proses penyusunan skripsi ini penulis menyadari bahwa penulis tidak lepas dari dukungan dan bantuan dari berbagai pihak. Penulis ingin mengucapkan rasa terima kasih yang tak terhingga dengan penuh rasa syukur dan rasa hormat yang dalam kepada:

1. Prof. Dr. Zainuddin, M.A., selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Hj. Sri Harini, M.Si., selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Fachrul Kurniawan, M.MT., IPU, selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Dr. Agung Teguh Wibowo Almais, M.T., selaku dosen pembimbing I, yang tidak hanya membimbing dalam penyusunan skripsi ini, tetapi juga menjadi sosok panutan yang berperan besar dalam perjalanan akademik penulis. Berkat dukungan, motivasi, dan bimbingan yang diberikan beliau dapat meyakinkan penulis akan kemampuannya.
5. Syahiduz Zaman, M.Kom selaku dosen pembimbing II yang telah memberikan banyak bimbingan dan bantuan dalam proses penyusunan skripsi ini dari awal hingga akhir.

6. Dr. Totok Chamidy, M.Kom dan Ashri Shabrina Afrah, M.T selaku dosen penguji yang telah menguji serta memberikan banyak kritik serta saran yang membangun dalam proses untuk menyelesaikan skripsi ini.
7. Segenap Dosen, Admin, Laboran, dan Jajaran Staff Program Studi Teknik Informatika yang telah banyak memberikan bantuan selama masa studi serta dalam penyusunan skripsi ini.
8. Keluarga penulis yaitu Ibu Zuanita Astuti, Ayah Diwang Hadi Parmono, Kakak Nadhia Kirana Dias, dan Kakak Ardhan Maulana Zuhdi yang telah memberikan kasih sayang, dukungan, dan doa tiada henti sehingga penulis dapat menyelesaikan skripsi ini dengan baik.
9. Segenap *crew* Unit Kegiatan Mahasiswa (UKM) Simfoni FM yang telah menjadi ruang positif untuk bertumbuh dan selalu memberikan dukungan yang sangat berarti.
10. Semua pihak yang tidak dapat disebutkan satu per satu, yang telah memberikan berbagai bentuk bantuan selama masa studi dan dalam penyusunan skripsi ini.
11. Untuk diri sendiri yang tidak pernah menyerah, selalu bekerja keras, serta menggunakan masa studi semaksimal mungkin fokus untuk bertumbuh dan mengembangkan diri.

Malang, 05 Juni 2025

Penulis

DAFTAR ISI

HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
ABSTRAK	xiv
ABSTRACT	xv
البحث مستخلص	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	6
1.3 Batasan Masalah.....	6
1.4 Tujuan Penelitian	6
1.5 Manfaat Penelitian	7
BAB II STUDI PUSTAKA	8
2.1 Penelitian Terkait	8
2.2 Bencana Alam	11
2.3 <i>Convolution Neural Network</i> (CNN)	14
2.3.1 <i>Input Image</i>	15
2.3.2 <i>Convolution Layer</i>	16
2.3.3 <i>Pooling Layer</i>	17
2.3.4 <i>Activation Function Layer</i>	17
2.3.5 <i>Fully Connected Layer</i>	18
2.3.6 <i>Output Layer</i>	18
2.4 DenseNet-121	19
4.1 <i>Cosine Similarity</i>	25
4.2 <i>Principal Component Analysis</i> (PCA)	25
BAB III METODE PENELITIAN	27
3.1 Desain Sistem.....	27
3.2 <i>Data Preparation</i>	30
3.2.1 <i>Image Resize</i>	31
3.2.2 Konversi <i>Image</i> ke <i>Array</i>	31
3.2.3 Penyesuaian <i>Channel RGB</i>	32
3.2.4 Normalisasi	32
3.2.5 Penambahan Dimensi <i>Batch</i>	33
3.2.6 Augmentasi	33
3.3 Ekstraksi Fitur oleh DenseNet-121	34
3.4 Skenario Pengujian.....	39

BAB IV HASIL DAN PEMBAHASAN	42
4.1 Uji Coba Sistem	42
4.2 Hasil Uji Coba.....	43
4.2.1 Persiapan Dataset	43
4.2.2 Praproses Data	43
4.2.3 Ekstraksi Fitur oleh DenseNet-121	49
4.2.4 Evaluasi Vektor Fitur.....	55
3.5 Pembahasan.....	60
3.6 Integrasi Islam.....	64
BAB V KESIMPULAN DAN SARAN	66
5.1 Kesimpulan	66
5.2 Saran	66
DAFTAR PUSTAKA	
LAMPIRAN-LAMPIRAN	

DAFTAR GAMBAR

Gambar 2. 1 Komponen CNN (Taye, 2023).....	15
Gambar 2. 2 Operasi konvolusi (Wani et al., 2020).....	16
Gambar 2. 3 Fungsi aktivasi (Taye, 2023).....	17
Gambar 2. 4 (Kiri) Arsitektur DenseNet-121. (Kanan) Dense_block, conv_block dan transition_layer (Ji et al., 2019).....	20
Gambar 2. 5 <i>Feature extraction pipeline</i> dari model <i>pre-trained</i> DenseNet-121 (Ahishali et al., 2021)	24
Gambar 3. 1 Desain Sistem.....	27
Gambar 4. 1 Citra asli dan citra hasil image resize.....	44
Gambar 4. 2 Augmentasi data.....	47
Gambar 4. 3 <i>Feature Vector Heatmap</i>	52
Gambar 4. 4 <i>Feature Value Distribution</i>	53
Gambar 4. 5 Sampel vektor fitur.....	54
Gambar 4. 6 <i>2D PCA of extracted features</i>	58
Gambar 4. 7 <i>3D PCA of extracted features</i>	59

DAFTAR TABEL

Tabel 2. 1 Penelitian Terkait	9
Tabel 4. 1 Lapisan ekstraksi fitur oleh DenseNet-121	51
Tabel 4. 2 Statistik fitur.....	53
Tabel 4. 3 Hasil perhitungan <i>cosine similarity</i>	57

ABSTRAK

Karindah, Iqlima Rahmafutri. 2025. **Implementasi Densenet-121 untuk Ekstraksi Fitur Tingkat Kerusakan Bangunan Pasca Bencana Alam**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Agung Teguh Wibowo Almais, M.T (II) Syahiduz Zaman, M.Kom

Kata kunci: Bencana Alam, Ekstraksi Fitur, DenseNet-121, *Cosine Similarity*, Kerusakan Bangunan, *Principal Component Analysis (PCA)*

Bencana alam dapat menyebabkan berbagai kerusakan yang signifikan pada infrastruktur, termasuk bangunan. Identifikasi dan analisis terhadap kerusakan yang diakibatkan bencana alam sangat penting untuk pemulihan pasca bencana alam. Penelitian ini bertujuan untuk mengimplementasikan DenseNet-121 untuk melakukan ekstraksi fitur pada citra kerusakan bangunan pasca bencana alam dan untuk mengukur hasil ekstraksi oleh DenseNet-12. Metodologi yang digunakan adalah CNN dengan arsitektur DenseNet-121. Dataset yang digunakan dalam penelitian merupakan data *open source* yang terdiri dari berbagai citra kerusakan bangunan. Proses ekstraksi fitur dilakukan dengan DenseNet-121 yang telah dimodifikasi dengan menghapus lapisan yang tidak diperlukan. Hasil dari penelitian ini menunjukkan bahwa DenseNet-121 mampu mengekstrak fitur dengan baik yang dibuktikan melalui analisis data statistik, perhitungan *cosine similarity*, dan visualisasi menggunakan PCA. Hasil positif pada penelitian ini dapat mendukung penilaian otomatis terhadap kerusakan bangunan pasca bencana alam dengan lebih cepat dalam upaya pemulihan pasca bencana alam.

ABSTRACT

Karindah, Iqlima Rahmafutri. 2025. **Implementasi Densenet-121 untuk Ekstraksi Fitur Tingkat Kerusakan Bangunan Pasca Bencana Alam**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Promotor: (I) Dr. Agung Teguh Wibowo Almais, M.T (II) Syahiduz Zaman, M.Kom

Natural disasters can lead to a significant range of damage to infrastructure, including buildings. Identification and analysis of damage caused by natural disasters is essential for post-disaster recovery. This research aims to implement DenseNet-121 to perform feature extraction on post-natural disaster building damage images and to measure the extraction results by DenseNet-12. The applied methodology is CNN with DenseNet-121 architecture. The dataset utilized in the research is an opensource data which consists of numerous images of building damage. The feature extraction process is carried out with DenseNet-121 which has been modified by removing unnecessary layers. The results of this study show that DenseNet-121 is able to extract features well as proven through statistical data analysis, cosine similarity calculation, and visualization through PCA. These findings can support the automatic assessment of building damage after natural disasters more quickly in post-disaster recovery programs.

Key words: Natural Disasters, Feature Extraction, DenseNet-121, Cosine Similarity, Building Damage, Principal Component Analysis (PCA)

البحث مستخلص

.لاستخراج ميزات مستوى تلف المباني بعد الكوارث الطبيعية DenseNet-121 كارينداه، إقليم رحمة فيتري. 2025. تطبيق
:أطروحة. قسم هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية، مالانج. المشرفون
شاهيدوز زمان، م. كوم (II) د. أغونغ تغوه ويوه ألاميس، م.ت (I)

تشابه جيب التمام، تلف المباني، تحليل المكونات، DenseNet-121، الكلمات المفتاحية: الكوارث الطبيعية، استخراج الميزات
(PCA) الرئيسية

يمكن أن تسبب الكوارث الطبيعية مجموعة واسعة من الأضرار الكبيرة للبنية التحتية، بما في ذلك المباني. ويُعدّ تحديد
DenseNet-121 وتحليل الأضرار الناجمة عن الكوارث الطبيعية أمرًا ضروريًا للتعافي بعد الكوارث. يهدف هذا البحث إلى تطبيق
DenseNet-121 لإجراء استخراج السمات على صور الأضرار التي لحقت بالمباني بعد الكوارث الطبيعية وقياس نتائج الاستخراج بواسطة
مجموعة البيانات المستخدمة في البحث هي بيانات DenseNet-121 مع بنية CNN المنهجية المستخدمة هي شبكة DenseNet-
121 التي تم DenseNet-121 مفتوحة المصدر تتكون من صور مختلفة لأضرار المباني. يتم تنفيذ عملية استخراج السمات باستخدام
قادر على استخراج الميزات بشكل DenseNet-121 تعديلها عن طريق إزالة الطبقات غير الضرورية. تُظهر نتائج هذا البحث أن
يمكن أن تدعم PCA جيد كما يتضح من خلال التحليل الإحصائي للبيانات وحساب تشابه جيب التمام والتصوير باستخدام
النتائج الإيجابية في هذا البحث التقييم التلقائي للأضرار التي لحقت بالمباني بعد الكوارث الطبيعية بسرعة أكبر في جهود التعافي بعد
الكوارث الطبيعية.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Bencana alam adalah bencana yang diakibatkan oleh peristiwa atau serangkaian peristiwa yang disebabkan oleh alam antara lain berupa gempa bumi, tsunami, gunung meletus, banjir, kekeringan, angin topan, dan tanah longsor (Undang-Undang RI Nomor 24 Tahun 2007 tentang Penanggulangan Bencana, 2007) . Bencana alam ini pada dasarnya terjadi karena terjadi perubahan di alam. Perubahan tersebut dapat terjadi secara perlahan maupun secara ekstrem (Ammelia et al., 2022). Secara geologis, Indonesia terletak pada pertemuan antara tiga lempeng tektonik, yaitu Lempeng Eurasia, lempeng Pasifik, dan lempeng Indo-Australia. Sedangkan secara geografis, Indonesia terletak di Kawasan Cincin Api Pasifik (*Ring of Fire*) yang menyebabkan Indonesia kaya akan sumber daya alam karena letusan gunung vulkanik mengakibatkan tanah menjadi subur yang cocok untuk pertanian dan cadangan mineral yang melimpah.

Di sisi lain, letak geologis dan letak geografis Indonesia juga menjadikan Indonesia sebagai negara yang rawan terjadi bencana alam. Tingginya aktivitas dari gunung vulkanik yang membentang dari ujung barat sampai ujung timur Indonesia berpotensi besar terjadi bencana alam seperti badai, angin topan, angin siklon tropis, banjir, dan gunung meletus. Kemungkinan terjadinya pergeseran lempeng tektonik juga meningkatkan risiko terjadinya bencana alam di Indonesia seperti tanah longsor, gempa bumi, dan tsunami. Data statistik yang bersumber dari Badan

Nasional Penanggulangan Bencana (BNPB) telah menunjukkan jika angka terjadinya bencana alam di Indonesia cukup fluktuatif dari tahun ke tahun.

Berbagai jenis bencana alam yang kemungkinan besar dapat terjadi di Indonesia ini dapat menyebabkan kerugian yang bervariasi bagi kehidupan masyarakat. Bencana alam yang terjadi di Indonesia telah memakan banyak korban serta berbagai kerusakan pada bangunan. Badan Nasional Penanggulangan Bencana (BNPB) mencatat bahwa pada tahun 2024 terdapat 1.071.642 rumah yang terdampak bencana alam dengan rincian klasifikasi sebanyak 7.297 rumah rusak berat, 8.340 rumah rusak sedang, 29.734 rumah rusak ringan, dan 1.026.271 rumah terendam. Sedangkan untuk fasilitas setidaknya terdapat 463 fasilitas pendidikan, 44 fasilitas kesehatan, dan 347 fasilitas peribadatan yang mengalami kerusakan akibat terjadinya bencana alam pada tahun 2024.

Sebagai umat muslim, kerusakan massif akibat dari bencana alam tidak hanya diartikan sebagai kerusakan saja. Tetapi terdapat nilai penting yang dapat menjadi peringatan bahwa terdapat ketidakseimbangan karena perbuatan manusia yang mempertuhankan hawa nafsu dan jauh dari tuntunan fitrah sehingga Allah SWT menghendaki agar mereka merasakan akibat dari perbuatan buruknya dan agar mereka kembali ke jalan yang benar (NU Online, 2023b). Hal tersebut sesuai dengan firman Allah SWT pada QS. Ar-Rum:41 berikut ini:

ظَهَرَ الْفَسَادُ فِي الْبَرِّ وَالْبَحْرِ بِمَا كَسَبَتْ أَيْدِي النَّاسِ لِيُذِيقَهُمْ بَعْضَ الَّذِي عَمِلُوا لَعَلَّهُمْ يَرْجِعُونَ

“Telah tampak kerusakan di darat dan di laut disebabkan perbuatan tangan manusia. (Melalui hal itu) Allah membuat mereka merasakan sebagian dari (akibat) perbuatan mereka agar mereka kembali (ke jalan yang benar).” (Q.S Ar-Rum: 41)

Menurut Peraturan Menteri Pekerjaan Umum No. 45/PRT/M/2007 yang terdapat dalam lampiran PermenPUPR No. 29/PRT/M/2018 tentang Standar Teknis Standar Pelayanan Minimal Pekerjaan Umum dan Perumahan Rakyat, teknis untuk melaksanakan penilaian terhadap kerusakan bangunan secara manual telah ditetapkan standarnya. Kriteria untuk menentukan tingkat kerusakan rumah akibat bencana juga telah ditetapkan pada Peraturan Menteri Pekerjaan Umum No. 45/PRT/M/2007 yang terdapat dalam lampiran PermenPUPR No. 29/PRT/M/2018 tentang Standar Teknis Standar Pelayanan Minimal Pekerjaan Umum dan Perumahan Rakyat. Namun penilaian kerusakan bangunan pasca bencana alam secara manual akan memakan waktu lama ketika operasi bantuan pasca bencana alam dan proses pemulihan seharusnya dilakukan sesegera mungkin.

Di samping itu, terdapat risiko terhadap penilaian secara manual menghasilkan penilaian yang bersifat subjektif. Hal tersebut dapat terjadi karena perbedaan interpretasi dan pengalaman yang membuatnya rentan terhadap variasi akurasi di antara para ahli yang berbeda. Proses penilaian kerusakan bangunan pasca bencana alam secara manual juga berisiko besar bagi para ahli yang harus pergi ke lapangan terutama di daerah yang masih belum stabil pasca bencana alam. Dibutuhkan juga biaya yang relatif tinggi karena membutuhkan peralatan, transportasi, dan sumber daya manusia (SDM) yang memadai untuk menuju situs yang terdampak bencana alam.

Kelemahan-kelemahan yang terdapat pada metode penilaian kerusakan bangunan secara manual ada karena kurangnya pemanfaatan teknologi yang tersedia di masa ini. Diperlukan metode yang dapat melakukan penilaian terhadap

kerusakan bangunan pasca bencana alam secara otomatis dengan waktu yang lebih singkat, menghasilkan hasil penilaian yang akurat dan objektif, tidak memerlukan banyak sumber daya manusia, dan lebih aman karena tidak memerlukan banyak survei di wilayah yang terdampak bencana alam. Tentunya dengan memanfaatkan kemajuan teknologi dan meninggalkan metode konvensional yang kurang efisien.

Ayat Al-Qur'an yang mendasari penelitian ini adalah Surah Al-Baqarah ayat 195 yang berbunyi:

وَأَنْفِقُوا فِي سَبِيلِ اللَّهِ وَلَا تُلْقُوا بِأَيْدِيكُمْ إِلَى التَّهْلُكَةِ وَأَحْسِنُوا إِنَّ اللَّهَ يُحِبُّ الْمُحْسِنِينَ

“Berinfaklah di jalan Allah, janganlah jerumuskan dirimu ke dalam kebinasaan, dan berbuat baiklah. Sesungguhnya Allah menyukai orang-orang yang berbuat baik.” (Q.S Al-Baqarah: 195)

Dari ayat tersebut ditafsirkan bahwa orang-orang mukmin diperintahkan untuk membelanjakan harta kekayaannya untuk berjihad dan dilarang menjatuhkan diri ke dalam jurang kebinasaan. Orang harus mempunyai niat yang baik agar dia senantiasa mendapatkan pertolongan Allah SWT dalam hal *infaq fi sabīlillāh* (NU Online, 2023a). Ayat ini telah menjadi landasan yang jelas bahwa sebagai manusia perlu melakukan usaha untuk berperan dalam upaya mengurangi risiko bencana alam melalui ilmu pengetahuan alam dan teknologi.

Maka untuk mengatasi permasalahan mengenai penilaian kerusakan bangunan pasca bencana alam dan sebagai upaya dalam membantu mitigasi bencana alam dapat dengan mengimplementasikan *Deep Learning* untuk membangun sebuah sistem penilaian kerusakan bangun pasca bencana alam secara otomatis. Pendekatan yang digunakan menggunakan *Convolutional Neural*

Network (CNN) yang dapat memproses data yang memiliki topologi seperti *grid*, contohnya yaitu data dalam bentuk gambar. CNN memiliki performa yang sangat baik dalam melakukan analisis terhadap citra.

Model yang digunakan pada penelitian ini adalah DenseNet (*Densely Connected Convolutional Networks*) 121 yang dinilai dapat mengekstrak informasi detail pada gambar kerusakan bangunan karena memanfaatkan koneksi yang padat pada antar lapisannya. Penelitian terdahulu pada umumnya membutuhkan dataset yang besar untuk implementasi DenseNet-121 dalam menjalankan tugas seperti klasifikasi dan ekstraksi fitur. Namun belum ada penelitian yang menggunakan DenseNet-121 untuk tugas ekstraksi fitur khususnya pada citra kerusakan bangunan pasca bencana alam tanpa adanya pemisahan data seperti yang umumnya dilakukan jika menggunakan model CNN. Penelitian ini hadir untuk mengisi kekosongan tersebut.

Diharapkan proses identifikasi tingkat kerusakan bangunan dapat dilakukan dengan lebih cepat dan akurat dibandingkan dengan metode penilaian tingkat kerusakan bangunan pasca bencana alam secara manual. Melalui penelitian ini diharapkan dapat memberikan solusi baru yang inovatif untuk mendukung percepatan mitigasi bencana melalui mengaplikasikan teknik pada *Deep Learning* dengan memanfaatkan metode CNN beserta DenseNet-121 sebagai model yang digunakan dalam membangun sistem yang diinginkan. Diharapkan proses mitigasi bencana alam dapat dilakukan dengan lebih mudah, lebih cepat, dan menghasilkan penilaian yang akurat serta objektif sehingga proses pemulihan pasca bencana alam dapat dilaksanakan dengan lebih cepat.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, masalah yang dapat dirumuskan adalah:

1. Bagaimana implementasi DenseNet-121 untuk ekstraksi fitur pada citra kerusakan bangunan pasca bencana alam?
2. Bagaimana pengukuran terhadap hasil ekstraksi fitur dilakukan?

1.3 Batasan Masalah

Agar penelitian dapat berjalan sesuai dengan rumusan masalah dan tujuan penelitian maka diperlukan batasan masalah yang jelas. Berikut ini merupakan batasan masalah yang telah ditetapkan pada penelitian ini:

1. Dataset yang digunakan pada penelitian adalah data berupa gambar kerusakan bangunan yang diperoleh dari platform Kaggle.
2. Penelitian hanya dilakukan dengan *Convolutional Neural Network (CNN)* yang menggunakan model DenseNet-121.
3. Fokus utama dari penelitian ini berada pada tahap ekstraksi fitur dari gambar bangunan rusak pasca bencana alam.

1.4 Tujuan Penelitian

Dari masalah yang telah dirumuskan, tujuan dari penelitian ini adalah:

1. Mengimplementasikan DenseNet-121 untuk ekstraksi fitur pada citra kerusakan bangunan pasca bencana alam.
2. Menganalisis pengukuran terhadap hasil ekstraksi fitur.

1.5 Manfaat Penelitian

Dengan adanya penelitian ini, diharapkan proses penilaian tingkat kerusakan bangunan yang terdampak bencana alam dapat dilakukan dengan lebih cepat namun memiliki tingkat akurasi yang tinggi dan menghasilkan hasil penilaian yang objektif. Dengan begitu, proses pemulihan dan pembangunan ulang infrastruktur pasca bencana alam dapat dilaksanakan lebih cepat dan sesegera mungkin.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terkait

Studi literatur terhadap penelitian terdahulu yang berhubungan dengan penelitian ini dilakukan agar dapat menghimpun penelitian terkait. Dari penelitian terkait ini peneliti dapat mengidentifikasi apa saja yang sudah diketahui, menemukan informasi-informasi baru terhadap pendekatan atau metode yang digunakan, dan untuk menemukan pembaruan pada penelitian terdahulu sehingga peneliti dapat memberikan kontribusi pada bidang terkait. Maka, pada subbab ini akan diuraikan beberapa penelitian terdahulu yang terkait dengan penelitian ini.

Penelitian oleh Zhou et al. (2022) telah membuktikan bahwa DenseNet dapat meningkatkan fitur yang diekstrak dan mengurangi masalah gradien yang menghilang. Selanjutnya dari aplikasi DenseNet pada bidang lain yakni klasifikasi kanker payudara, akurasi klasifikasi ditingkatkan dengan memanfaatkan fitur yang diekstraksi dari gambar mammogram pada penelitian oleh Pattanaik et al. (2022). Pendekatan yang diterapkan pada penelitian yang dilakukan oleh Fekri-Ershad et al. (2022) juga membuktikan bahwa ekstraksi fitur mendalam yang dilakukan DenseNet efektif digunakan dibandingkan dengan model tradisional lain untuk klasifikasi *cell phenotype*.

Hasan et al. (2021) telah membuktikan bahwa DenseNet-121 dapat mencapai hasil akurasi yang melebihi skor 92% untuk klasifikasi *CT scan* yang digunakan untuk memprediksi Covid-19. Sedangkan Jeyabose. et al. (2022) dalam

penelitiannya untuk klasifikasi penyakit daun pada berbagai jenis tumbuhan menunjukkan hasil akurasi yang sangat tinggi yakni 99.81% berhasil dicapai oleh DenseNet-121.

Tabel 2. 1 Penelitian Terkait

Referensi	Topik	Metode	Subjek	Hasil Penelitian
(Elharrouss et al., 2024)	Aplikasi kecerdasan buatan (AI) untuk memahami data menggunakan teknik yang berbeda	CNN dan DRL (<i>Deep Reinforcement Learning</i>)	Teknik pada kecerdasan buatan (AI), deteksi objek, pengenalan wajah, pengenalan tindakan, dengan fokus pada DRL	Perbandingan hasil evaluasi berbagai DL (Deep Learning) networks yang digunakan sebagai backbone untuk menjalankan berbagai tugas visi komputer
(Fekri-Ershad et al., 2022)	Klasifikasi fenotip sel	CNN (DenseNet)	Pengembangan pendekatan baru untuk menggabungkan ekstraksi fitur tingkat tinggi dan rendah menggunakan DenseNet untuk meningkatkan akurasi pada klasifikasi fenotip sel yang penting untuk digunakan dalam bidang medis	Pendekatan baru yang diaplikasikan menghasilkan akurasi yang tinggi pada dua dataset yang berbeda (95,59% dan 93,36%) dengan lebih cepat jika dibandingkan dengan metode lain
(Pattanaik et al., 2022)	Klasifikasi kanker payudara	ELM (<i>Extreme Learning Machine</i>) berbasis DenseNet-121	Klasifikasi kanker payudara melalui gambar mammogram menggunakan ELM berbasis DenseNet-121	Model yang digunakan mencapai tingkat akurasi 99,14% sehingga dapat membantu diagnosis otomatis di bidang medis
(Ahmed et al., 2024)	Penerapan CNN, ekstraksi fitur, dan klasifikasi gambar	CNN (VGG-19, DenseNet, dan AlexNet)	Metodologi lanjutan untuk analisis gambar dan ekstraksi fitur tingkat tinggi dan rendah	Metode yang digunakan menghasilkan klasifikasi gambar dengan presisi tinggi untuk membedakan objek <i>foreground</i> dan <i>background</i> pada dataset (Caltech-256, Cifar-10, dan Corel-1000).
(Musleh et al., 2023)	Keamanan IoT, aplikasi ML (<i>Machine Learning</i>) di	KNN (<i>K-Nearest Neighbor</i>), SVM (<i>Support</i>	Evaluasi IDS pada ML khusus untuk IoT dan evaluasi akurasi model ML	Model untuk mendeteksi ancaman IoT berhasil dikembangkan

Referensi	Topik	Metode	Subjek	Hasil Penelitian
	IDS, dan evaluasi metode ekstraksi fitur	<i>Vector Machine</i>), IDS (<i>Intrusion Detection System</i>), CNN (DenseNet dan VGG-16), dan SMOTE (<i>Synthetic Minority Over-sampling Technique</i>)	untuk meningkatkan kinerja IDS	dengan mencapai tingkat akurasi 98,3%
(Zhou et al., 2022)	Diagnosis kesalahan <i>rolling bearings</i> pada bidang mekanik dan kelistrikan	GAF (<i>Gramian Angular Field</i>), CNN, TL (<i>Transfer Learning</i>)	Diagnosis kesalahan karena parameter karakteristik yang berlebihan dan kopling sinyal yang parah dalam <i>rolling bearings</i>	Kombinasi GAF dan CNN dapat mendeteksi kerusakan <i>bearing</i> secara akurat
(Hasan et al., 2021)	Prediksi Covid-19	CNN	Prediksi pasien Covid-19 menggunakan CNN melalui gambar <i>CT scan</i> pasien	Model DenseNet-121 dapat mendeteksi pasien Covid-19 dengan tingkat akurasi 92% dengan waktu yang cepat selama 3 menit
(Jeyabose et al., 2022)	Identifikasi penyakit pada tanaman	CNN (Densenet-121, ResNet-50, VGG-16, dan Inception V4)	Peningkatan akurasi deteksi penyakit melalui beberapa <i>pretrained</i> model CNN yang dipilih	DenseNet-121 paling unggul dibanding model lain karena dapat mengklasifikasikan 38 jenis penyakit tanaman dengan akurasi paling tinggi sebesar 99,81%
(Almais et al., 2024)	Penilaian tingkat kerusakan bangunan pasca bencana alam secara otomatis	GLCM (<i>Gray Level Co-occurrence Matrix</i>) dan PCA (<i>Principal Component Analysis</i>)	Klasifikasi tingkat kerusakan bangunan pasca bencana alam	Integrasi antara GLCM dan PCA terbukti efektif untuk klasifikasi tingkat kerusakan bencana alam dengan hasil yang sesuai dengan penilaian surveyor

Dari beberapa penelitian yang terkait telah menunjukkan hasil positif bahwa penggunaan DenseNet-121 untuk berbagai tugas seperti ekstraksi fitur, deteksi penyakit dalam bidang medis dan pertanian, klasifikasi gambar, dan diagnosis

kesalahan pada bidang mekanik dan kelistrikan. Namun Almais et al. (2024) menyebutkan bahwa CNN memerlukan data yang lebih banyak dan komputasi yang lebih tinggi dibandingkan dengan integrasi metode GLCM-PCA untuk topik kerusakan bangunan pasca bencana alam. Tetapi perlu diperhatikan bahwa belum ada penelitian terkait yang melakukan ekstraksi fitur untuk kerusakan bangunan yang terdampak bencana alam menggunakan DenseNet-121 yang merupakan model CNN tanpa memerlukan banyak data dan tidak melakukan pemisahan data khususnya untuk topik kerusakan bangunan pasca bencana alam. Maka dari itu peneliti ingin mengisi *gap* tersebut dengan mengimplementasikan DenseNet-121 untuk ekstraksi fitur kerusakan bangunan yang terdampak bencana alam serta untuk mengeksplorasi apakah model layak untuk diimplementasikan walaupun tidak dilakukan pemisahan data dan tidak menggunakan banyak data.

2.2 Bencana Alam

Bencana alam adalah suatu peristiwa yang terjadi karena proses alamiah yang merugikan bagi kehidupan manusia, lingkungan, dan harta benda (Faturahman, 2021). Sedangkan Badan Nasional Penanggulangan Bencana Alam (BNPB) memaknai bencana alam sebagai bencana yang diakibatkan oleh peristiwa atau serangkaian peristiwa yang disebabkan oleh alam antara lain berupa gempa bumi, tsunami, gunung meletus, banjir, kekeringan, angin topan, dan tanah longsor. Umumnya jenis bencana alam yang sering terjadi di Indonesia meliputi badai, banjir, gempa bumi, tanah longsor, dan letusan gunung api.

Indonesia merupakan salah satu negara yang paling rawan terhadap bencana alam karena terletak di kawasan Cincin Api Pasifik (Ring of Fire) dimana Lempeng

Eurasia, Pasifik, Filipina, dan Indo-Australia bertemu sehingga menyebabkan aktivitas vulkanik dan seismik yang tinggi (Rohadi, 2015). Sebagai contoh, peristiwa bencana alam yang melanda Kota Palu pada tahun 2018 mengakibatkan kerusakan parah dan memakan korban jiwa karena kombinasi dari gempa supershear dan tanah longsor pesisir yang mengakibatkan terjadinya tsunami (Schambach et al., 2021). Sebagai contoh lain, jika bergeser pada daerah Papua Barat memiliki risiko tinggi terjadi tsunami karena topografi pesisir dan kesiapsiagaan yang kurang dari masyarakat (Naryanto, 2019).

Penelitian yang dilakukan oleh Pratama et al. (2019) juga membuktikan bahwa gempa bumi yang sering terjadi di Indonesia baik dalam skala kecil sampai menengah berpengaruh terhadap terjadinya deformasi kerak bumi yang juga dapat meningkatkan risiko terjadinya bencana yang lebih lanjut. Kerentanan Indonesia terhadap bencana alam ini tidak hanya terletak pada frekuensi terjadinya bencana alam, tetapi juga pada dampak negatif yang diakibatkan. Bencana alam yang terjadi tidak hanya menimbulkan kerusakan pada infrastruktur tetapi juga mengakibatkan kerusakan lingkungan dan melumpuhkan berbagai aspek kehidupan manusia seperti gangguan pada struktur sosial, gangguan kesehatan baik secara fisik maupun mental (De Rubeis et al., 2021; Keya et al., 2023), kemunduran pertumbuhan ekonomi (Noy & Uher, 2022), dll.

Mitigasi bencana alam membutuhkan tindakan tepat waktu dan alokasi sumber daya yang efektif (Nikolić et al., 2023). World Health Organization (WHO) juga menetapkan bahwa proses penilaian risiko cepat atau *Rapid Risk Assessment*

(RRA) seharusnya dilakukan dalam kurun waktu 24-48 jam setelah terjadinya bencana alam (Goode et al., 2021).

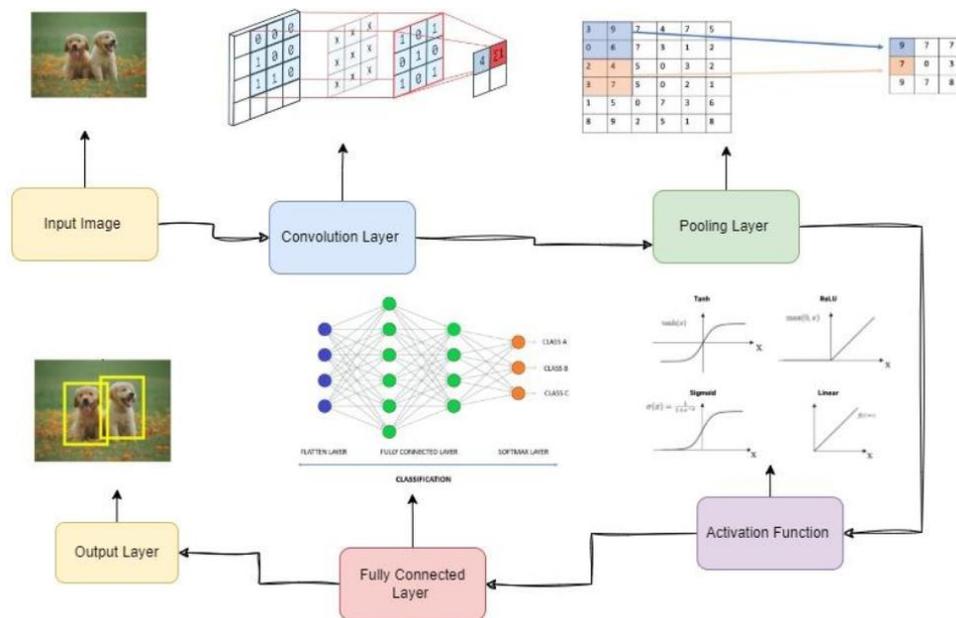
Teknis terhadap penilaian terhadap kerusakan bangunan telah ditetapkan melalui Peraturan Menteri Pekerjaan Umum No. 45/PRT/M/2007 yang terdapat dalam lampiran PermenPUPR No. 29/PRT/M/2018 tentang Standar Teknis Standar Pelayanan Minimal Pekerjaan Umum dan Perumahan Rakyat. Namun standar teknis untuk penilaian kerusakan bangunan masih dilakukan secara manual. Tahapan-tahapan yang perlu dilakukan untuk pengumpulan data memiliki rangkaian yang panjang dimulai dari identifikasi perumahan di lokasi rawan bencana, identifikasi lahan-lahan potensial sebagai lokasi relokasi perumahan, survei data rumah korban bencana alam bersama Badan Penanggulangan Bencana Daerah (BPBD), pendataan tingkat kerusakan rumah oleh Tim Satgas BPBD, verifikasi penerima layanan SPM (Standar Pelayanan Minimal), penyusunan rencana aksi, sampai pelaporan pelaksanaan SPM. Terdapat 3 kategori tingkat kerusakan bangunan yang telah ditetapkan, yaitu rusa ringan (Kelas A, Kelas B, Kelas C), rusak sedang, dan rusak berat. Proses klasifikasi tingkat kerusakan bangunan sesuai dengan kategori tersebut dilakukan secara manual, yakni dengan melakukan survei dan mengamati kerusakan secara visual yang dilakukan oleh tenaga ahli dengan skor persentase yang berbeda untuk setiap kategori.

Dari standar teknis yang telah disahkan tersebut, masih terdapat kekurangan karena penilaian kerusakan bangunan secara manual memakan waktu yang lama. Di samping itu, terdapat risiko terhadap penilaian secara manual menghasilkan penilaian yang bersifat subjektif. Hal tersebut dapat terjadi karena perbedaan

interpretasi dan pengalaman yang membuatnya rentan terhadap variasi akurasi di antara para ahli yang berbeda. Proses penilaian kerusakan bangunan pasca bencana alam secara manual juga berisiko besar bagi para ahli yang harus pergi ke lapangan terutama di daerah yang masih belum stabil pasca bencana alam. Dibutuhkan juga biaya yang relatif tinggi karena membutuhkan peralatan, transportasi, dan sumber daya manusia (SDM) yang memadai untuk menuju area yang terdampak bencana alam.

2.3 Convolution Neural Network (CNN)

Convolutional Neural Network atau yang biasa dikenal sebagai CNN atau ConvNet merupakan jaringan saraf tiruan yang terdiri dari *multi-layer neural network* yang dapat mengenali, mengidentifikasi, dan mengklasifikasi sebuah objek melalui pemrosesan citra (Taye, 2023). CNN memiliki kemampuan untuk memperoleh fitur dari kumpulan data yang luas secara mandiri melalui transformasi nonlinier *multi-layer* (Du, 2024). Kemampuan tersebut dapat mengurangi kebutuhan terhadap fitur yang dibuat secara manual karena jaringan dapat mengidentifikasi dan mengekstrak pola yang relevan langsung dari data.



Gambar 2. 1 Komponen CNN (Taye, 2023)

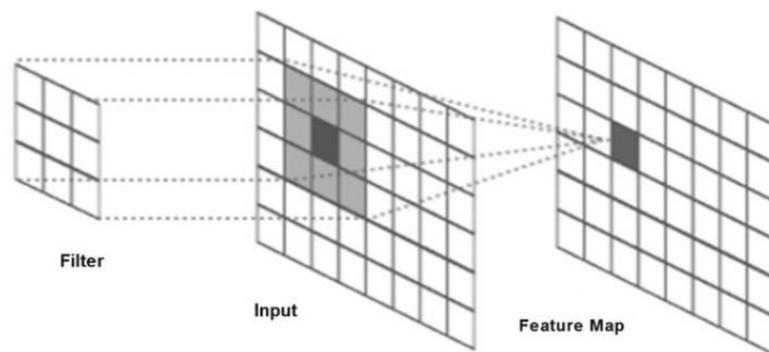
CNN terdiri dari beberapa komponen penyusun yang memiliki tugas masing-masing. Desain modular pada komponen CNN memungkinkan penggabungan berbagai modul untuk mencapai *task* tertentu dengan arsitektur yang lebih kompleks (Qi et al., 2024). Gambar 2.1 memberikan gambaran terhadap struktur komponen penyusun CNN yang dibagi menjadi enam bagian (Taye, 2023):

2.3.1 Input Image

Data awal yang digunakan berupa citra yang tersusun dari piksel. Setiap piksel pada citra memiliki nilai dimulai dari 0 sampai 255, dimana nilai 0 ini berwarna hitam pekat dan nilai 255 adalah putih terang. Sedangkan angka di antara 0-255 tersebut merupakan warna *grayscale* dan kombinasi warna dasar RGB (*Red, Green, and Blue*). Piksel sendiri diatur dalam susunan seperti matriks secara berurutan pada *layout* citra digital.

2.3.2 Convolution Layer

Lapisan konvolusi merupakan blok bangunan inti dari CNN yang bertugas untuk mengestrak fitur pada citra menggunakan filter yang dapat dipelajari atau disebut dengan kernel untuk menghasilkan *feature map* (Wani et al., 2020). Pada lapisan inilah terdapat *hyperparameter* yang digunakan untuk mengontrol perilaku dari model yang digunakan.



Gambar 2. 2 Operasi konvolusi (Wani et al., 2020)

Operasi konvolusi dapat diterapkan pada citra dua dimensi dengan memperlakukan citra dua dimensi sebagai fungsi dua dimensi dengan melakukan berbagai transformasi. Operasi konvolusi dapat dilakukan dengan menerapkan persamaan berikut ini (Wardani & Leonardi, 2023):

$$O_{i,j} = \sum_{M=0}^{M-1} \sum_{N=0}^{N-1} I_{(i+m,j+n)} * K_{(m,n)} \quad (2.1)$$

Keterangan:

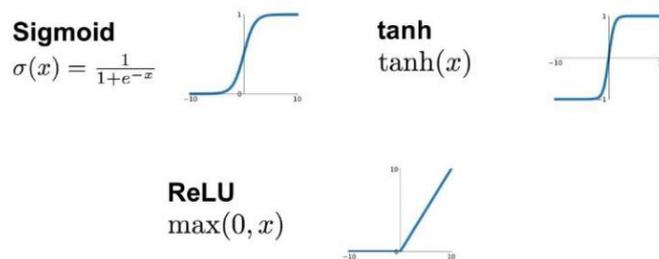
- | | |
|--------------------|--|
| a) $O_{i,j}$ | = nilai matriks keluaran posisi (i, j) |
| b) $I_{(i+m,j+n)}$ | = nilai matriks masukan |
| c) * | = proses konvolusi |
| d) $K_{(m,n)}$ | = nilai kernel |
| e) i | = koordinat x pada citra |
| f) j | = koordinat y pada citra |
| g) M | = lebar citra |
| h) N | = tinggi citra |
| i) m | = variabel iterasi untuk lebar citra |

j) n = variabel iterasi untuk tinggi citra

2.3.3 Pooling Layer

Lapisan ini bisa disebut sebagai lapisan penyatuan yang mengurangi dimensi *feature map* dengan tetap mempertahankan informasi penting. Informasi yang kurang signifikan akan dibuang oleh lapisan ini tetap dengan mempertahankan informasi penting dalam representasi yang lebih kecil. Terdapat banyak teknik untuk melakukan penyatuan pada CNN seperti *average pooling*, *max pooling*, *global average pooling*, *mixed pooling method*, *tree pooling*, *spectral pooling*, dll.

2.3.4 Activation Function Layer



Gambar 2. 3 Fungsi aktivasi (Taye, 2023)

Fungsi aktivasi dibutuhkan untuk memperkenalkan nonlinearitas sehingga memungkinkan model mempelajari pola yang kompleks. Pada penelitian ini akan digunakan fungsi aktivasi ReLU karena dapat berlatih dengan lebih cepat dibandingkan fungsi aktivasi tanh dan sigmoid. Fungsi aktivasi ReLU sendiri dapat dituliskan pada persamaan 2.2.

$$f(x) = \max(0, x) \quad (2.2)$$

Dari persamaan 2.2 dapat diketahui jika fungsi aktivasi ReLU menghitung aktivasi dengan melakukan *thresholding* pada input di nilai nol. Dengan kata lain,

unit linier yang diperbaiki memiliki *output* 0 jika input kurang dari 0, dan *output* mentah sebaliknya (Wani et al., 2020).

2.3.5 Fully Connected Layer

Pada dasarnya pada CNN ini terdiri dari dua tahap utama yaitu tahap ekstraksi dan tahap klasifikasi. Lapisan FC (*Fully Connected*) ini merupakan lapisan yang berada pada tahap klasifikasi setelah dilakukan ekstraksi fitur pada lapisan-lapisan sebelumnya untuk tahap ekstraksi fitur. Lapisan FC ini bekerja dengan cara menggabungkan fitur yang telah diekstrak pada lapisan sebelumnya untuk menghasilkan *output*.

2.3.6 Output Layer

Lapisan ini menghasilkan prediksi akhir dan menghitung *loss function* untuk memperbarui bobot selama pelatihan. Kemampuan CNN untuk mengekstrak fitur secara otomatis sangat efektif di berbagai aplikasi, seperti pengenalan wajah, deteksi objek, klasifikasi umur dan gender, dan analisis citra medis (Basir & Shantta, 2021; Benkaddour, 2021; Taye, 2023). Dalam bidang penilaian kerusakan bangunan pascabencana, CNN dapat digunakan untuk meneliti gambar struktur yang rusak dan mengekstrak fitur tingkat kerusakan bangunan yang terjadi. CNN dapat mengidentifikasi berbagai pola kerusakan (Alqethami et al., 2022; Fan & Chung, 2022), memfasilitasi penilaian yang lebih cepat dan lebih tepat dibandingkan dengan teknik manual. Maka, CNN dapat digunakan untuk mempersingkat waktu yang dibutuhkan untuk membantu proses penilaian

kerusakan bangunan yang terdampak bencana alam dan juga meningkatkan tingkat akurasi (Bouchard et al., 2022; Dunphy et al., 2022).

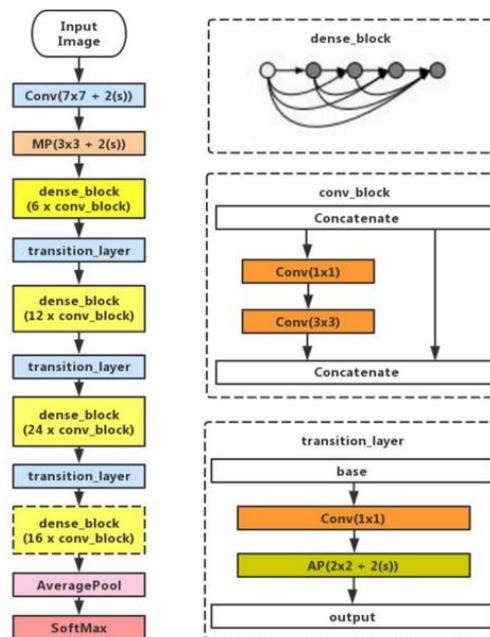
2.4 DenseNet-121

DenseNet-121 merupakan salah satu arsitektur yang dimiliki oleh CNN. DenseNet-121 pertama kali diperkenalkan oleh Huang et al. (2017) dengan tujuan untuk memaksimalkan aliran informasi untuk setiap layer pada jaringan. DenseNet-121 terdiri dari 121 lapisan-lapisan dengan konektivitas yang padat dimana setiap lapisan menerima input dari semua lapisan sebelumnya. Arsitektur ini didesain untuk mengatasi *vanishing gradient*, mendorong kembali penggunaan fitur, serta menghemat parameter dengan memanfaatkan konektivitas padatnya dan penggunaan lapisan *bottleneck* sehingga dapat mengurangi biaya komputasi (Huang et al., 2017; Masud, 2022).

Densenet-121 tersusun oleh empat *dense block* yang pada setiap *dense block* tersebut dipisahkan oleh *transition layers*. Setiap *dense block* terdiri dari beberapa *convolutional block* yang terhubung satu sama lain. *Dense block* sendiri merupakan bangunan blok dasar dari DenseNet. *Dense block* ini terdiri dari beberapa lapisan konvolusi yang diikuti fungsi aktivasi. Jika *dense block* memiliki m lapisan dan setiap lapisannya menghasilkan k *feature maps*, maka lapisan ke- l memiliki $k(l + l_0)$ input *feature maps* dimana l_0 merupakan input *channel* menuju *dense block*.

Sedangkan *transition layer* Merupakan lapisan yang menghubungkan *dense blocks* yang biasanya terdiri dari *batch normalization* untuk normalisasi *feature maps*, 1×1 *convolution* untuk mengurangi jumlah *feature maps*, dan *pooling layer*.

Berikut ini merupakan struktur dari arsitektur DenseNet-121 beserta rincian lebih lanjut agar lebih memahami *dense block*, *convolutional block*, *transition layer* seperti yang dapat dilihat pada Gambar 2.4.



Gambar 2. 4 (Kiri) Arsitektur DenseNet-121. (Kanan) Dense_block, conv_block dan transition_layer (Ji et al., 2019)

DenseNet-121 hanya menerima *input* berupa citra dengan ukuran 224x224x3 yang kemudian ketika sudah diproses akan menghasilkan *feature maps* 1024 dimensi melalui GAP (*Global Average Pooling*). Tetapi citra yang digunakan sebagai masukan perlu melalui tahap praproses data (*data preprocessing*) yang terdiri dari beberapa tahap. Tahapan praproses data tersebut perlu dilakukan agar citra yang digunakan sebagai masukan dapat diproses oleh DenseNet-121.

Tahap pertama untuk praproses data adalah melakukan *resize* terhadap citra agar memiliki ukuran yang sesuai dengan lapisan *input* DenseNet-121. Citra diubah

ukurannya ke resolusi tetap berukuran 224×224 piksel menggunakan interpolasi bilinear yang dapat dituliskan dalam persamaan berikut ini:

$$I'(x', y') = \sum_{i=0}^1 \sum_{j=0}^1 I(x_i, y_j) \cdot (1 - |x' - x_i|) \cdot (1 - |y' - y_j|) \quad (2.3)$$

Keterangan:

- a) $I(x, y)$ = nilai piksel citra asli pada koordinat x dan y
- b) $I'(x', y')$ = nilai piksel citra yang di-*resize* pada koordinat x' dan y'
- c) x_i, y_j = titik koordinat piksel dari citra asli untuk interpolasi
- d) $\sum_{i=0}^1 \sum_{j=0}^1$ = penjumlahan untuk semua kombinasi i dan j dari 0 sampai 1

Setelah citra diubah ukurannya menjadi 224×224 piksel, citra selanjutnya perlu dipastikan agar memiliki format RGB (*Red, Green, Blue*) yang memiliki tiga *channel* warna. Representasi array RGB dapat dituliskan sesuai dengan Persamaan 2.2 berikut ini:

$$I \in R^{224 \times 224 \times 3} \quad (2.4)$$

Keterangan:

- a) I = tensor citra RGB
- b) R = himpunan bilangan *real*
- c) $224 \times 224 \times 3$ = dimensi spasial (tinggi, lebar, dan jumlah *channel* warna)

Untuk mengonversi citra ke dalam format RGB jika citra awalnya dalam format *grayscale* yang hanya memiliki satu *channel* dilakukan dengan mereplikasi nilai intensitas ke seluruh *channel* RGB:

$$G \in R^{224 \times 224}, I_{RGB} = \text{stack}(G, G, G) \quad (2.5)$$

Keterangan:

- a) G = citra *grayscale*
- b) I_{RGB} = hasil konversi citra *grayscale* menjadi RGB
- c) $\text{stack}(G, G, G)$ = gabungan tiga salinan G di sepanjang sumbu *channel* yang menghasilkan bentuk $224 \times 224 \times 3$

Jika citra awalnya dalam format RGBA yang memiliki empat *channel* warna yaitu *red*, *green*, *blue*, dan *alpha* maka hanya tiga *channel* awal yang diambil:

$$I_{RGB} = I_{RGBA}[:, :, 0:3] \quad (2.6)$$

Keterangan:

- a) I_{RGB} = tensor hasil potongan citra RGB 3 *channel*
- b) I_{RGBA} = tensor citra RGBA 4 *channel*
- c) $[:, :, 0:3]$ = operasi *slicing* untuk mengambil *channel* ke-0 sampai ke-2 (RGB)

Tahap praproses data selanjutnya yaitu normalisasi nilai piksel yang awalnya berada pada rentang 0-255 agar berada dalam rentang [0,1]. Normalisasi dilakukan dengan membagi setiap nilai intensitas piksel dengan 255 seperti pada persamaan 2. 7 ini:

$$I_{norm}(x, y, c) = \frac{I(x, y, c)}{255} \quad (2.7)$$

Keterangan:

- a) $I_{norm}(x, y, c)$ = nilai intensitas piksel pada baris ke- x , kolom ke- y , dan *channel* warna ke- c (*red*, *green*, atau *blue*)
- b) $I_{norm}(x, y, c)$ = nilai piksel setelah normalisasi
- c) 255 = nilai maksimum piksel (8 *bit per channel*)

Setelah citra telah dinormalisasi akan memiliki tiga dimensi (224, 224, 3) yang kemudian diproses ke tahap praproses data lanjutan yaitu menambah dimensi *batch*. Pada dasarnya DenseNet-121 menerima *input* yang memiliki empat tensor dengan dimensi (B, H, W, C) dimana B mewakili jumlah citra dalam satu *batch*, H mewakili tinggi citra, W mewakili lebar citra, dan C mewakili jumlah *channel* warna citra. Penambahan dimensi *batch* ini dilakukan dengan menambahkan satu dimensi di awal agar tensor berubah menjadi (1, 224, 224, 3). Tahap ini dapat dituliskan pada persamaan berikut:

$$X_{batch}[0, i, j, c] = X[i, j, c] \quad (2.8)$$

Keterangan:

- a) $X_{batch}[0, i, j, c]$ = nilai piksel baris ke- i , kolom ke- j , dan channel warna ke- c
- b) $X[i, j, c]$ = tensor empat dimensi dengan indeks *batch* ke-0

Tahap akhir dari praproses data adalah augmentasi data yang dilakukan dengan melakukan berbagai transformasi terhadap citra *input*. Augmentasi data dilakukan untuk menambah keragaman atau menambah variasi dari citra *input*. Untuk tugas ekstraksi fitur pada citra kerusakan bangunan pasca bencana alam, lapisan klasifikasi pada bagian akhir jaringan dihapus karena pada penelitian in DenseNet-121 fokus untuk melakukan ekstraksi fitur saja. DenseNet-121 merupakan *pretrained model* atau model yang telah dilatih pada dataset ImageNet.

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}]) \quad (2.9)$$

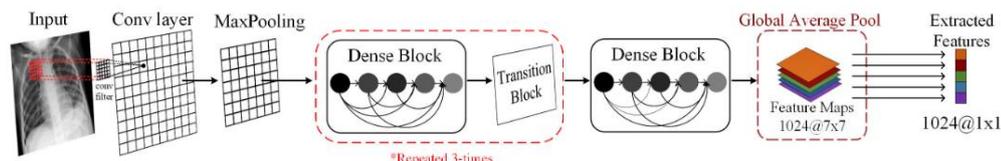
Keterangan:

- a) x_l = *output* dari lapisan ke- l
- b) H_l = fungsi transformasi pada lapisan ke- l
- c) $[x_0, x_1, \dots, x_{l-1}]$ = penggabungan semua *output* dari lapisan

Dense connection atau koneksi padat yang dimiliki oleh DenseNet-121 dituliskan seperti pada Persamaan 3.6 secara matematis dimana x_l adalah *output* yang dihasilkan lapisan ke- l . Lapisan ke- l menerima *feature map* dari seluruh lapisan sebelumnya yaitu x_0, x_1, \dots, x_{l-1} sebagai *input*. Kemudian H_l adalah fungsi gabungan dari normalisasi *batch*, fungsi aktivasi ReLU, dan konvolusi (Huang et al., 2017). Adanya *dense connection* atau koneksi padat pada DenseNet-121 dapat meningkatkan efisiensi parameter serta *reuse* informasi antar lapisan (Gu et al., 2018).

DenseNet-121 dipilih untuk penelitian in karena sudah terbukti unggul untuk mengekstrak fitur pada citra *X-Ray* dada yang digunakan untuk deteksi dini

Covid-19 pada penelitian oleh Ahishali et al. (2021) karena menghasilkan fitur yang bervariasi namun tetap tereduksi dengan baik. Berikut ini merupakan Gambar 2.5 yang merupakan ilustrasi alur dari ekstraksi fitur yang dilakukan pada penelitian oleh Ahishali et al. (2021).



Gambar 2. 5 *Feature extraction pipeline* dari model *pre-trained* DenseNet-121 (Ahishali et al., 2021)

Gambar 2.5 merupakan diagram yang menunjukkan alur ekstraksi fitur menggunakan DenseNet-121 pada citra *X-Ray* dada. Dari diagram tersebut dapat diketahui jika citra yang digunakan untuk *input* akan mulai diproses melalui lapisan konvolusi, *max pooling*, melalui *dense block* dan *transition layer* dengan iterasi tiga kali, melalui *dense block* terakhir, kemudian diproses oleh GAP (*Global Average Pooling*) yang menghasilkan fitur vektor dengan dimensi (1, 1024). Diagram tersebut dapat memberikan gambaran untuk proses ekstraksi fitur pada penelitian ini meskipun terdapat perbedaan pada jenis citra yang digunakan yaitu citra kerusakan bangunan pasca bencana alam.

Setiap lapisan yang dimiliki oleh DenseNet-121 memiliki akses ke gradien dari *loss function* dan input awal. Sehingga DenseNet-121 dapat membantu mengurangi permasalahan gradien hilang saat pelatihan, karena konektivitas padat pada DenseNet-121 memungkinkan aliran gradien di seluruh lapisan dan meningkatkan pemanfaatan fitur dengan parameter yang lebih sedikit (Masud, 2022; Zhang et al., 2024). DenseNet-121 menjadi pilihan tepat untuk dapat

melakukan ekstraksi fitur dengan cepat dan mencapai hasil yang akurasi tinggi dengan parameter yang lebih sedikit (Huang et al., 2022) untuk membantu proses pemulihan pasca bencana alam.

4.1 *Cosine Similarity*

Persamaan kosinus atau *cosine similarity* pertama kali diperkenalkan oleh Salton et al. (1975) untuk *information retrieval* sebagai pendekatan untuk mengukur kedekatan semantik antar dokumen. *Cosine similarity* merupakan sebuah metode pengukuran untuk mencari adanya kesamaan atau kemiripan antara dua vektor dalam ruang yang berdimensi tinggi. Metode ini umumnya digunakan untuk membandingkan representasi visual pada bidang *deep learning* karena stabil terhadap perbedaan skala dan dapat digunakan untuk mengetahui konsistensi fitur antar citra (Salton et al., 1975).

Cosine similarity ini dihitung berdasarkan arah vektor beserta nilai sudutnya. *Cosine similarity* dapat dituliskan ke dalam persamaan berikut ini secara matematis:

$$\text{Cosine Similarity} = \frac{A \cdot B}{\|A\| \cdot \|B\|} \quad (2.10)$$

Keterangan:

- a) $A \cdot B$ = perkalian *dot product* antara dua vektor
- b) $\|A\| \cdot \|B\|$ = panjang (norma) setiap vektor

4.2 *Principal Component Analysis (PCA)*

Principal component analysis atau PCA merupakan teknik yang dikembangkan oleh Karl Pearson (1901). Tujuan dari teknik ini adalah melakukan representasi data yang berdimensi tinggi ke dalam dimensi rendah tetapi dengan

mempertahankan sebanyak mungkin variasi di dalamnya. Dengan kata lain PCA merupakan metode untuk mereduksi dimensi dari data yang multivariat. PCA dapat digunakan untuk mengukur apakah distribusi dari vektor fitur citra augmentasi mirip dengan distribusi vektor fitur dari citra *original* yang dapat menjadi indikator kestabilan representasi numerik (Pearson, 1901).

PCA dilakukan dengan memusatkan data terhadap *mean*, perhitungan kovarian, dan dekomposisi eigen dari matriks kovarian. PCA kemudian melakukan transformasi terhadap data ke dalam komponen utama. PCA dapat dilakukan dengan persamaan berikut ini:

$$Z = X_{centered} \cdot W \quad (2.11)$$

Keterangan:

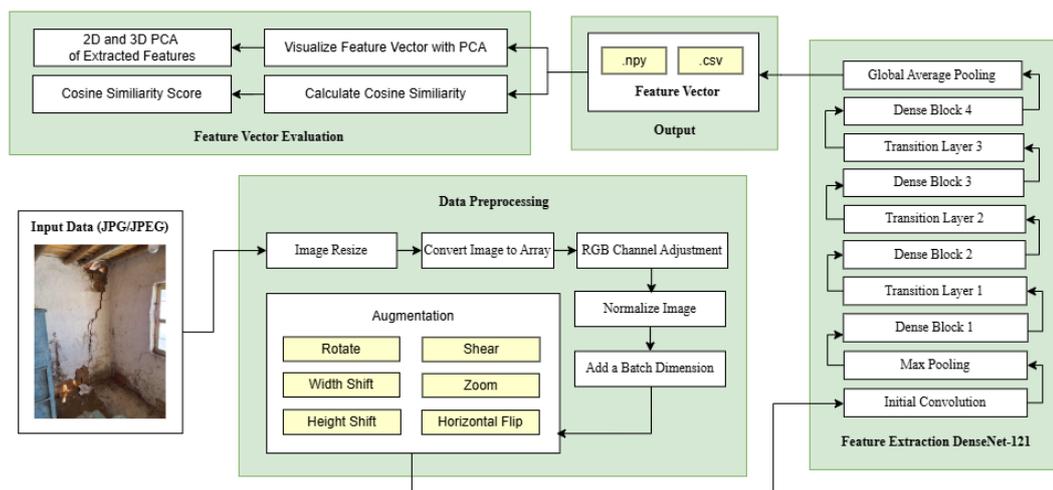
- a) $X_{centered}$ = matriks fitur yang sudah dinormalisasi
- b) W = matriks *eigenvector* dari matriks kovarians X
- c) Z = hasil visualisasi ke ruang dimensi rendah

BAB III

METODE PENELITIAN

3.1 Desain Sistem

Pada bagian ini akan dijelaskan desain sistem dalam penelitian ini yang disusun secara dengan struktur yang jelas agar dapat mencapai tujuan penelitian yang telah ditetapkan. Sistem pada penelitian ini dikembangkan menggunakan *Python* sebagai bahasa pemrograman karena kompatibel dengan *library* yang dibutuhkan. Sedangkan Google Colab dipilih sebagai *programming tool* yang berbasis *cloud*. Secara garis besar, sistem yang dikembangkan pada penelitian ini mengintegrasikan tahapan pemrosesan citra, arsitektur DenseNet-121 yang dimodifikasi, dan evaluasi dari hasil ekstraksi fitur dalam alur yang runtut seperti yang dapat dilihat dari gambar 3.1.



Gambar 3. 1 Desain Sistem

Berikut ini merupakan penjelasan lebih lanjut terhadap alur kerja dari desain sistem yang dikembangkan:

1. *Input Data*

Tahap pertama adalah memasukkan data ke dalam sistem berupa citra kerusakan bangunan pasca bencana alam. Format citra yang dapat diterima oleh sistem yaitu JPG/JPEG. Ukuran data input ini dapat bervariasi karena diambil dari dataset yang digunakan untuk penelitian ini.

2. *Data Preprocessing*

Sistem akan memproses data input dengan melakukan praproses. Tahap ini diperlukan karena data-data di dalam dataset memiliki ukuran dan resolusi yang berbeda-beda, maka praproses data ini dilakukan agar data input memiliki standar. Praproses data dimulai dengan mengubah ukuran data input sesuai dengan ukuran yang dapat diterima oleh DenseNet-121. Setelah itu data input akan diubah bentuknya menjadi array terlebih dahulu.

Sistem juga akan mengecek apakah data input sudah memiliki format RGB (*Red, Green, Blue*), jika belum maka data input akan dikonversi formatnya menjadi RGB. Normalisasi dilakukan untuk memastikan nilai piksel pada data input ini seragam pada rentang 0-1. Penambahan dimensi batch dilakukan sebelum augmentasi data menggunakan berbagai transformasi untuk meningkatkan keragaman data.

3. *Feature Extraction* oleh DenseNet-121

Ekstraksi fitur dilakukan dengan DenseNet-121 yang telah dimodifikasi dengan menghapus lapisan klasifikasi dan menggantinya dengan lapisan GAP (*Global Average Pooling*). Lapisan GAP ini bertugas untuk mereduksi *feature map*

agar menghasilkan representasi numerik dari citra berupa vektor fitur yang lebih padat dengan tetap mempertahankan informasi penting.

Proses ekstraksi fitur pada citra kerusakan bangunan pasca bencana alam oleh DenseNet-121 dimulai dengan melakukan praproses data untuk memastikan agar ukuran citra yang akan diproses oleh DenseNet-121 sesuai dengan ketentuan citra yang diterima DenseNet-121 yaitu memiliki ukuran sebesar $224 \times 224 \times 3$ dengan format RGB (*Red, Green, Blue*). Kemudian citra melalui lapisan konvolusi yang setelahnya terdapat empat *dense block* yang masing-masing memiliki unit konvolusi *batch normalization* dan ReLU. Pada setiap *dense block* terdapat *transition layer* yang bertugas untuk mereduksi dimensi spasial. Di dalam masing-masing *dense block* terdapat. Setelah itu dihasilkan *feature map* dengan ukuran $7 \times 7 \times 1024$ yang selanjutnya dirangkum oleh GAP (*Global Average Pooling*) agar menjadi vektor fitur 1024 dimensi.

4. *Output*

Output yang dihasilkan dari sistem yang dikembangkan pada penelitian ini adalah vektor fitur dari citra original (data input) dan vektor fitur dari seluruh citra augmentasi yang kemudian disimpan dalam format *.npy* dan juga *.csv*.

5. *Evaluasi*

Tahap ini dilakukan untuk mengukur vektor fitur yang dihasilkan pada tahap sebelumnya. Evaluasi dilakukan dengan melakukan perhitungan *cosine similarity* untuk mengetahui apakah terdapat kesamaan antara vektor fitur dari setiap citra hasil augmentasi dengan vektor fitur dari citra *original* yang merupakan data input itu sendiri. Untuk mengetahui persebaran distribusi dari fitur citra *original* dengan

fitur citra augmentasi dibuatlah proyeksi visual dengan menggunakan PCA (*Principal Component Analysis*) dalam bentuk 2D dan 3D.

3.2 Data Preparation

Pada penelitian ini, data yang digunakan diambil dari dataset “*Damaged Constructions Image Dataset*” yang bersumber dari platform Kaggle yang diakses pada tanggal 18 Februari 2025. Dataset tersebut merupakan kumpulan data yang berisi kumpulan gambar bangunan rusak, bangunan tidak rusak, puing-puing bangunan, jalan raya, dan jalan raya yang retak dalam format JPG dan JPEG. Tetapi gambar yang akan digunakan dari dataset tersebut merupakan citra yang objek utamanya berupa bangunan serta puing-puing bangunan, sedangkan untuk gambar yang objek utamanya berupa jalan raya tidak akan digunakan pada penelitian ini.

Pada umumnya penelitian yang mengimplementasikan metode CNN akan memisahkan data atau *split data* terhadap data yang digunakan menjadi data latih, data uji, dan data validasi (Goodfellow et al., 2016). Pada penelitian ini *split data* tidak akan dilakukan karena model DenseNet-121 merupakan model *pretrained* yang telah dilatih pada dataset besar yaitu ImageNet. *Split data* tidak dilakukan juga karena mempertimbangkan bahwa fokus utama pada penelitian ini adalah ekstraksi fitur kemudian mengevaluasi representasi dari fitur yang dihasilkan, bukan untuk *training model classifier* dari awal.

Namun penting untuk diketahui bahwa sebelum diproses lebih lanjut oleh model untuk ekstraksi fitur, data berupa citra yang dimasukkan perlu melalui tahap pra-proses agar formatnya menjadi seragam dan meningkatkan kualitas data (Varma

et al., 2023). Berikut ini merupakan tahapan-tahapan yang dilakukan untuk melakukan praproses data pada penelitian ini.

3.2.1 Image Resize

Tahap pertama pada praproses data yaitu mengubah ukuran citra menjadi 224×224 piksel yang disesuaikan untuk memenuhi ukuran input yang dapat diterima oleh arsitektur DenseNet-121 (Talebi & Milanfar, 2021). Dengan dilakukan *resize* tersebut maka semua citra memiliki ukuran yang sama dan diharapkan dapat meminimalisir kemungkinan terjadi distorsi saat diproses pada lapisan konvolusi. *Image resize* ini dapat dilakukan dengan mengubah ukuran citra dari dimensi asli (W, H) ke dimensi target (W', H') menggunakan persamaan interpolasi bilinear pada Persamaan 2.3.

3.2.2 Konversi Image ke Array

Setelah ukuran citra yang dimasukkan ke dalam sistem telah sesuai dengan ukuran yang dapat diterima oleh DenseNet-121, citra dikonversi bentuknya ke format array. Dengan dilakukannya konversi ini dapat menjadi perantara dari data visual menjadi data numerik yang dapat diterima dan diproses lebih lanjut oleh DenseNet-121. Mengubah data input berupa citra menjadi representasi numerik merupakan praktik standar yang dilakukan untuk jaringan sistem saraf agar dapat melakukan beberapa tugas yang memerlukan proses pada pikselnya (Jacob et al., 2018). Contoh untuk bentuk array dari citra dengan ukuran 224×224 piksel yang memiliki format RGB (*Red, Green, Blue*) dapat dituliskan pada Persamaan 2.4.

3.2.3 Penyesuaian Channel RGB

Setiap citra memiliki *channel* yang berbeda-beda. DenseNet-121 hanya dapat memproses citra dengan *channel* RGB (*Red, Green, Blue*) yang memiliki dimensi (224, 224, 3) (Jeyabose et al., 2022). Maka citra yang *channel*-nya masih dalam format *grayscale* yang memiliki satu *channel* maupun citra RGBA (*Red, Green, Blue, Alpha*) dengan empat *channel* perlu disesuaikan terlebih dahulu agar menjadi RGB yang memiliki tiga *channel*. Jika citra hanya memiliki satu *channel* (*grayscale*) memiliki bentuk awal array seperti pada Persamaan 2.5.

Agar citra *grayscale* tersebut dapat dikonversi ke dalam RGB maka diperlukan penyalinan *channel* agar menjadi tiga. Sedangkan untuk citra RGBA yang memiliki 4 *channel* ini dapat dikonversi menjadi citra RGB dengan menghapus *channel* keempat yaitu *channel* Alpha dan hanya mengambil *channel* ke-0 (*Red*), ke-1 (*Green*), dan ke-2 (*Blue*). Konversi citra RGBA menjadi citra RGB ini dapat dirumuskan sesuai dengan Persamaan 2.6.

3.2.4 Normalisasi

Tahap selanjutnya adalah melakukan normalisasi ukuran piksel dari citra yang awalnya berkisar dari 0-255 dinormalisasi ke rentang 0-1. Proses normalisasi ini dilakukan dengan membagi setiap nilai piksel dengan 255. Normalisasi ini dilakukan agar dapat meningkatkan stabilitas komputasi selama proses ekstraksi fitur karena dengan berkurangnya rentang nilai maka gradien dalam jaringan saraf lebih terkontrol (Pramana, 2024). Sehingga konvergensi model bisa menjadi lebih cepat serta mencegah *overflow* atau *underflow* saat terdapat operasi matematika

yang dijalankan (Pramana, 2024). Persamaan 2.7 dapat digunakan untuk normalisasi nilai piksel $[0, 255]$ ke rentang $[0, 1]$.

3.2.5 Penambahan Dimensi *Batch*

Penambahan dimensi *batch* dilakukan agar citra memiliki format $(1, 224, 224, 3)$ karena DenseNet-121 ini mengharuskan input dalam bentuk *batch* tensor 4 dimensi (*batch size*, tinggi, lebar, *channel*). Sebelumnya citra telah dikonversi ke dalam array dengan ukuran $(224, 224, 3)$, maka untuk mengubah dimensi batch-nya agar dapat menjadi array dengan ukuran $(1, 224, 224, 3)$ dapat dilakukan dengan menambahkan satu dimensi baru di depan dalam penelitian ini adalah 1 yang mewakili citra tunggal. Penambahan dimensi *batch* ini dilakukan dengan Persamaan 2.8.

3.2.6 Augmentasi

Tahapan terakhir pada praproses data yaitu augmentasi data yang memiliki tujuan agar data memiliki lebih banyak variasi dan mencegah terjadinya *overfitting* (Goceri, 2023). Maka dari itu dilakukan augmentasi yang melibatkan berbagai transformasi. Seperti dilakukan rotasi dengan acak pada -20° sampai 20° agar model dapat mengenali objek berupa citra dari berbagai orientasi. Pergeseran juga dilakukan dengan menggeser maksimal 20% dari lebar citra asli serta digeser atas-bawah maksimal 20% dari citra asli. Transformasi terhadap sudut kemiringan citra juga dilakukan untuk mencontoh efek sudut pandang. Transformasi lain juga dapat dilakukan seperti dengan melakukan *zoom in* dan *zoom out* yang kemudian ruang

kosong pada berbagai hasil transformasi diisi dengan *fill mode* yang mengisi piksel kosong hasil transformasi dengan nilai piksel terdekat.

3.3 Ekstraksi Fitur oleh DenseNet-121

Pada penelitian ini DenseNet-121 digunakan sebagai ekstraktor terhadap fitur citra kerusakan bangunan pasca bencana alam dengan menghapus lapisan klasifikasi dan menambahkan GAP. Berikut ini akan dijelaskan perhitungan manual yang dilakukan pada setiap lapisan yang dimiliki oleh DenseNet-121 sebagai ekstraktor fitur untuk memberikan gambaran mengenai bagaimana pemrosesan fitur dilakukan pada setiap lapisan yang dimiliki oleh jaringan DenseNet-121.

Ekstraksi fitur pada citra kerusakan bangunan pasca bencana alam diawali dengan *input* citra kerusakan bangunan pasca bencana alam dalam format JPG/JPEG. Citra yang diterima oleh DenseNet-121 ini memiliki resolusi 224×224 piksel yang memiliki 3 *channel* warna RGB. Kemudian citra tersebut dinormalisasi agar memastikan nilai piksel dari citra *input* sudah sesuai dengan kriteria citra yang dapat diterima oleh DenseNet-121. Tahap ini menghasilkan *output* berupa tensor dengan dimensi $224 \times 224 \times 3$. Berikut ini merupakan sampel kecil dari *patch* ($3 \times 3 \times 3$) di bagian *top-left* citra yang mewakili 3 baris, 3 kolom, dan 3 *channel*:

Channel 1 (red):

-0.748	-0.748	-0.748	...	-0.748	-0.748	-0.748
-0.748	-0.748	-0.748	...	-0.748	-0.748	-0.748
-0.731	-0.731	-0.731	...	-0.731	-0.731	-0.731
...
-0.603	-0.603	-0.603	...	-0.603	-0.603	-0.603
-0.582	-0.582	-0.582	...	-0.582	-0.582	-0.582
-0.561	-0.561	-0.561	...	-0.561	-0.561	-0.561

Channel 2 (green):

0.625	0.625	0.625	...	0.625	0.625	0.625
0.625	0.625	0.625	...	0.625	0.625	0.625
0.643	0.643	0.643	...	0.643	0.643	0.643
...
0.700	0.700	0.700	...	0.700	0.700	0.700
0.705	0.705	0.705	...	0.705	0.705	0.705
0.710	0.710	0.710	...	0.710	0.710	0.710

Channel 3 (blue):

1.681	1.681	1.681	...	1.681	1.681	1.681
1.681	1.681	1.681	...	1.681	1.681	1.681
1.699	1.699	1.699	...	1.699	1.699	1.699
...
1.750	1.750	1.750	...	1.750	1.750	1.750
1.755	1.755	1.755	...	1.755	1.755	1.755
1.760	1.760	1.760	...	1.760	1.760	1.760

Perlu diketahui bahwa umumnya area *top-left* atau sudut atas kiri dari citra adalah area kosong atau *background*. Sampel *patch* yang diambil dari area tersebut hanya sebagian kecil dari seluruh citra. Nilai negatif yang mendominasi *channel 1* dapat mengindikasikan area gelap pada citra. Pada *channel 2* dan *channel 3* semuanya bernilai positif yang menunjukkan jika area pada *channel* tersebut diisi oleh warna yang netral atau terang. Kemudian selanjutnya fitur diproses oleh lapisan lanjutan yaitu lapisan konvolusi (7×7 *stride* 2). Berikut ini merupakan contoh kernel yang memiliki 1 filter:

Channel 1 (red):

$$\begin{bmatrix} 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \end{bmatrix}$$

Channel 2 (green):

$$\begin{bmatrix} 0.05 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.05 \end{bmatrix}$$

Channel 3 (blue):

$$\begin{bmatrix} -0.05 & -0.05 & -0.05 \\ -0.05 & -0.05 & -0.05 \\ -0.05 & -0.05 & -0.05 \end{bmatrix}$$

Pada konvolusi awal ini setiap *patch input* dikalikan dengan nilai kernel untuk setiap elemen. Perhitungan tersebut dilakukan terhadap ketiga *channel patch input* yang kemudian jika hasil perhitungan ketiga *channel* tersebut dijumlahkan akan dihasilkan *dot product*. Berikut ini merupakan perhitungan *dot product* terhadap ketiga *channel patch input*:

$$\text{Channel 1} = 0.1 \times (-0.7479 \times 6 \pm 0.7308 \times 3) = 0.1 \times (-6.6798) = -0.668$$

$$\text{Channel 2} = 0.05 \times (0.6254 \times 6 + 0.6429 \times 3) = 0.05 \times 5.6805 = 0.284$$

$$\text{Channel 3} = -0.05 \times (1.6814 \times 6 + 1.6988 \times 3) = -0.05 \times 15.1848 = -0.759$$

Setelah perhitungan *dot product* pada ketiga *channel patch input* dengan kernel konvolusi dilakukan, didapatkan total nilai sebesar $-0.668 + 0.284 - 0.759 = -1.143$ sebelum aktivasi menggunakan ReLU. Selanjutnya akan diterapkan fungsi aktivasi ReLU untuk mengubah *dot product* yang bernilai negatif menjadi nol, jika *dot product* bernilai positif maka ReLU akan mempertahankan nilai tersebut. Karena hasil perhitungan *dot product* adalah -1.143 maka ReLU mengubahnya menjadi $\max(0, -1.143) = 0$.

Patch output yang dihasilkan setelah ReLU ini berisi nilai nol yang kemudian diteruskan menuju lapisan berikutnya. Berikut ini merupakan bentuk *patch output* yang dihasilkan oleh fungsi aktivasi ReLU:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Patch tersebut diproses pada lapisan selanjutnya yaitu lapisan lanjutan setelah lapisan konvolusi yang terdiri dari *maxpooling*, *dense block 1*, *transition layer 1*, *dense block 2*, *transition layer 2*, *dense block 3*, *transition layer 3*, *dense block 4*, hingga GAP (*Global Average Pooling*). Tetapi fungsi aktivasi ReLU yang sebelumnya mengubah nilai negatif menjadi nol mengakibatkan seluruh lapisan lanjutan memproses *input* nol tersebut secara berurutan tanpa menghasilkan aktivasi baru.

Transition layer 1 berisi konvolusi 1×1 yang bertujuan untuk melakukan kompresi terhadap ukuran *feature map*. Namun karena seluruh *input* dari lapisan konvolusi awal ke lapisan ini adalah matriks yang seluruhnya bernilai nol maka berapapun besar nilai konvolusi pada lapisan ini akan tetap menghasilkan *output* nol dan nilai rata-rata *pooling* juga nol.

Lapisan selanjutnya adalah *dense block 2* yang terdiri dari 12 lapisan *bottleneck*. Pada setiap lapisan *bottleneck* memiliki 1×1 konvolusi dan 3×3 untuk mengekstrak pola yang diikuti oleh penggabungan *channel* sebelumnya atau *concatenation*. *Input* pada lapisan ini berupa matriks yang nilainya nol sehingga hasilnya juga nol pada konvolusi 1×1 dan konvolusi 3×3 . Lapisan ini juga tidak membentuk fitur baru berdasarkan nilai *input*.

Pada lapisan selanjutnya yaitu *transition layer 2* yang terdiri dari 1×1 konvolusi dan *average pooling* juga memproses nilai nol yang hasilnya juga tetap nol. Nilai ini diproses lebih lanjut oleh lapisan *dense block 3* yang lapisan *bottleneck*-nya lebih besar dari *dense block* sebelumnya yaitu sebesar 24 lapisan *bottleneck*. Tetapi karena nilai *input*-nya adalah nol maka setiap until lapisan *bottleneck* dalam *dense block 3* ini hanya memproses dan menggabungkan nilai nol.

Pola yang sama terulang kembali pada lapisan selanjutnya yaitu *transition layer 3* yang konvolusi 1×1 serta *average pooling* di dalamnya tetap menghasilkan nol. Hasil tersebut kemudian diproses oleh *dense block 4* yang memiliki 16 lapisan *bottleneck* tetapi lagi-lagi menghasilkan *output* nol meskipun seharusnya blok ini yang dapat menangkap pola kompleks pada citra. Karena tidak ada *input* yang nilainya aktif maka blok ini tidak mengekstrak informasi apapun.

Lapisan paling akhir dari arsitektur DenseNet-121 sebagai ekstraktor fitur adalah GAP (*Global Average Pooling*). Lapisan ini menghitung rata-rata dari setiap *channel* tetapi karena *feature map* yang diterima bernilai nol maka vektor fitur akhir yang dihasilkan menjadi $[0, 0, \dots, 0]$ yang berdimensi 1024. Hasil tersebut dapat menunjukkan jika *patch input* dari *top-left* citra tidak dianggap memiliki informasi penting karena pada umumnya area tersebut pada citra merupakan area kosong atau bisa juga *background*. Tetapi DenseNet-121 sebagai salah satu model dari CNN bekerja secara keseluruhan pada citra tidak hanya pada area tertentu dari citra, sehingga *patch* dari area lain di citra yang sama yang mengandung fitur relevan tetap menghasilkan nilai aktivasi baru dan fitur yang penting.

3.4 Skenario Pengujian

Tahap pengujian merupakan rangkaian yang dilakukan untuk mengevaluasi representasi numerik berupa vektor fitur yang dihasilkan oleh DenseNet-121 dari ekstraksi fitur citra kerusakan bangunan pasca bencana alam. Melalui pengujian ini diharapkan dapat mengetahui pengaruh dari augmentasi data terhadap vektor fitur yang dihasilkan.

Total data yang diproses sistem yang dikembangkan pada penelitian ini berjumlah 50 citra kerusakan bangunan pasca bencana alam. Setiap citra diproses dengan cara yang sama tanpa melalui proses yang berbeda. Pengujian ini dilakukan dengan dua pendekatan. Yang pertama adalah pendekatan kuantitatif menggunakan *cosine similarity* untuk mengukur tingkat kemiripan antara vektor fitur yang dihasilkan oleh citra asli dengan vektor fitur dari citra augmentasi. Kedua, pendekatan kualitatif dilakukan dengan visualisasi distribusi fitur melalui PCA (*Principal Component Analysis*) berdimensi rendah (2D dan 3D). Melalui kedua pendekatan tersebut dapat dianalisis apakah citra tetap mempertahankan karakteristik yang relevan berdasarkan DenseNet-121 yang digunakan untuk ekstraksi fitur pada penelitian ini.

Pengujian dimulai dengan melakukan input data yang kemudian melalui praproses data seperti yang telah ditetapkan sebelumnya. Setelah itu citra melalui proses augmentasi data yang akan menghasilkan beberapa citra hasil augmentasi. Citra *original* dan seluruh citra hasil augmentasi kemudian diekstrak fiturnya oleh DenseNet-121. Dari vektor fitur yang dihasilkan oleh citra augmentasi dilakukan pengukuran kemiripan dengan vektor fitur yang dihasilkan oleh citra *original*.

Pengukuran menggunakan *cosine similarity* diperlukan agar dapat mengetahui seberapa besar kemiripan dua vektor dalam ruang berdimensi tinggi. *Cosine similarity* dapat dihitung menggunakan Persamaan 2.10.

Dari hasil perhitungan menggunakan *cosine similarity* ini jika hasilnya semakin mendekati 1 antara vektor fitur citra augmentasi dengan citra *original* karena menunjukkan jika vektor satu arah dan sejajar. Nilai *cosine similarity* yang mendekati 1 ini menunjukkan jika hasil ekstraksi fitur oleh DenseNet-121 konsisten walaupun sudah dilakukan berbagai transformasi visual saat augmentasi data. Jika nilai *cosine similarity* adalah 0 diartikan jika tidak ada kemiripan. Sedangkan jika nilainya semakin mendekati -1, maka hal ini menunjukkan bahwa vektor berlawanan arah.

Dilakukan juga pengujian melalui pendekatan kualitatif pada pengujian ini dengan melakukan visualisasi terhadap distribusi fitur menggunakan PCA yang memiliki dimensi rendah. PCA dapat digunakan untuk mengubah data kompleks atau data yang memiliki dimensi tinggi menjadi lebih rendah, namun tetap mempertahankan informasi penting di dalamnya. PCA bekerja dengan cara yang mirip dengan meringkas sebuah fitur rumit ke dalam titik-titik pada sumbu X, Y, dan Z.

Visualisasi ini dilakukan agar mengetahui apakah persebaran fitur dari citra augmentasi memiliki pola yang mirip dengan citra *original* dan menganalisis hubungan antar fitur. Hasil ekstraksi fitur oleh DenseNet-121 akan direduksi terlebih dahulu karena dimensinya terlalu tinggi untuk PCA. PCA bekerja dengan mencari arah dari data fitur dengan tetap mempertahankan sebagian besar informasi

penting. Hasil dari reduksi dimensi tersebut kemudian divisualisasikan dalam bentuk grafik *scatter plot* 2D dan 3D. Visualisasi PCA dapat dilakukan dengan persamaan 2.11.

Dengan menerapkan kombinasi evaluasi melalui pendekatan secara kuantitatif dan kualitatif, pengukuran terhadap hasil ekstraksi fitur oleh DenseNet-121 tidak hanya diukur kemiripannya secara matematis tetapi juga memberikan gambaran mengenai distribusi fitur yang dihasilkan melalui proyeksi visual. Pengujian ini penting untuk dilakukan agar dapat menilai kemampuan DenseNet-121 dalam mempertahankan karakteristik penting dari berbagai citra kerusakan bangunan pasca bencana alam bahkan setelah citra tersebut mengalami berbagai transformasi.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Uji Coba Sistem

Penelitian ini berhasil mengimplementasikan DenseNet-121 untuk melakukan ekstraksi fitur pada citra kerusakan bangunan pasca bencana alam dengan sistem yang dikembangkan. Sistem yang dikembangkan ini dibangun khusus untuk mendapatkan fitur dari citra bangunan yang mengalami kerusakan. Implementasi pada sistem menggunakan bahasa pemrograman *Python* yang dijalankan pada Google Colab untuk eksekusi kode yang berbasis *cloud*. Sistem ini memanfaatkan beberapa *library* yang dapat digunakan seperti TensorFlow, Keras, Matplotlib, Scikit-Learn, Pandas, PIL (*Python Imaging Library*), dan NumPy.

Sistem yang telah dibangun melalui proses pengujian dengan menggunakan dataset yang telah disiapkan. Dataset yang mencakup berbagai kerusakan bangunan tersebut harus melalui tahapan praproses data (*data preprocessing*) untuk memastikan bahwa data yang digunakan ini berkualitas dan memiliki memiliki standar yang seragam. Proses pengujian sistem ini dilakukan untuk mengetahui kemampuan sistem yang dikembangkan untuk melakukan ekstraksi fitur menggunakan DenseNet-121 yang dimodifikasi untuk menghasilkan representasi numerik yang kemudian dievaluasi berdasarkan *output* yang dihasilkan untuk mengekstrak fitur dari input *image*.

4.2 Hasil Uji Coba

4.2.1 Persiapan Dataset

Penelitian ini menggunakan dataset “*Damaged Constructions Image Dataset*” yang bersifat *open source* yang bersumber dari situs Kaggle. Data yang digunakan pada penelitian ini sejumlah 50 citra dari kerusakan bangunan yang terdampak bencana alam dan semua diproses dengan cara yang sama pada sistem yang dikembangkan. Dataset tersebut dipindahkan ke Google Drive dalam format .zip untuk selanjutnya melalui proses *mounting* ke direktori lokal di Google Colab.

4.2.2 Praproses Data

Seluruh data yang akan diproses oleh sistem untuk ekstraksi fitur perlu melalui tahapan fundamental dari penelitian ini yaitu praproses data. Tahap ini dilakukan dengan cara mengubah ukuran citra, konversi ke array, penyesuaian *channel* RGB, normalisasi nilai piksel, dan augmentasi . Tahap ini dilakukan hanya pada satu citra yang digunakan sebagai input pada pada sistem yang dikembangkan. Citra yang digunakan sebagai input tersebut akan dikenali oleh sistem dengan memanfaatkan *library* PIL (*Python Imaging Library*) atau *pillow* menggunakan fungsi *Image.open(img_path)* yang kemudian disimpan sebagai objek PIL ke dalam memori. Praproses data yang dilakukan penelitian ini terbagi melalui tahap berikut:

1. *Image Resize*

Arsitektur DenseNet-121 yang digunakan pada penelitian ini hanya dapat memproses citra dengan ukuran 224×224 piksel. Sedangkan tidak semua citra yang dimuat pada dataset memiliki piksel yang sesuai dengan ukuran yang dapat diproses

DenseNet-121. Maka perlu dilakukan *image resize* dengan cara mengubah ukuran dimensi dari seluruh citra yang digunakan pada penelitian ini ke 224×224 piksel menggunakan *resize()*. Berikut ini merupakan *pseudocode* untuk *image resize*.

```
FUNCTION resize_image(image) :
    Resize image to (224, 224)
    RETURN resized_image
```

Fungsi untuk melakukan *image resize* pada *pseudocode* tersebut didefinisikan dalam fungsi untuk melakukan praproses data. Berikut ini merupakan perbedaan yang menunjukkan antara citra asli yang digunakan sebagai input dengan hasil dari *image resize*:



Gambar 4. 1 Citra asli dan citra hasil *image resize*

2. Konversi ke Array dan Penyesuaian Channel RGB

Citra yang telah direduksi ukuran pikselnya kemudian dikonversi ke *array* NumPy menggunakan fungsi *np.array()*. Tahap ini perlu dilakukan karena model *deep learning* dapat melakukan operasi matematika dalam bentuk numerik seperti *array*, bukan dalam bentuk objek gambar.

Selanjutnya diperlukan penanganan *channel* citra agar dapat dipastikan bahwa citra yang digunakan sebagai input ini dalam format RGB yang memiliki 3 *channel* sesuai dengan jumlah yang diharapkan oleh DenseNet-121. Jika citra yang digunakan sebagai input ternyata dalam format *grayscale* yang hanya memiliki 1 *channel*, maka fungsi *np.stack()* digunakan untuk menduplikat *channel grayscale* menjadi 3. Jika citra yang diinputkan memiliki format RGBA yang memiliki 4 *channel*, maka *channel A (alpha)* pada RGBA akan dihapus sehingga citra memiliki format RGB. Berikut ini merupakan *pseudocode* untuk tahap konversi citra ke *array* dan penyesuaian *channel*:

```

FUNCTION convert_image_to_array(resized_image):
    Convert resized_image to numpy array (img_array)

    IF image is grayscale (2D array):
        Convert to 3-channel RGB by stacking

    IF image has 4 channels (RGBA):
        Discard alpha channel and retain RGB

    RETURN img_array

```

3. Normalisasi dan Penambahan Dimensi *Batch*

Normalisasi pada piksel untuk memudahkan pemrosesan citra pada DenseNet-121 dengan mengurangi variasi yang tidak diperlukan dalam data. Seluruh citra pada dataset awalnya memiliki nilai antara 0 (hitam) sampai 255 (putih). Inti dari tahap normalisasi ini adalah mengubah rentang nilai pada citra menjadi 1 sampai 0 dengan cara membagi setiap nilai piksel dengan 255.

Setelah seluruh citra pada dataset dinormalisasi maka nilai pikselnya akan seragam sehingga model dapat mempelajari pola kerusakan bangunan pada dataset dengan lebih baik. Proses ini juga dapat mempercepat konvergensi model selama proses pelatihan data. Normalisasi dilakukan dengan memanfaatkan fungsi *preprocess_input()* dari TensorFlow/Keras.

Setelah normalisasi berhasil dilakukan, fungsi *np.expand_dims()* ditambahkan ke dalam *array* untuk menambah dimensi dari *batch* dengan mengubah bentuk *array* dari (224, 224, 3) menjadi (1, 224, 224, 3). Hal ini dilakukan sesuai dengan *pseudocode* untuk normalisasi dan menambah dimensi *batch* seperti berikut:

```

FUNCTION normalize_image(img_array):
    Apply DenseNet121-specific normalization using
    preprocess_input
    RETURN normalized_image

FUNCTION add_batch_dimension(normalized_image):
    Expand image shape to (1, height, width, channels)
    RETURN batch_image

```

4. Augmentasi Data

Pada penelitian ini augmentasi data yang dilakukan dengan memanfaatkan fungsi *ImageDataGenerator* dari *library* Keras. Augmentasi ini bertujuan untuk meningkatkan keragaman data melalui berbagai transformasi pada citra seperti rotasi acak hingga 20°, menggeser gambar secara acak (vertikal dan horizontal) 20% dari lebar dan tinggi asli, transformasi *shear*, *zoom* secara acak, dan *flip* horizontal.



Gambar 4. 2 Augmentasi data

Setiap citra yang melalui augmentasi ini menghasilkan 5 citra. Hal tersebut terjadi karena *ImageDataGenerator* menerapkan kombinasi acak dari berbagai transformasi yang telah ditetapkan. Sehingga dari augmentasi data yang dilakukan pada penelitian ini bukan berarti menghasilkan 1 citra augmentasi dari setiap jenis transformasi yang sudah ditetapkan. Dari augmentasi data ini dihasilkan area piksel kosong yang kemudian diisi oleh metode *nearest*. Gambar 4.3 merupakan gambar yang memuat hasil dari augmentasi data yang dilakukan dengan *pseudocode* berikut ini untuk augmentasi data.

```

FUNCTION augment_image_with_visualization(img_array)
  TRY:
    // Konfigurasi augmentasi
    CREATE ImageDataGenerator dengan parameter:
      - rotation_range = 20
  
```

```

- width_shift_range = 0.2
- height_shift_range = 0.2
- shear_range = 0.2
- zoom_range = 0.2
- horizontal_flip = True
- fill_mode = 'nearest'

INISIALISASI augmented_images = []

// Tampilkan gambar asli
TAMPILKAN gambar asli pada subplot

// Generate 5 gambar augmented
FOR i = 1 TO 5:
    GENERATE gambar augmented dengan datagen.flow()
    TAMBAHKAN ke augmented_images
    TAMPILKAN gambar augmented pada subplot
END FOR

RETURN array augmented_images
CATCH Exception:
    RETURN array kosong
END FUNCTION

```

ImageDataGenerator digunakan untuk menetapkan parameter dari berbagai transformasi yang dilakukan. *ImageDataGenerator* akan menghasilkan transformasi baru yang acak setiap dijalankan. Citra hasil augmentasi dihasilkan dengan melakukan iterasi augmentasi sejumlah lima kali dan setiap kali *next()* dipanggil maka sebuah citra baru akan dihasilkan dengan kombinasi transformasi acak. Seluruh hasil augmentasi ini disimpan dalam *augmented_images* lalu ditampilkan oleh Matplotlib bersama dengan citra *original*. Array NumPy yang

berisi citra augmentasi selanjutnya akan digunakan untuk ekstraksi fitur menggunakan DenseNet-121.

4.2.3 Ekstraksi Fitur oleh DenseNet-121

Ekstraksi fitur yang dilakukan pada penelitian ini mengimplementasikan arsitektur DenseNet-121. Tetapi penting untuk digaris bawahi bahwa pada penelitian ini fokus untuk menjalankan tugas ekstraksi fitur saja. Sehingga perlu sedikit modifikasi pada arsitektur DenseNet-121 dengan menghapus lapisan *classifier* yaitu lapisan FC (*Fully Connected*). Kemudian untuk menggantikan lapisan yang dihapus, ditambahkan lapisan GAP (*Global Average Pooling*) yang dapat mereduksi dimensi spasial dari lapisan *output* menjadi vektor. Proses ekstraksi fitur dari satu citra yang diinputkan pada sistem akan menghasilkan representasi numerik. *Pseudocode* berikut ini dapat menggambarkan proses ekstraksi fitur yang dilakukan pada penelitian ini:

```

FUNCTION extract_features_with_visualization(img_array)
  TRY:
    // Load model DenseNet121 pre-trained
    LOAD DenseNet121 dengan:
      - weights = 'imagenet'
      - include_top = False
      - pooling = 'avg'
    // Ekstraksi fitur
    features = PREDIKSI model pada img_array
    // Visualisasi fitur
    CREATE subplot untuk visualisasi:
      1. Heatmap vektor fitur
      2. Distribusi nilai fitur (histogram)
    // Tampilkan statistik fitur
    CALCULATE dan SHOW:
  
```

```

        - Shape fitur
        - Mean, Std Dev, Min, Max
    RETURN features
    CATCH Exception:
        RETURN None
END FUNGSI

// Fungsi penyimpanan feature
FUNCTION save_features(features, save_path, filename)
    TRY:
        IF direktori save_path tidak ada THEN
            BUAT direktori save_path
        END IF
        // Simpan sebagai .npy
        npy_path = GABUNG save_path dan filename + '.npy'
        SAVE features sebagai numpy file
        // Simpan sebagai .csv
        csv_path = GABUNG save_path dan filename + '.csv'
        CONVERT features ke DataFrame
        SAVE DataFrame sebagai CSV
        RETURN True
    CATCH Exception:
        RETURN False
END FUNCTION

```

Proses ekstraksi fitur dimulai dengan *load* model *pretrained* DenseNet-121. Terdapat tiga parameter penting saat *load* model yaitu *weights='imagenet'*, *include_top=False*, dan *pooling='avg'*. Dengan *include_top=False*, lapisan paling atas yang biasanya digunakan untuk klasifikasi dihapus karena tidak diperlukan untuk tugas ekstraksi fitur. Agar model dapat menggabungkan hasil dari lapisan konvolusional terakhir digunakan *pooling='avg'* yang menghitung rata-rata (*mean*) dari setiap *feature map*, sehingga kita mendapatkan satu nilai dari masing-masing

peta. Hasil akhirnya adalah vektor satu dimensi yang lebih ringkas dan tidak terlalu sensitif terhadap posisi objek dalam gambar.

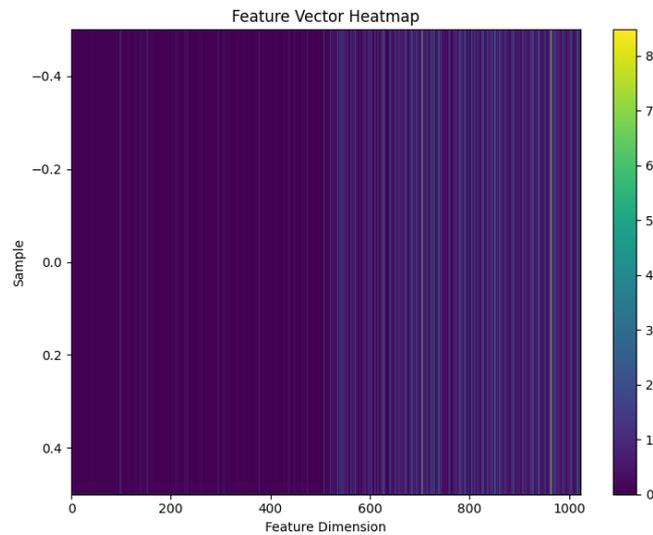
Untuk memberikan pemahaman lebih lanjut, berikut ini merupakan Tabel 4.1 yang berisi rincian dari *output shape* dari setiap lapisan DenseNet-121 untuk proses ekstraksi fitur pada citra kerusakan bangunan pasca bencana alam yang sudah dimodifikasi dengan menghapus lapisan klasifikasi. Tabel 4.1 juga menunjukkan urutan dari lapisan-lapisan yang dilalui oleh *input* citra dalam arsitektur DenseNet-121 yang telah dimodifikasi. *Output* akhir berupa vektor fitur 1×1024 yang dihasilkan oleh GAP.

Tabel 4. 1 Lapisan ekstraksi fitur oleh DenseNet-121

Tahap	Lapisan	Output Shape
<i>Input</i>	-	224×224×3
<i>Initial Conv</i>	7×7 Conv + BN + ReLU	112×112×64
<i>Max Pooling</i>	3×3, stride 2	56×56×64
<i>Dense Block 1</i>	6 unit	56×56×256
<i>Transition Layer 1</i>	1×1 Conv + AvgPool	28×28×128
<i>Dense Block 2</i>	12 unit	28×28×512
<i>Transition Layer 2</i>	1×1 Conv + AvgPool	14×14×256
<i>Dense Block 3</i>	24 unit	14×14×1024
<i>Transition Layer 3</i>	1×1 Conv + AvgPool	7×7×512
<i>Dense Block 4</i>	16 unit	7×7×1024
<i>Global Average Pooling</i>	<i>Pool over 7×7</i>	1×1024

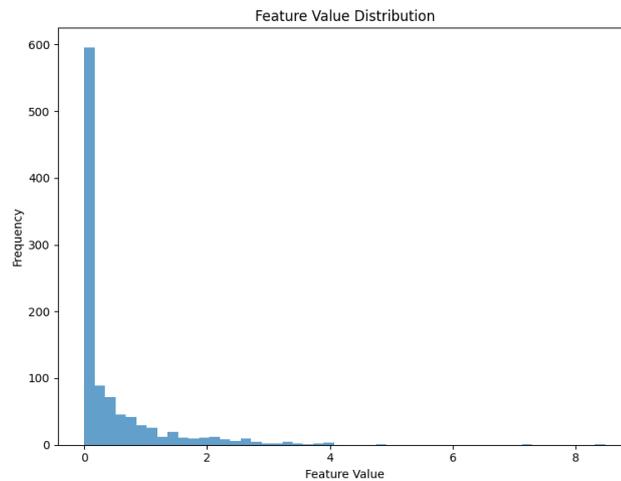
Citra diinputkan ke dalam model dan akan melewati berbagai lapisan seperti lapisan konvolusi, aktivasi (ReLU), *pooling*, hingga normalisasi untuk mengekstrak fitur. Pada lapisan awal cenderung menangkap ciri-ciri dasar seperti tepi atau warna, semakin dalam lapisan maka model mulai mengenali pola-pola yang lebih kompleks dan spesifik terhadap bentuk atau objek. *Output* akhir dari proses ekstraksi fitur oleh DenseNet-121 adalah representasi numerik berupa vektor fitur berdimensi (1, 1024) untuk setiap citra. Angka 1024 ini mengindikasikan jumlah filter yang aktif di lapisan konvolusional terakhir setelah dilakukan *average*

pooling. Vektor fitur ini bisa dianggap sebagai "ringkasan" dari informasi penting dalam citra yang diinputkan.



Gambar 4.3 *Feature Vector Heatmap*

Gambar 4.3 dan Gambar 4.4 merupakan visualisasi dari ekstraksi fitur pada citra asli yang menjadi input dalam sistem yang menghasilkan vektor berdimensi (1, 1024). Warna pada *heatmap* merepresentasikan intensitas nilai fitur dari yang rendah (ungu) ke tinggi (kuning). *Heatmap* pada Gambar 4.3 menunjukkan hanya sebagian dimensi fitur yang menunjukkan nilai yang sangat rendah karena sebagian besar didominasi warna ungu dan gradasi warna hijau sampai kuning menjadi minoritas. Hal tersebut dapat mengindikasikan bahwa model selektif dalam memproses fitur penting dan mengabaikan sisanya.



Gambar 4. 4 *Feature Value Distribution*

Histogram pada Gambar 4.4 yang menampilkan distribusi nilai fitur menunjukkan bahwa sebagian besar nilai terletak pada kisaran yang rendah dan hanya sebagian kecil yang mencapai nilai tinggi. Karena dari total 1024 fitur, lebih dari 600 fitur memiliki nilai yang mendekati angka 0 yang mewakili fitur tidak aktif. Hanya sebagian kecil fitur penting dengan nilai tinggi. Hal ini menunjukkan jika mayoritas fitur dianggap tidak terlalu berkontribusi.

Tahapan ekstraksi untuk citra-citra yang dihasilkan dari proses augmentasi dilakukan sama persis seperti pada citra asli. Setiap versi citra augmentasi akan menghasilkan vektor fitur tersendiri, dan semua vektor ini digabungkan menggunakan fungsi *np.vstack()* menjadi satu matriks berdimensi (n, 1024), di mana n adalah jumlah seluruh gambar, baik asli maupun yang diaugmentasi. Melalui ini dapat dipelajari bagaimana fitur dapat berubah karena variasi gambar.

Tabel 4. 2 Statistik fitur

No	Citra	Mean	Standar Deviasi	Min	Max
1	Original	0.4466	0.8038	0.0000	8.4816
2	Augmentasi 1	0.4454	0.7512	0.0000	6.6514
3	Augmentasi 2	0.4208	0.7650	0.0000	7.2909
4	Augmentasi 3	0.4658	0.8172	0.0000	8.8094

No	Citra	Mean	Standar Deviasi	Min	Max
5	Augmentasi 4	0.4322	0.7455	0.0000	5.1155
6	Augmentasi 5	0.4313	0.7425	0.0000	5.4469

Tabel 4.2 menunjukkan data statistik dari sampel citra *original* dan juga citra hasil augmentasi. Nilai rata-rata berada pada kisaran 0.42 sampai 0.46 ketika citra *original* nilainya 0.4466. Hal tersebut menunjukkan bahwa tidak terdapat perubahan besar dari keseluruhan fitur. Nilai standar deviasi memiliki kisaran 0.74 hingga 0.81, perbedaan ini menunjukkan bahwa aktivasi fitur karena augmentasi cukup ragam tetapi dalam rentang yang wajar. Sedangkan untuk nilai minimum semua citra memiliki nilai yang sama. Nilai maksimum pada citra augmentasi 3 merupakan yang paling tinggi bahkan lebih tinggi dibandingkan citra *original* itu sendiri. Hal tersebut menunjukkan bahwa terdapat aktivasi baru pada fitur karena augmentasi namun tetap mempertahankan informasi penting.

```

      f1      f2      f3      f4      f5      f6      f7 \
0 0.000525 0.005306 0.000811 0.001790 0.090910 0.326968 0.000152
1 0.000403 0.002862 0.001458 0.000984 0.139915 0.416453 0.000049
2 0.000532 0.002753 0.000584 0.001062 0.083683 0.290955 0.000088
3 0.000381 0.003432 0.000726 0.001416 0.108345 0.337179 0.000235
4 0.000348 0.002790 0.001237 0.001551 0.063333 0.337840 0.000132
5 0.000521 0.002624 0.001604 0.001546 0.102106 0.274441 0.000051

      f8      f9      f10 ...      f1015      f1016      f1017      f1018 \
0 0.000973 0.061198 0.000235 ... 1.800221 0.003588 4.831315 1.081568
1 0.001292 0.026282 0.000236 ... 2.709798 0.071232 3.124098 1.809974
2 0.001601 0.033236 0.000217 ... 0.918367 0.172063 5.026719 1.369896
3 0.000680 0.053551 0.000312 ... 4.097234 0.003275 4.885095 1.590819
4 0.000529 0.076290 0.000297 ... 3.510531 0.038926 4.120932 1.073446
5 0.001295 0.048946 0.000360 ... 1.819502 0.095754 4.217321 1.186378

      f1019      f1020      f1021      f1022      f1023      f1024
0 1.040476 0.295120 0.113976 1.162595 4.055098 0.441669
1 0.694427 0.505413 0.545111 0.975273 2.238926 0.284613
2 0.653743 1.137931 0.343436 1.511672 2.221571 0.238607
3 0.891153 0.454728 0.317441 0.677512 5.738722 0.464519
4 1.059730 0.405338 0.341969 0.516457 4.481302 0.524043
5 0.810966 0.763701 0.502281 1.301674 1.436771 0.306094

[6 rows x 1024 columns]

```

Gambar 4. 5 Sampel vektor fitur

Gambar 4.5 menampilkan sampel dari vektor fitur citra asli (1 citra) pada baris ke-0 dan citra hasil augmentasi (5 citra) pada baris ke-1 sampai ke-5.

Sedangkan kolom f_1 sampai f_{1024} menunjukkan satu dimensi fitur. Nilai-nilai pada vektor fitur tersebut menunjukkan jika citra yang diaugmentasi dapat menghasilkan pola fitur yang bervariasi dalam rentang yang serupa. Tetapi konsistensi sebagian fitur antara citra asli dengan citra hasil augmentasi menunjukkan bahwa model mempertahankan pemahaman dasar terhadap konten visual utama. Hasil ekstraksi fitur dalam representasi numerik ini kemudian disimpan dalam format .csv dan .npy.

4.2.4 Evaluasi Vektor Fitur

Setelah ekstraksi fitur berhasil dilakukan perlu diketahui bagaimana cara pengukuran terhadap representasi numerik berupa vektor fitur dapat dilakukan. Pengukuran ini dilakukan dengan melakukan evaluasi terhadap vektor fitur melalui pendekatan kuantitatif serta pendekatan kualitatif.

Pertama, evaluasi kuantitatif dilakukan dengan menghitung kemiripan dari citra *original* dengan citra augmentasi. Dalam penelitian ini *cosine similarity* digunakan untuk mengukur kesamaan antara citra *original* dan citra augmentasi. *Cosine similarity* dapat dihitung secara matematis dengan menggunakan persamaan 3.7. Berikut ini merupakan *pseudocode* untuk melakukan perhitungan *cosine similarity* pada penelitian ini:

<p>INPUT:</p> <ul style="list-style-type: none"> - <i>original_features</i>: feature vector [1, 1024] - <i>augmented_features_list</i>: list of feature vectors
<p>OUTPUT:</p> <ul style="list-style-type: none"> - <i>similarity_scores</i>: array of similarity values
<p>BEGIN</p>

```

// Import:      from sklearn.metrics.pairwise import
cosine_similarity

original_features_flat ← RESHAPE(original_features, 1, -1)
similarity_scores ← EMPTY_LIST()

FOR each aug_features IN augmented_features_list DO
    aug_feat_flat ← RESHAPE(aug_features, 1, -1)

    // Using sklearn's optimized cosine_similarity function
    // This function internally applies the formula:
    //  $\cos(\theta) = (A \cdot B) / (||A|| \times ||B||)$ 

    // Returns a 2D similarity matrix
    similarity_matrix ←
COSINE_SIMILARITY(original_features_flat, aug_feat_flat)

    // Extract scalar value from similarity matrix [0][0]
    similarity_value ← similarity_matrix[0][0]

    similarity_scores.APPEND(similarity_value)
    PRINT("Similarity:", similarity_value)
END FOR

RETURN similarity_scores
END

```

Implementasi perhitungan *cosine similarity* pada penelitian ini dilakukan dengan fungsi *cosine_similarity()* dari *library* Scikit-Learn. Proses perhitungan dimulai dengan melakukan *reshape original_features* yang memiliki ukuran [1, 1024] agar formatnya *flat*. Setelah itu dilakukan iterasi untuk setiap elemen yang ada di dalam *augmented_features_list*. Fokus utama dari *pseudocode* di atas adalah memanggil fungsi *built-in* yang mengimplementasikan rumus pada persamaan 3.7.

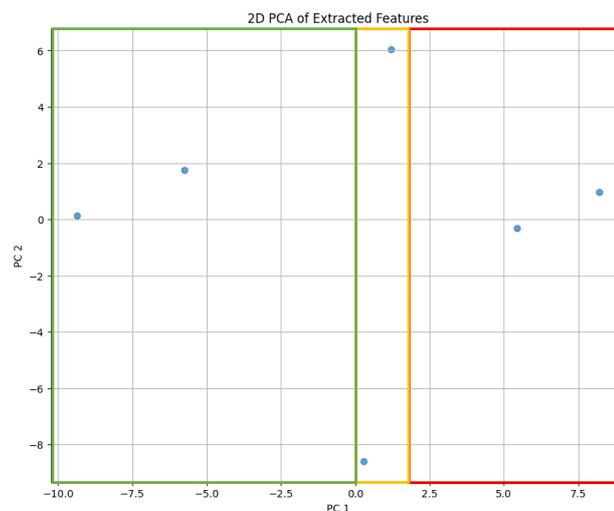
Berikut ini merupakan hasil dari perhitungan *cosine similarity* yang dilakukan oleh *cosine_similarity()*:

Tabel 4. 3 Hasil perhitungan *cosine similarity*

No	Kombinasi Citra		<i>Cosine Similarity</i>
1	<p style="text-align: center;">Original</p> 	<p style="text-align: center;">Augmented 1</p> 	0.8658
2	<p style="text-align: center;">Original</p> 	<p style="text-align: center;">Augmented 2</p> 	0.9081
3	<p style="text-align: center;">Original</p> 	<p style="text-align: center;">Augmented 3</p> 	0.8850
4	<p style="text-align: center;">Original</p> 	<p style="text-align: center;">Augmented 4</p> 	0.8898
5	<p style="text-align: center;">Original</p> 	<p style="text-align: center;">Augmented 5</p> 	0.9027

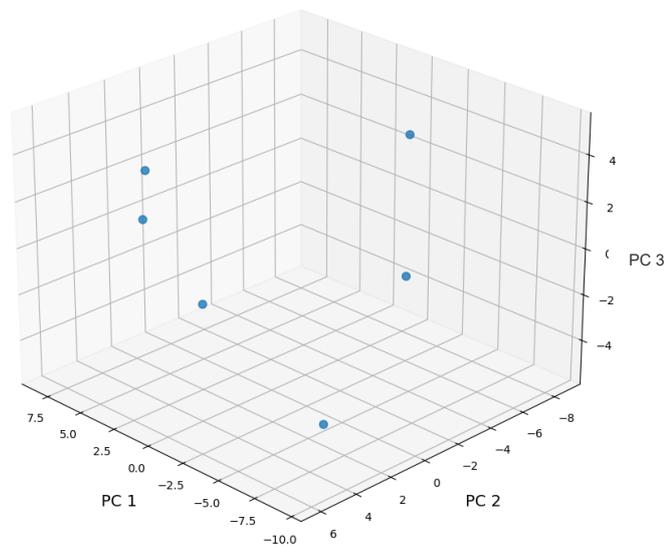
Nilai *cosine similarity* yang dihasilkan pada seluruh kombinasi citra *original* dengan masing-masing citra hasil augmentasi berada di atas 0.86. Nilai tertinggi dihasilkan dari kombinasi citra *original* dan citra augmentasi 2 yaitu sebesar 0.9081. Sedangkan nilai *cosine similarity* terendah diperoleh dari kombinasi citra *original* dan citra augmentasi 1 sebesar 0.8658 yang masih tergolong tinggi. Dari hasil perhitungan nilai *cosine similarity* ini terbukti bahwa vektor fitur yang dihasilkan dari berbagai transformasi saat augmentasi tetap mirip dengan citra *original*.

Selanjutnya akan dilakukan visualisasi menggunakan PCA agar dapat memberikan gambaran visual terhadap persebaran fitur yang di ekstraksi untuk dievaluasi lebih lanjut. Fitur yang dimuat pada visualisasi ini berasal dari satu citra *original* dan seluruh citra augmentasi yang telah diekstraksi menggunakan model DenseNet-121 yang sudah dimodifikasi. Berikut ini merupakan hasil visualisasi 2D dan 3D menggunakan PCA untuk menganalisis pola kerusakan bangunan pada citra.



Gambar 4. 6 2D PCA of extracted features

Dari visualisasi melalui PCA 2D dapat diperoleh enam titik yang tersebar dengan tidak secara acak di seluruh ruang. Dari *scatter plot* tersebut tidak terlihat jika terdapat titik yang terpisah jauh dengan titik lain secara ekstrem. Hal ini menunjukkan bahwa augmentasi yang dilakukan memang menambah keragaman data namun tetap dengan mempertahankan karakteristik utama citra dalam fituranya.



S

Gambar 4. 7 3D PCA of extracted features

Visualisasi 3D menggunakan PCA dibuat dengan tujuan melakukan evaluasi lanjutan terhadap distribusi fitur pada ruang vektor dengan dimensi yang lebih tinggi. Gambar 4.7 menunjukkan PCA 3D yang memiliki tiga sumbu yaitu sumbu X, Y, dan Z. Ketiga sumbu tersebut mewakili tiga komponen utama pada visualisasi PCA 3D yaitu PC 1 (*Principal Component 1*) pada sumbu X, PC 2 (*Principal Component 2*) pada sumbu Y, dan PC 3 (*Principal Component 3*) pada sumbu Z.

Pada dasarnya ketiga komponen yang terdapat pada gambar 4.7 mewakili arah utama dari distribusi fitur berdasarkan variasi yang paling signifikan. PC 1 memuat fitur yang variasinya paling besar karena fitur-fitur penting dari citra original dan citra augmentasi paling banyak diproyeksikan ke arah ini. PC 2 merupakan sumbu yang menunjukkan variasi terbesar kedua setelah PC 1. PC 2 dapat memberikan perspektif tambahan terhadap perbedaan antar fitur karena dapat menangkap informasi yang tidak tertangkap oleh PC 1. Terdapat sumbu yang berperan melengkapi PC 1 dan PC 2 yaitu PC 3 yang menyimpan informasi tambahan yang tidak dijelaskan PC 1 dan PC 2.

Titik-titik yang tersebar tidak ada yang berjauhan secara ekstrem dan relatif dekat. Dengan jarak distribusi fitur tersebut dapat menjadi indikasi bahwa memang terdapat variasi kecil dari augmentasi tetapi tidak mengubah fitur yang direpresentasikan. Hasil ini cukup untuk membuktikan bahwa ekstraksi fitur menggunakan DenseNet-121 fiturnya terjaga secara konsisten.

Dari dua pendekatan yang dilakukan untuk mengukur hasil ekstraksi fitur oleh DenseNet-121 dapat diketahui bahwa vektor fitur dari citra augmentasi sangat mirip dengan citra *original* baik dari perhitungan matematis dan jika dilihat secara visual. Augmentasi yang dilakukan berhasil mempertahankan informasi penting dari citra. DenseNet-121 terbukti dapat mengekstrak fitur dari citra kerusakan bangunan pasca bencana alam dengan konsisten.

3.5 Pembahasan

Dari penjabaran hasil uji coba pada bab sebelumnya, dapat dibuktikan bahwa DenseNet-121 berhasil diimplementasikan untuk melakukan tugas ekstraksi

fitur terhadap citra kerusakan bangunan pasca bencana alam. Ekstraksi fitur dilakukan dengan DenseNet-121 yang dimodifikasi dengan cara menghapus lapisan FC (*Fully Connected*) yang merupakan lapisan klasifikasi kemudian menggantinya dengan GAP (*Global Average Pooling*) karena sistem yang dibangun hanya fokus pada ekstraksi fitur.

Citra yang digunakan sebagai *input* pada sistem telah melalui praproses data kemudian diekstrak fiturnya oleh DenseNet-121 yang telah dimodifikasi menghasilkan representasi numerik berupa vektor fitur dengan dimensi 1024. Vektor fitur yang dihasilkan oleh DenseNet-121 telah menunjukkan jika model berhasil mengenali karakteristik utama dari kerusakan bangunan dengan selektif karena fokus pada fitur yang memiliki nilai tinggi yang membawa informasi penting dari citra kerusakan bangunan.

DenseNet-121 juga terbukti *robust* terhadap keragaman citra yang telah mengalami berbagai transformasi seperti rotasi acak, *flipping*, *zoom*, dan *shear* karena tetap menghasilkan vektor fitur yang konsisten. Semua ini telah dibuktikan dengan data statistik, perhitungan kuantitatif, dan proyeksi visual. Hasil ini menegaskan jika DenseNet-121 adalah model yang tangguh karena tidak hanya mampu untuk melakukan tugas ekstraksi fitur dari berbagai citra kerusakan bangunan yang beragam tanpa kehilangan informasi penting dari citra bangunan yang mengalami berbagai jenis kerusakan.

Pengukuran terhadap hasil ekstraksi dilakukan dengan kombinasi pendekatan kuantitatif dan kualitatif. *Cosine similarity* dipilih sebagai alat ukur untuk melakukan evaluasi kuantitatif. Hasil dari perhitungan *cosine similarity*

antara citra *original* dan setiap citra augmentasi menghasilkan nilai yang tinggi karena berada pada rentang 0.8658 hingga 0.9081. *Cosine similarity* terbukti menjadi alat ukur yang baik terhadap dua gambar yang berbeda tampilan visualnya memiliki fitur yang mirip secara matematis.

Kemudian proyeksi visual dilakukan untuk evaluasi kualitatif dengan visualisasi 2D dan 3D menggunakan PCA (*Principal Component Analysis*). Hasil visualisasi yang dilakukan menunjukkan distribusi fitur dari citra *original* dan citra augmentasi dengan jarak yang relatif berdekatan karena tidak ada titik yang terpisah jauh secara ekstrem dengan lainnya. Hal tersebut cukup untuk membuktikan bahwa fitur yang diekstraksi dari citra *original* dan citra augmentasi memiliki representasi yang serupa. Dengan ini dapat diketahui jika PCA dapat digunakan sebagai alat bantu untuk melakukan analisis distribusi fitur yang mendukung pengukuran kuantitatif.

Dari visualisasi 2D dan 3D yang telah dilakukan menggunakan PCA seperti telah menunjukkan kemiripan fitur yang dihasilkan oleh citra augmentasi dengan fitur yang diekstrak dari citra *original*. Gambar 4.6 dan gambar 4.7 dapat memberi indikasi bahwa terdapat potensi pengelompokan berdasarkan ciri-ciri kerusakan bangunan.

Penelitian oleh Almais et al. (2024) mengintegrasikan metode GLCM dengan PCA untuk klasifikasi otomatis tingkat kerusakan bangunan pasca bencana alam. Pada penelitian tersebut PCA digunakan untuk visualisasi fitur yang dihasilkan oleh GLCM menggunakan koordinat PC1 untuk menetapkan label tingkat kerusakan bangunan. Tingkat kerusakan bangunan yang ditetapkan

berdasarkan PC1 jika berada pada rentang $n < 0$ termasuk kerusakan ringan, jika berada pada rentang $0 \leq n < 2$ maka diklasifikasikan sebagai rusak sedang, dan termasuk rusak berat jika berada pada rentang $n > 2$.

Rentang nilai untuk tingkat kerusakan bangunan pasca bencana alam tersebut telah divalidasi langsung oleh surveyor. Telah terbukti jika klasifikasi tingkat kerusakan bangunan pasca bencana alam pada penelitian tersebut akurat. Hasil visualisasi menggunakan PCA pada penelitian Almais et al. (2024) dapat dibandingkan dengan hasil visualisasi PCA pada penelitian ini karena memiliki pendekatan yang serupa dan memberikan pengetahuan untuk menginterpretasikan hasil visualisasi menggunakan PCA ini terhadap tingkat kerusakan bangunan pasca bencana alam.

Dari penjabaran di atas dapat dinyatakan jika tujuan penelitian ini berhasil tercapai dengan baik karena DenseNet-121 berhasil diimplementasikan sebagai *feature extractor* dari citra bangunan yang mengalami kerusakan akibat bencana alam. Model dapat bekerja dengan handal karena menghasilkan representasi numerik berupa vektor fitur yang selektif dan stabil terhadap keragaman citra kerusakan bangunan. Kombinasi pengukuran terhadap hasil ekstraksi fitur telah membuktikan performa dari sistem yang dikembangkan serta kualitas dari representasi numerik yang dihasilkan. Maka dengan tujuan penelitian yang berhasil tercapai tersebut telah menjadi bukti bahwa DenseNet-121 adalah model yang handal, efektif, dan sangat layak untuk diimplementasikan dalam melakukan tugas ekstraksi fitur terhadap citra kerusakan bangunan pasca bencana alam.

3.6 Integrasi Islam

Integrasi antara perkembangan ilmu pengetahuan dan teknologi dengan islam dapat menjadi hal penting agar dapat membangun karakter keilmuan yang cerdas dengan nilai moral yang baik dan memiliki nilai kemanusiaan. Islam merupakan agama yang tidak hanya mengajarkan berbagai kebaikan tetapi juga mendukung umatnya untuk mengembangkan ilmu pengetahuan.

Rasulullah SAW bersabda bahwa “Sebaik-baiknya manusia adalah yang paling bermanfaat bagi orang lain” (HR ath-Thabrani, Al-Mu’jam al-Ausath, juz VII, hal. 58, dari Jabir bin Abdullah r.a. Dishahihkan Muhammad Nashiruddin al-Albani dalam kitab: As-Silsilah Ash-Shahihah). Hadits ini menjadi dasar untuk menebar kebermanfaatn diri untuk orang lain dan alam yang bisa dimulai dari diri sendiri melalui hal kecil (NU Online, 2023). Hadits ini juga menjadi landasan utama dalam penelitian ini karena diharapkan dengan pemanfaatan teknologi secara tepat dapat memberikan kontribusi untuk kemanusiaan .

Penelitian ini merupakan bagian dari cabang ilmu pengetahuan khususnya bidang teknologi yang bergerak dalam *subset* keilmuan kecerdasan buatan atau AI (*Artificial Intelligence*) yaitu *deep learning*. Melalui penelitian ini dikembangkan sistem yang dapat mengenali pola kerusakan bangunan akibat bencana alam secara otomatis dan cepat. Sehingga proses pemulihan pasca bencana alam dapat dilakukan dengan waktu yang lebih cepat tanpa memerlukan banyak alokasi dana dan sumber daya manusia.

Dalam Islam sendiri upaya untuk menyelamatkan nyawa manusia merupakan bagian dari *maqashid syariah* yang mencakup penjagaan terhadap

agama, jiwa, akal, keturunan, dan harta. Sistem yang dibangun pada penelitian ini dapat digunakan untuk membantu pengambilan keputusan atau penilaian yang cepat mengenai tingkat kerusakan bangunan akibat bencana alam yang membutuhkan penanganan untuk pemulihan. Hal ini sejalan dengan prinsip menjaga jiwa dan menjaga harta benda masyarakat dalam *maqashid syariah*.

Ayat berikut ini menjadi dasar penting terhadap integrasi nilai-nilai keislaman dengan pendekatan teknologi pada penelitian ini:

هُوَ الَّذِي جَعَلَ لَكُمُ الْأَرْضَ ذَلُولًا فَامْشُوا فِي مَنَاكِبِهَا وَكُلُوا مِنْ رِزْقِهِ ۗ وَإِلَيْهِ تُشْجَرُونَ

“Dialah yang menjadikan bumi untuk kamu dalam keadaan mudah dimanfaatkan. Maka, jelajahilah segala penjurunya dan makanlah sebagian dari rezeki-Nya. Hanya kepada-Nya kamu (kembali setelah) dibangkitkan.” (Q.S. Al Mulk: 15)

Ayat tersebut menerangkan bahwa Allah telah menjadikan bumi sebagai tempat yang dapat dijelajahi dan dimanfaatkan oleh manusia pada berbagai aspek kehidupan. Dari ayat ini diisyaratkan jika manusia dituntut untuk melakukan eksploitasi dan memanfaatkan segala potensi yang ada di muka bumi. Hal tersebut harus dilakukan dengan tanggungjawab karena nantinya akan dipertanggungjawabkan kepada Allah SWT. Penelitian ini merupakan salah satu bentuk implementasi ayat tersebut dalam kehidupan. Karena penelitian ini mengembangkan sebuah sistem yang berbasis teknologi yang mana teknologi sendiri adalah bagian dari rezeki Allah berupa ilmu yang dimanfaatkan untuk memberikan manfaat dan menebarkan kebaikan kepada masyarakat. Pemanfaatan kemajuan teknologi seperti pada penelitian ini perlu dijaga agar tetap mengintegrasikan nilai-nilai keislaman karena sejatinya penelitian ini fokus pada efisiensi demi kemaslahatan banyak pihak di masyarakat.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini membuktikan bahwa ekstraksi fitur dari citra kerusakan bangunan pasca bencana alam dapat dilakukan dengan implementasi DenseNet-121 yang dapat mengenali pola kerusakan bangunan dengan mempertahankan informasi penting di dalamnya secara stabil. Pengukuran terhadap hasil ekstraksi fitur oleh DenseNet-121 menunjukkan hasil yang baik terhadap kinerja model dalam melakukan ekstraksi fitur pada berbagai variasi citra kerusakan bangunan pasca bencana alam yang telah dibuktikan melalui data statistik, perhitungan kuantitatif menggunakan *cosine similarity*, serta proyeksi visual melalui PCA 2D dan 3D.

5.2 Saran

Untuk pengembangan penelitian lanjutan di masa yang akan datang, disarankan agar menggunakan dataset yang lebih besar dan bervariasi. Selain itu juga disarankan menggunakan hasil ekstraksi fitur untuk tugas lain yang lebih *advanced* seperti sistem klasifikasi otomatis tingkat kerusakan bangunan yang diintegrasikan ke dalam aplikasi agar mendukung penggunaan sistem di lapangan secara langsung.

DAFTAR PUSTAKA

- Ahishali, M., Degerli, A., Yamac, M., Kiranyaz, S., Chowdhury, M. E. H., Hameed, K., Hamid, T., Mazhar, R., & Gabbouj, M. (2021). Advance Warning Methodologies for COVID-19 Using Chest X-Ray Images. *IEEE Access*, *9*, 41052–41065. <https://doi.org/10.1109/ACCESS.2021.3064927>
- Ahmed, K. T., Shahid, N., Tahir, S. B. U. D., Shabir, A., Khan, M. Y., & Hameed, M. (2024). Signature Elevation Using Parametric Fusion for Large Convolutional Network for Image Extraction. *VFAST Transactions on Software Engineering*, *12*(2), 174–191. <https://doi.org/10.21015/vtse.v12i2.1810>
- Almais, A. T. W., Susilo, A., Naba, A., Sarosa, M., Juwono, A. M., Crysdiyan, C., Muslim, M. A., & Wicaksono, H. (2024). Characterization of Structural Building Damage in Post-Disaster Using GLCM-PCA Analysis Integration. *IEEE Access*, *12*, 146190–146201. <https://doi.org/10.1109/ACCESS.2024.3469637>
- Alqethami, S., Alghamdi, S., Alsubait, T., & Alhakami, H. (2022). RoadNet: Efficient Model to Detect and Classify Road Damages. *Applied Sciences*, *12*(22), 11529. <https://doi.org/10.3390/app122211529>
- Ammelia, I., Lestari, D. S., Al Ghazy, G. T., & Wibowo, Y. A. (2022). Integrasi Materi Kebencanaan pada Jenjang Sekolah Dasar di Kecamatan Ngargoyoso, Jawa Tengah. *Indonesian Journal of Environment and Disaster*, *1*(1), 60–72. <https://doi.org/10.20961/ijed.v1i1.66>
- Basir, O., & Shantta, K. (2021). Deep Learning Feature Extraction for Brain Tumor Characterization and Detection. *IRA-International Journal of Applied Sciences* (ISSN 2455-4499), *16*(1), 1. <https://doi.org/10.21013/jas.v16.n1.p1>
- Benkaddour, M. K. (2021). CNN Based Features Extraction for Age Estimation and Gender Classification. *Informatica*, *45*(5), 697–704. <https://doi.org/10.31449/inf.v45i5.3262>
- BNPB. (2024). *Data Informasi Bencana Indonesia*. Data Informasi Bencana Indonesia (DIBI). https://dibi.bnpb.go.id/statistik_menurut_waktu

- Bouchard, I., Rancourt, M.-È., Aloise, D., & Kalaitzis, F. (2022). On Transfer Learning for Building Damage Assessment from Satellite Imagery in Emergency Contexts. *Remote Sensing*, 14(11), 2532. <https://doi.org/10.3390/rs14112532>
- De Rubeis, V., Lee, J., Anwer, M. S., Yoshida-Montezuma, Y., Andreacchi, A. T., Stone, E., Iftikhar, S., Morgenstern, J. D., Rebinsky, R., Neil-Sztramko, S. E., Alvarez, E., Apatu, E., & Anderson, L. N. (2021). Impact of Disasters, Including Pandemics, on Cardiometabolic Outcomes Across the Life-Course: A Systematic Review. *BMJ Open*, 11(5), e047152. <https://doi.org/10.1136/bmjopen-2020-047152>
- Du, Z. (2024). Review of Convolutional Neural Network. *Science and Technology of Engineering, Chemistry and Environmental Protection*, 1(9). <https://doi.org/10.61173/1xfrez14>
- Dunphy, K., Fekri, M. N., Grolinger, K., & Sadhu, A. (2022). Data Augmentation for Deep-Learning-Based Multiclass Structural Damage Detection Using Limited Information. *Sensors*, 22(16), 6193. <https://doi.org/10.3390/s22166193>
- Elharrouss, O., Akbari, Y., Almadeded, N., & Al-Maadeed, S. (2024). Backbones-Review: Feature Extractor Networks for Deep Learning and Deep Reinforcement Learning Approaches in Computer Vision. *Computer Science Review*, 53, 100645. <https://doi.org/10.1016/j.cosrev.2024.100645>
- Fan, C.-L., & Chung, Y.-J. (2022). Design and Optimization of CNN Architecture to Identify the Types of Damage Imagery. *Mathematics*, 10(19), 3483. <https://doi.org/10.3390/math10193483>
- Faturahman, B. M. (2021). Diskursus Manajemen Bencana Era Covid-19. *Madani Jurnal Politik Dan Sosial Kemasyarakatan*, 13(1), 68–85. <https://doi.org/10.52166/madani.v13i1.2291>
- Fekri-Ershad, S., Al-Imari, M. J., Hamad, M. H., Alsaffar, M. F., Hassan, F. G., Hadi, M. E., & Mahdi, K. S. (2022). Cell Phenotype Classification Based on Joint of Texture Information and Multilayer Feature Extraction in DenseNet. *Computational Intelligence and Neuroscience*, 2022, 1–12. <https://doi.org/10.1155/2022/6895833>

- Goceri, E. (2023). Medical image data augmentation: Techniques, comparisons and interpretations. *Artificial Intelligence Review*, 56(11), 12561–12605. <https://doi.org/10.1007/s10462-023-10453-z>
- Goode, E.-J., Thomas, E., Landeg, O., Duarte-Davidson, R., Hall, L., Roelofs, J., Schulpen, S., De Bruin, A., Wigenstam, E., Liljedahl, B., Waleij, A., Simonsson, L., & Göransson Nyberg, A. (2021). Development of a Rapid Risk and Impact Assessment Tool to Enhance Response to Environmental Emergencies in the Early Stages of a Disaster: A Tool Developed by the European Multiple Environmental Threats Emergency NETWORK (EMETNET) Project. *International Journal of Disaster Risk Science*, 12(4), 528–539. <https://doi.org/10.1007/s13753-021-00352-8>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Express. www.deeplearningbook.org
- Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., & Chen, T. (2018). Recent Advances in Convolutional Neural Networks. *Pattern Recognition*, 77, 354–377. <https://doi.org/10.1016/j.patcog.2017.10.013>
- Hasan, N., Bao, Y., Shawon, A., & Huang, Y. (2021). Densenet Convolutional Neural Networks Application for Predicting Covid-19 Using Ct Image. *SN Computer Science*, 2(5), 389. <https://doi.org/10.1007/s42979-021-00782-7>
- Huang, G., Liu, Z., Pleiss, G., Maaten, L. V. D., & Weinberger, K. Q. (2022). Convolutional Networks with Dense Connectivity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12), 8704–8716. <https://doi.org/10.1109/TPAMI.2019.2918284>
- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2261–2269. <https://doi.org/10.1109/CVPR.2017.243>
- Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., & Kalenichenko, D. (2018). Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2704–2713. <https://doi.org/10.1109/CVPR.2018.00286>

- Jeyabose, A., Eunice, J., Popescu, D. E., Chowdary, M. K., & Hemanth, J. (2022). Deep Learning-Based Leaf Disease Detection in Crops Using Images for Agricultural Applications. *Agronomy*, 12(10), 2395. <https://doi.org/10.3390/agronomy12102395>
- Ji, Q., Huang, J., He, W., & Sun, Y. (2019). Optimized Deep Convolutional Neural Networks for Identification of Macular Diseases from Optical Coherence Tomography Images. *Algorithms*, 12(3), 51. <https://doi.org/10.3390/a12030051>
- Keya, T. A., Leela, A., Habib, N., Rashid, M., & Bakthavatchalam, P. (2023). Mental Health Disorders Due to Disaster Exposure: A Systematic Review and Meta-Analysis. *Cureus*. <https://doi.org/10.7759/cureus.37031>
- Khan, A., Sohail, A., Zahoor, U., & Qureshi, A. S. (2020). A Survey of the Recent Architectures of Deep Convolutional Neural Networks. *Artificial Intelligence Review*, 53(8), 5455–5516. <https://doi.org/10.1007/s10462-020-09825-6>
- Maselli, V., Oppo, D., Moore, A. L., Gusman, A. R., Mtelela, C., Iacopini, D., Taviani, M., Mjema, E., Mulaya, E., Che, M., Tomioka, A. L., Mshiu, E., & Ortiz, J. D. (2020). A 1000-Yr-Old Tsunami in the Indian Ocean Points to Greater Risk for East Africa. *Geology*, 48(8), 808–813. <https://doi.org/10.1130/G47257.1>
- Masud, M. (2022). A Light-Weight Convolutional Neural Network Architecture for Classification of Covid-19 Chest X-Ray Images. *Multimedia Systems*, 28(4), 1165–1174. <https://doi.org/10.1007/s00530-021-00857-8>
- Musleh, D., Alotaibi, M., Alhaidari, F., Rahman, A., & Mohammad, R. M. (2023). Intrusion Detection System Using Feature Extraction with Machine Learning Algorithms in IoT. *Journal of Sensor and Actuator Networks*, 12(2), 29. <https://doi.org/10.3390/jsan12020029>
- Naryanto, H. S. (2019). Analisis Bahaya, Kerentanan dan Risiko Bencana Tsunami di Provinsi Papua Barat. *Jurnal Alami : Jurnal Teknologi Reduksi Risiko Bencana*, 3(1), 10. <https://doi.org/10.29122/alami.v3i1.3399>
- Nikolić, Ž., Benvenuti, E., & Chiozzi, A. (2023). Special Issue on Natural Hazards Risk Assessment for Disaster Mitigation. *Applied Sciences*, 13(3), 1940. <https://doi.org/10.3390/app13031940>

- Noy, I., & Uher, T. (2022). *Economic Consequences of Pre-Covid-19 Epidemics: A Literature Review*. Edward Elgar Publishing. <https://doi.org/10.4337/9781839103735.00015>
- NU Online. (2023a). *Surat Al-Baqarah Ayat 195: Arab, Latin, Terjemah dan Tafsir Lengkap* | *Quran NU Online*. NU Online. <https://quran.nu.or.id/al-baqarah/195>
- NU Online. (2023b). *Surat Ar-Rum Ayat 41: Arab, Latin, Terjemah dan Tafsir Lengkap* | *Quran NU Online*. NU Online. <https://quran.nu.or.id/ar-rum/41>
- NU Online. (2023c, April 8). *Menjadi Insan yang Bermanfaat*. <https://banten.nu.or.id/ramadhan/menjadi-insan-yang-bermanfaat-eiJsz>
- Pardede, J., & Kleb, S. S. (2024). Face Race Classification using ResNet-152 and DenseNet-121. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 12(3), 798. <https://doi.org/10.26760/elkomika.v12i3.798>
- Pattanaik, R. K., Mishra, S., Siddique, M., Gopikrishna, T., & Satapathy, S. (2022). Breast Cancer Classification from Mammogram Images Using Extreme Learning Machine-Based DenseNet121 Model. *Journal of Sensors*, 2022, 1–12. <https://doi.org/10.1155/2022/2731364>
- Pearson, K. (1901). On Lines and Planes of Closest Fit to Systems of Points in Space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11), 559–572. <https://doi.org/10.1080/14786440109462720>
- Peraturan Menteri Pekerjaan Umum dan Perumahan Rakyat Republik Indonesia Nomor 29/PRT/M/2018 tentang Standar Teknis Standar Pelayanan Minimal Pekerjaan Umum dan Perumahan Rakyat, Pub. L. No. 29/PRT/M/2018 (2018). <https://peraturan.bpk.go.id/Details/159743/permen-pupr-no-29prtm2018-tahun-2018>
- Pramana, A. L. (2024). Analisis Perbandingan Evaluasi Metode Deep Learning pada Klasifikasi Jenis Kendaraan. *Journal of Computer Science and Visual Communication Design*, 9, 451–465. <https://doi.org/10.55732/jikdiskomvis.v9i1.1252>
- Pratama, C., Susanta, F. F., Ilahi, R., Khomaini, A. F., & Abdillah, H. W. K. (2019). Coseismic Displacement Accumulation Between 1996 and 2019 Using A

Global Empirical Law on Indonesia Continuously Operating Reference Station (InaCORS). *JGISE: Journal of Geospatial Information Science and Engineering*, 2(2). <https://doi.org/10.22146/jgise.51130>

Qi, B., Sun, H., Zhang, H., Zhao, R., & Gao, X. (2024). Modularizing while Training: A New Paradigm for Modularizing DNN Models. *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, 353–364. <https://doi.org/10.1145/3597503.3608135>

Rohadi, S. (2015). Studi Seismotektonik sebagai Indikator Potensi Gempabumi di Wilayah Indonesia. *Jurnal Meteorologi Dan Geofisika*, 10(2). <https://doi.org/10.31172/jmg.v10i2.40>

Salton, G., Wong, A., & Yang, C. S. (1975). A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18(11), 613–620. <https://doi.org/10.1145/361219.361220>

Schambach, L., Grilli, S. T., & Tappin, D. R. (2021). New High-Resolution Modeling of the 2018 Palu Tsunami, Based on Supershear Earthquake Mechanisms and Mapped Coastal Landslides, Supports a Dual Source. *Frontiers in Earth Science*, 8, 598839. <https://doi.org/10.3389/feart.2020.598839>

Talebi, H., & Milanfar, P. (2021). Learning to Resize Images for Computer Vision Tasks. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 487–496. <https://doi.org/10.1109/ICCV48922.2021.00055>

Taye, M. M. (2023). Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions. *Computation*, 11(3), 52. <https://doi.org/10.3390/computation11030052>

Tuncer, T. (2023). *Damaged Constructions Image Dataset* [Dataset]. <https://www.kaggle.com/datasets/turkertuncer/damaged-constructions-image-dataset>

Undang-Undang RI Nomor 24 Tahun 2007 tentang Penanggulangan Bencana, Pub. L. No. 24 (2007). <https://peraturan.bpk.go.id/Details/39901/uu-no-24-tahun-2007>

Varma, D., Nehansh, A., & Swathy, P. (2023). Data Preprocessing Toolkit: An Approach to Automate Data Preprocessing. *Interantional Journal of*

Scientific Research in Engineering and Management, 07(03).
<https://doi.org/10.55041/IJSREM18270>

Wani, M. A., Bhat, F. A., Afzal, S., & Khan, A. I. (2020). *Advances in Deep Learning* (Vol. 57). Springer Singapore. <https://doi.org/10.1007/978-981-13-6794-6>

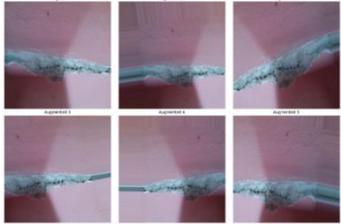
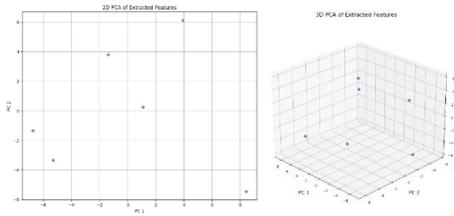
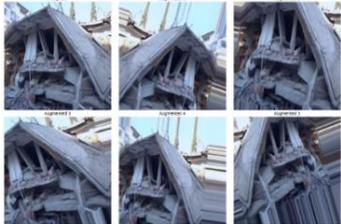
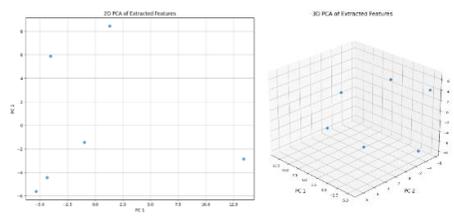
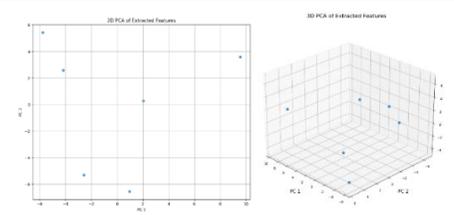
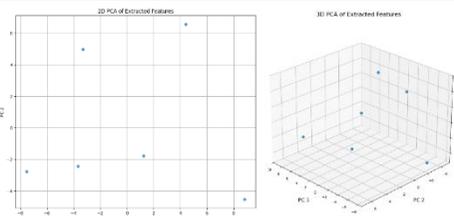
Wardani, K. R., & Leonardi, L. (2023). Klasifikasi Penyakit pada Daun Anggur menggunakan Metode Convolutional Neural Network. *Jurnal Tekno Insentif*, 17(2), 112–126. <https://doi.org/10.36787/jti.v17i2.1130>

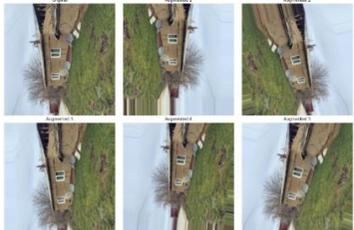
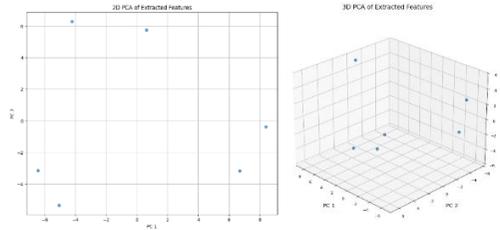
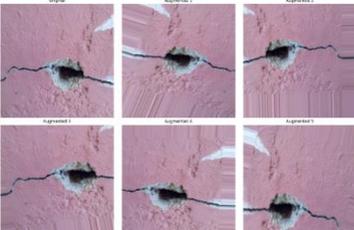
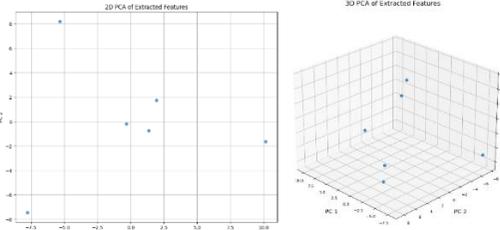
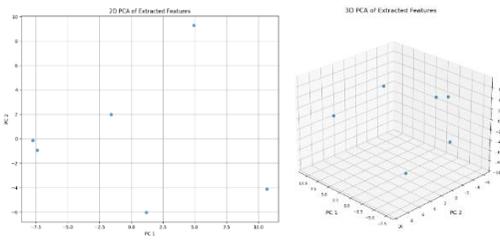
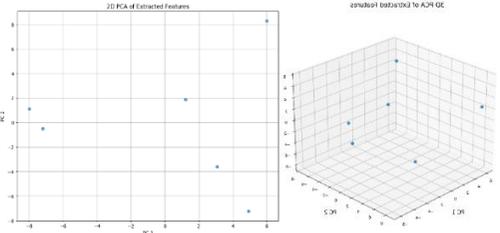
Zhang, Y.-D., Satapathy, S. C., Zhang, X., & Wang, S.-H. (2024). Covid-19 Diagnosis Via Densenet and Optimization of Transfer Learning Setting. *Cognitive Computation*, 16(4), 1649–1665. <https://doi.org/10.1007/s12559-020-09776-8>

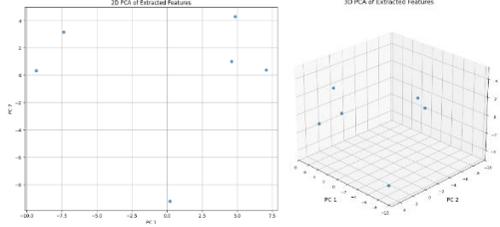
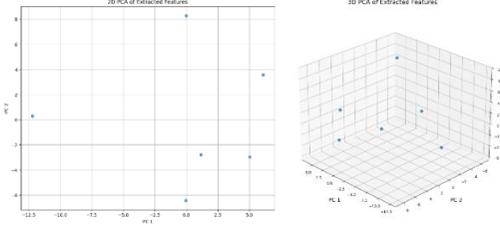
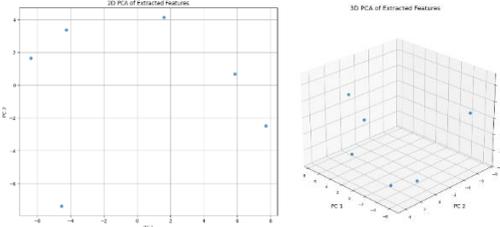
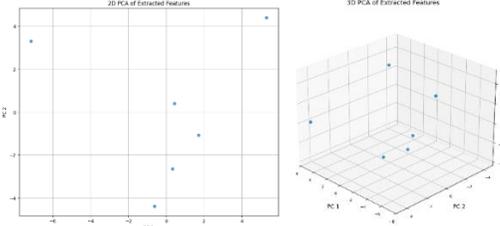
Zhou, Y., Long, X., Sun, M., & Chen, Z. (2022). Bearing Fault Diagnosis Based on Gramian Angular Field and Densenet. *Mathematical Biosciences and Engineering*, 19(12), 14086–14101. <https://doi.org/10.3934/mbe.2022656>

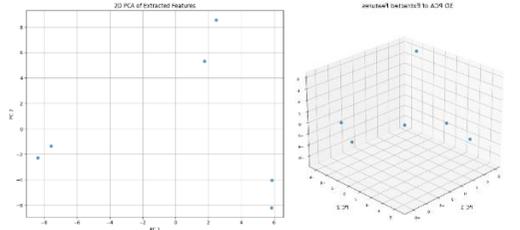
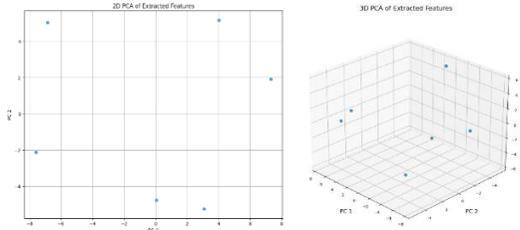
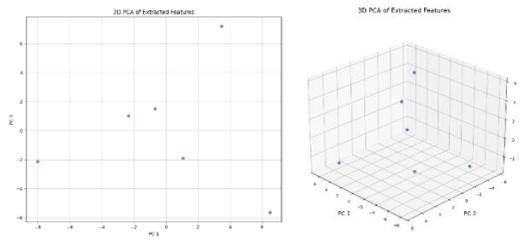
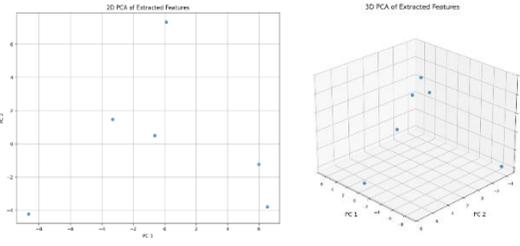
LAMPIRAN

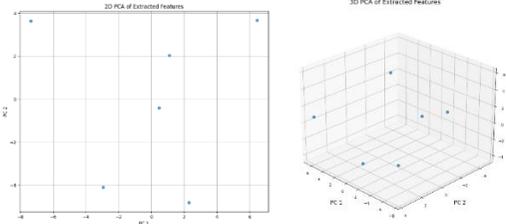
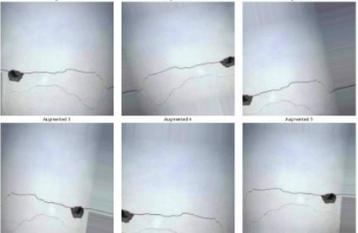
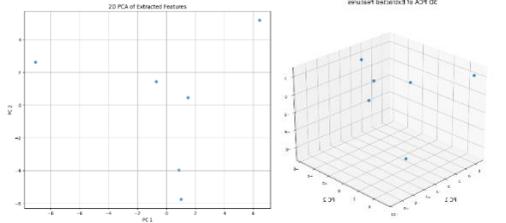
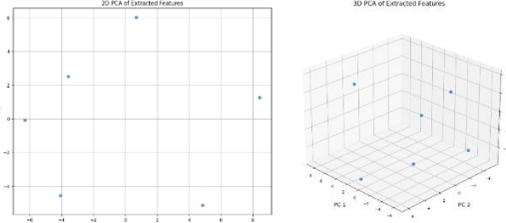
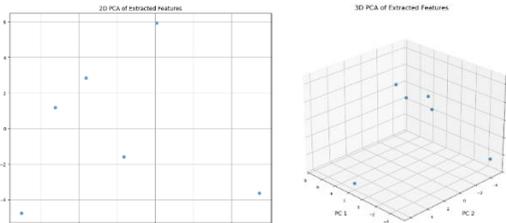
Lampiran 1 Hasil Ekstraksi Fitur

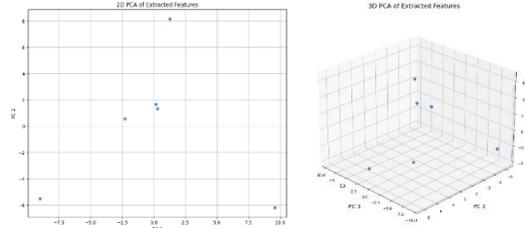
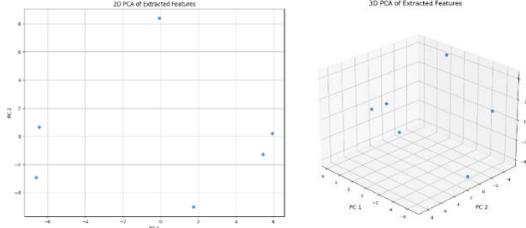
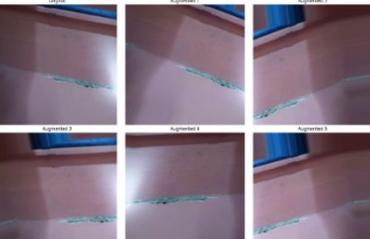
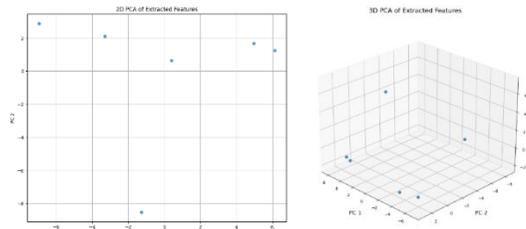
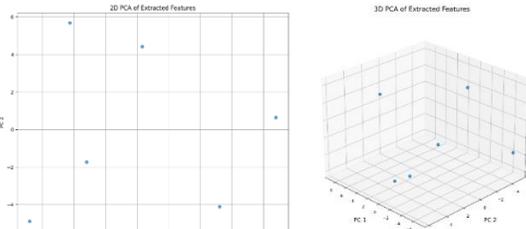
No	Citra <i>Original</i> dan Citra Augmentasi	<i>Cosine Similarity</i>	Visualisasi PCA
1		<p><i>augmented image 1: 0.8830</i> <i>augmented image 2: 0.8173</i> <i>augmented image 3: 0.8510</i> <i>augmented image 4: 0.8817</i> <i>augmented image 5: 0.8309</i></p>	
2		<p><i>augmented image 1: 0.8830</i> <i>augmented image 2: 0.8173</i> <i>augmented image 3: 0.8510</i> <i>augmented image 4: 0.8817</i> <i>augmented image 5: 0.8309</i></p>	
3		<p><i>augmented image 1: 0.9036</i> <i>augmented image 2: 0.8230</i> <i>augmented image 3: 0.8635</i> <i>augmented image 4: 0.8491</i> <i>augmented image 5: 0.8355</i></p>	
4		<p><i>augmented image 1: 0.8824</i> <i>augmented image 2: 0.8948</i> <i>augmented image 3: 0.8913</i> <i>augmented image 4: 0.8503</i> <i>augmented image 5: 0.8534</i></p>	

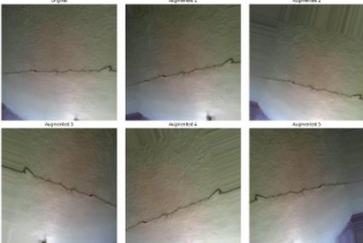
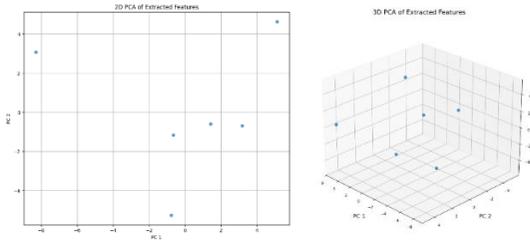
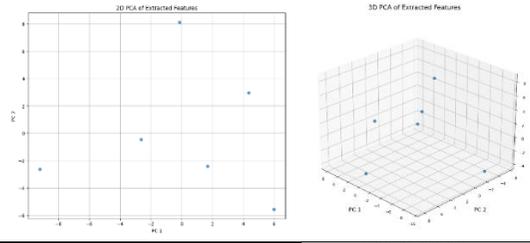
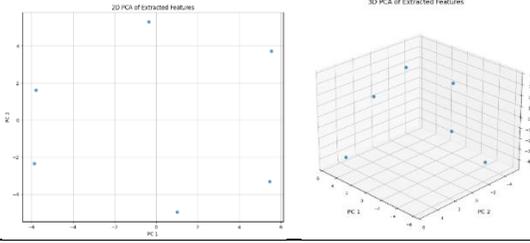
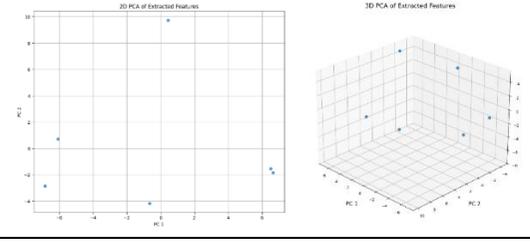
No	Citra <i>Original</i> dan Citra Augmentasi	<i>Cosine Similarity</i>	Visualisasi PCA
5		<p> <i>augmented image 1: 0.8512</i> <i>augmented image 2: 0.8539</i> <i>augmented image 3: 0.9239</i> <i>augmented image 4: 0.8336</i> <i>augmented image 5: 0.9017</i> </p>	
6		<p> <i>augmented image 1: 0.8175</i> <i>augmented image 2: 0.8583</i> <i>augmented image 3: 0.8693</i> <i>augmented image 4: 0.8664</i> <i>augmented image 5: 0.8358</i> </p>	
7		<p> <i>augmented image 1: 0.8715</i> <i>augmented image 2: 0.8641</i> <i>augmented image 3: 0.7565</i> <i>augmented image 4: 0.8233</i> <i>augmented image 5: 0.8147</i> </p>	
8		<p> <i>augmented image 1: 0.8713</i> <i>augmented image 2: 0.8338</i> <i>augmented image 3: 0.8502</i> <i>augmented image 4: 0.8609</i> <i>augmented image 5: 0.8502</i> </p>	

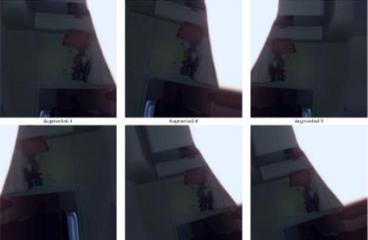
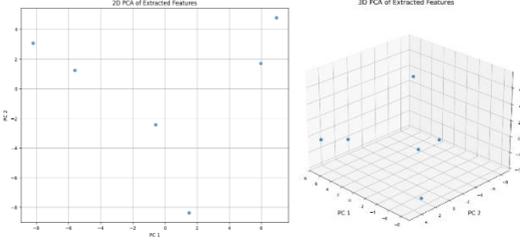
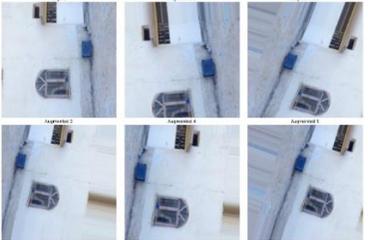
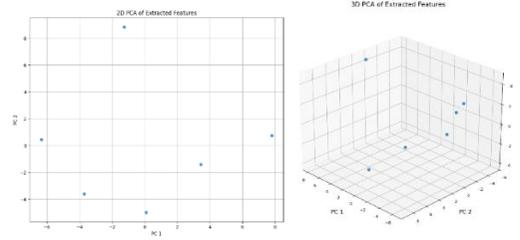
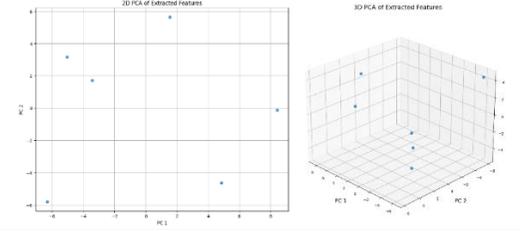
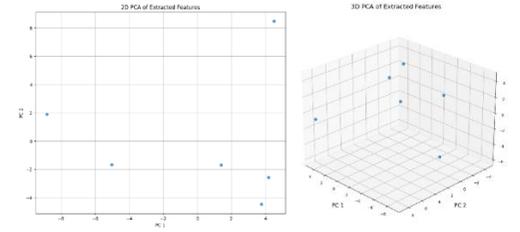
No	Citra <i>Original</i> dan Citra <i>Augmentasi</i>	<i>Cosine Similarity</i>	Visualisasi PCA
9		<p> <i>augmented image 1: 0.8146</i> <i>augmented image 2: 0.9254</i> <i>augmented image 3: 0.7689</i> <i>augmented image 4: 0.8178</i> <i>augmented image 5: 0.9421</i> </p>	
10		<p> <i>augmented image 1: 0.8576</i> <i>augmented image 2: 0.9021</i> <i>augmented image 3: 0.9006</i> <i>augmented image 4: 0.7727</i> <i>augmented image 5: 0.8574</i> </p>	
11		<p> <i>augmented image 1: 0.8658</i> <i>augmented image 2: 0.8644</i> <i>augmented image 3: 0.8820</i> <i>augmented image 4: 0.8240</i> <i>augmented image 5: 0.8732</i> </p>	
12		<p> <i>augmented image 1: 0.8356</i> <i>augmented image 2: 0.9156</i> <i>augmented image 3: 0.9188</i> <i>augmented image 4: 0.9069</i> <i>augmented image 5: 0.8857</i> </p>	

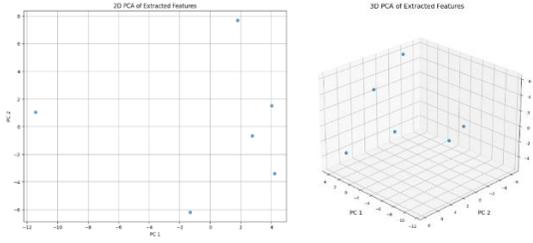
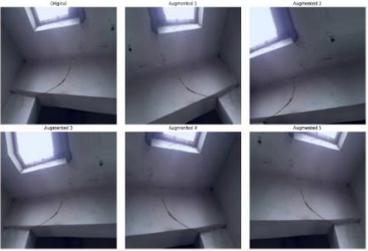
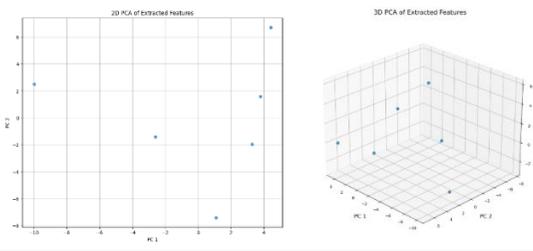
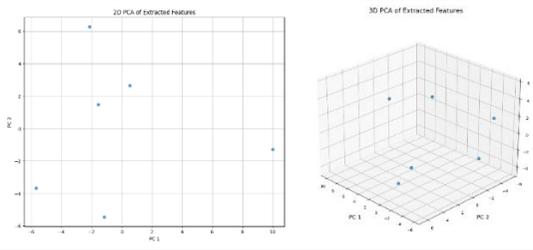
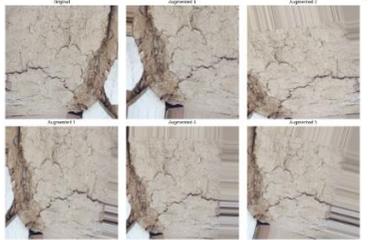
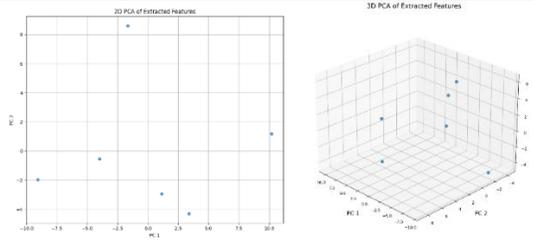
No	Citra <i>Original</i> dan Citra Augmentasi	<i>Cosine Similarity</i>	Visualisasi PCA
13		<p> <i>augmented image 1: 0.8957</i> <i>augmented image 2: 0.8624</i> <i>augmented image 3: 0.8927</i> <i>augmented image 4: 0.8693</i> <i>augmented image 5: 0.9089</i> </p>	
14		<p> <i>augmented image 1: 0.8126</i> <i>augmented image 2: 0.8245</i> <i>augmented image 3: 0.8170</i> <i>augmented image 4: 0.8471</i> <i>augmented image 5: 0.8913</i> </p>	
15		<p> <i>augmented image 1: 0.9086</i> <i>augmented image 2: 0.9115</i> <i>augmented image 3: 0.9247</i> <i>augmented image 4: 0.8908</i> <i>augmented image 5: 0.9031</i> </p>	
16		<p> <i>augmented image 1: 0.8573</i> <i>augmented image 2: 0.8407</i> <i>augmented image 3: 0.8351</i> <i>augmented image 4: 0.8255</i> <i>augmented image 5: 0.8922</i> </p>	

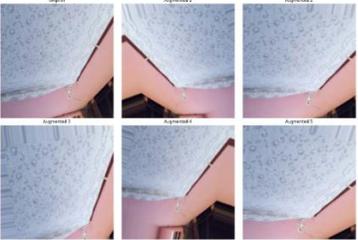
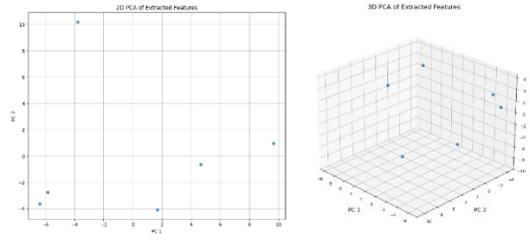
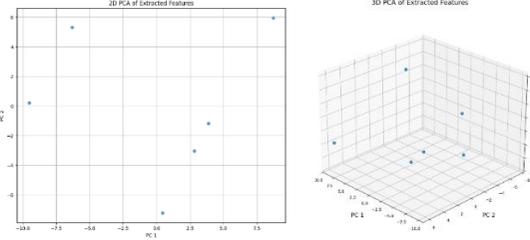
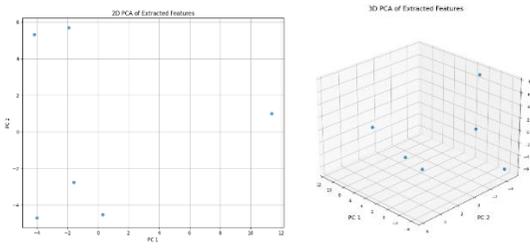
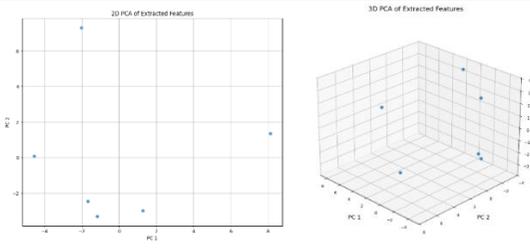
No	Citra <i>Original</i> dan Citra Augmentasi	<i>Cosine Similarity</i>	Visualisasi PCA
17		<p> <i>augmented image 1: 0.8927</i> <i>augmented image 2: 0.8770</i> <i>augmented image 3: 0.8641</i> <i>augmented image 4: 0.9115</i> <i>augmented image 5: 0.8385</i> </p>	
18		<p> <i>augmented image 1: 0.8399</i> <i>augmented image 2: 0.7475</i> <i>augmented image 3: 0.8214</i> <i>augmented image 4: 0.8273</i> <i>augmented image 5: 0.8231</i> </p>	
19		<p> <i>augmented image 1: 0.8880</i> <i>augmented image 2: 0.8816</i> <i>augmented image 3: 0.8050</i> <i>augmented image 4: 0.8448</i> <i>augmented image 5: 0.8059</i> </p>	
20		<p> <i>augmented image 1: 0.9138</i> <i>augmented image 2: 0.9013</i> <i>augmented image 3: 0.9037</i> <i>augmented image 4: 0.9249</i> <i>augmented image 5: 0.9199</i> </p>	

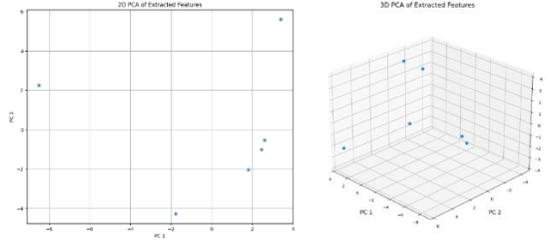
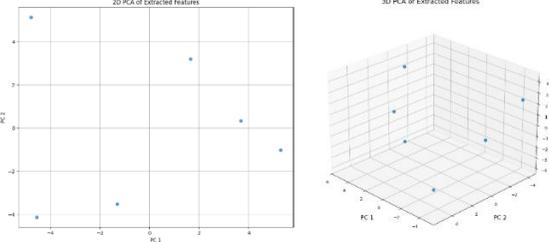
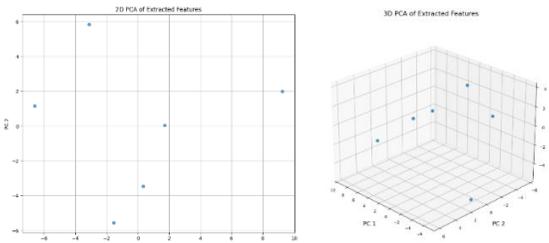
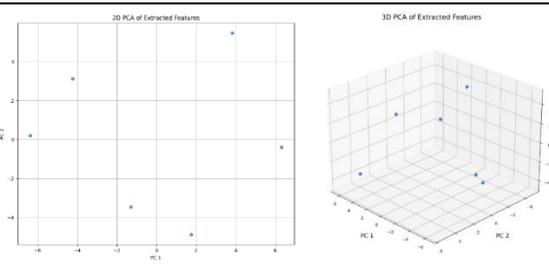
No	Citra <i>Original</i> dan Citra Augmentasi	<i>Cosine Similarity</i>	Visualisasi PCA
21		<p> <i>augmented image 1: 0.7930</i> <i>augmented image 2: 0.8757</i> <i>augmented image 3: 0.8616</i> <i>augmented image 4: 0.8356</i> <i>augmented image 5: 0.8729</i> </p>	
22		<p> <i>augmented image 1: 0.8169</i> <i>augmented image 2: 0.8629</i> <i>augmented image 3: 0.8501</i> <i>augmented image 4: 0.8226</i> <i>augmented image 5: 0.8458</i> </p>	
23		<p> <i>augmented image 1: 0.9327</i> <i>augmented image 2: 0.8257</i> <i>augmented image 3: 0.8585</i> <i>augmented image 4: 0.8175</i> <i>augmented image 5: 0.8116</i> </p>	
24		<p> <i>augmented image 1: 0.8993</i> <i>augmented image 2: 0.9304</i> <i>augmented image 3: 0.9028</i> <i>augmented image 4: 0.8967</i> <i>augmented image 5: 0.8902</i> </p>	

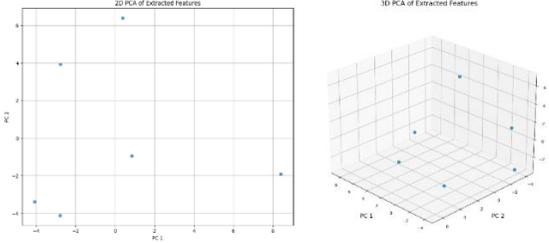
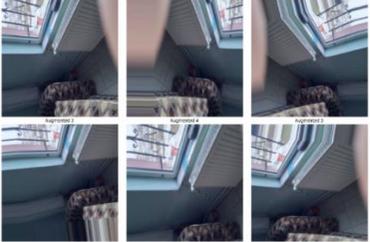
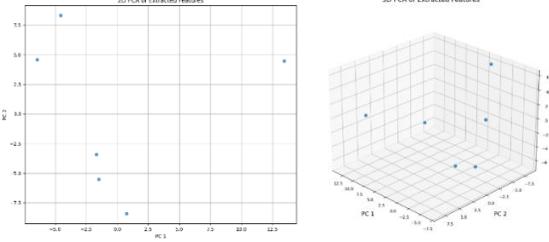
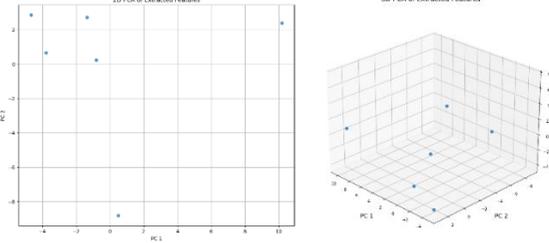
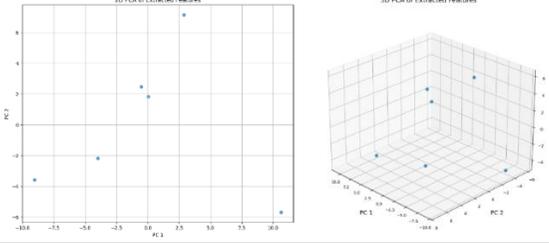
No	Citra <i>Original</i> dan Citra Augmentasi	<i>Cosine Similarity</i>	Visualisasi PCA
25		<p> <i>augmented image 1: 0.8493</i> <i>augmented image 2: 0.7617</i> <i>augmented image 3: 0.8105</i> <i>augmented image 4: 0.7674</i> <i>augmented image 5: 0.8035</i> </p>	
26		<p> <i>augmented image 1: 0.8722</i> <i>augmented image 2: 0.8650</i> <i>augmented image 3: 0.8659</i> <i>augmented image 4: 0.9027</i> <i>augmented image 5: 0.8765</i> </p>	
27		<p> <i>augmented image 1: 0.8919</i> <i>augmented image 2: 0.9121</i> <i>augmented image 3: 0.8669</i> <i>augmented image 4: 0.8650</i> <i>augmented image 5: 0.8639</i> </p>	
28		<p> <i>augmented image 1: 0.8377</i> <i>augmented image 2: 0.9049</i> <i>augmented image 3: 0.8367</i> <i>augmented image 4: 0.8977</i> <i>augmented image 5: 0.8272</i> </p>	

No	Citra <i>Original</i> dan Citra Augmentasi	<i>Cosine Similarity</i>	Visualisasi PCA
29		<p> <i>augmented image 1: 0.9022</i> <i>augmented image 2: 0.8828</i> <i>augmented image 3: 0.8543</i> <i>augmented image 4: 0.8937</i> <i>augmented image 5: 0.8738</i> </p>	
30		<p> <i>augmented image 1: 0.8509</i> <i>augmented image 2: 0.8307</i> <i>augmented image 3: 0.8816</i> <i>augmented image 4: 0.9093</i> <i>augmented image 5: 0.8646</i> </p>	
31		<p> <i>augmented image 1: 0.8305</i> <i>augmented image 2: 0.8993</i> <i>augmented image 3: 0.8423</i> <i>augmented image 4: 0.8841</i> <i>augmented image 5: 0.7964</i> </p>	
32		<p> <i>augmented image 1: 0.8838</i> <i>augmented image 2: 0.8258</i> <i>augmented image 3: 0.9439</i> <i>augmented image 4: 0.9244</i> <i>augmented image 5: 0.8598</i> </p>	

No	Citra <i>Original</i> dan Citra Augmentasi	<i>Cosine Similarity</i>	Visualisasi PCA
33		<p> <i>augmented image 1: 0.7330</i> <i>augmented image 2: 0.7287</i> <i>augmented image 3: 0.7381</i> <i>augmented image 4: 0.7543</i> <i>augmented image 5: 0.7674</i> </p>	
34		<p> <i>augmented image 1: 0.9153</i> <i>augmented image 2: 0.8316</i> <i>augmented image 3: 0.9364</i> <i>augmented image 4: 0.9490</i> <i>augmented image 5: 0.9296</i> </p>	
35		<p> <i>augmented image 1: 0.8481</i> <i>augmented image 2: 0.8929</i> <i>augmented image 3: 0.9167</i> <i>augmented image 4: 0.8946</i> <i>augmented image 5: 0.9430</i> </p>	
36		<p> <i>augmented image 1: 0.8103</i> <i>augmented image 2: 0.7909</i> <i>augmented image 3: 0.8475</i> <i>augmented image 4: 0.8162</i> <i>augmented image 5: 0.7098</i> </p>	

No	Citra <i>Original</i> dan Citra Augmentasi	<i>Cosine Similarity</i>	Visualisasi PCA
37		<p> <i>augmented image 1: 0.8168</i> <i>augmented image 2: 0.8487</i> <i>augmented image 3: 0.7900</i> <i>augmented image 4: 0.8218</i> <i>augmented image 5: 0.8411</i> </p>	
38		<p> <i>augmented image 1: 0.7522</i> <i>augmented image 2: 0.8018</i> <i>augmented image 3: 0.8611</i> <i>augmented image 4: 0.8591</i> <i>augmented image 5: 0.8041</i> </p>	
39		<p> <i>augmented image 1: 0.8311</i> <i>augmented image 2: 0.9016</i> <i>augmented image 3: 0.8383</i> <i>augmented image 4: 0.8581</i> <i>augmented image 5: 0.7825</i> </p>	
40		<p> <i>augmented image 1: 0.8827</i> <i>augmented image 2: 0.8616</i> <i>augmented image 3: 0.8972</i> <i>augmented image 4: 0.8489</i> <i>augmented image 5: 0.8749</i> </p>	

41		<p> <i>augmented image 1: 0.8332</i> <i>augmented image 2: 0.8240</i> <i>augmented image 3: 0.8629</i> <i>augmented image 4: 0.8400</i> <i>augmented image 5: 0.8091</i> </p>	
42		<p> <i>augmented image 1: 0.8958</i> <i>augmented image 2: 0.8682</i> <i>augmented image 3: 0.8856</i> <i>augmented image 4: 0.8697</i> <i>augmented image 5: 0.8740</i> </p>	
43		<p> <i>augmented image 1: 0.8690</i> <i>augmented image 2: 0.8829</i> <i>augmented image 3: 0.8295</i> <i>augmented image 4: 0.8072</i> <i>augmented image 5: 0.8603</i> </p>	
44		<p> <i>augmented image 1: 0.8669</i> <i>augmented image 2: 0.8560</i> <i>augmented image 3: 0.8744</i> <i>augmented image 4: 0.8897</i> <i>augmented image 5: 0.9044</i> </p>	

No	Citra <i>Original</i> dan Citra Augmentasi	<i>Cosine Similarity</i>	Visualisasi PCA
45		<p> <i>augmented image 1: 0.8206</i> <i>augmented image 2: 0.9070</i> <i>augmented image 3: 0.8961</i> <i>augmented image 4: 0.8714</i> <i>augmented image 5: 0.8713</i> </p>	
46		<p> <i>augmented image 1: 0.9170</i> <i>augmented image 2: 0.8633</i> <i>augmented image 3: 0.8373</i> <i>augmented image 4: 0.9044</i> <i>augmented image 5: 0.8655</i> </p>	
47		<p> <i>augmented image 1: 0.9324</i> <i>augmented image 2: 0.9164</i> <i>augmented image 3: 0.8706</i> <i>augmented image 4: 0.8860</i> <i>augmented image 5: 0.9486</i> </p>	
48		<p> <i>augmented image 1: 0.7308</i> <i>augmented image 2: 0.8685</i> <i>augmented image 3: 0.8797</i> <i>augmented image 4: 0.8239</i> <i>augmented image 5: 0.8439</i> </p>	

No	Citra <i>Original</i> dan Citra Augmentasi	<i>Cosine Similarity</i>	Visualisasi PCA
49		<p> <i>augmented image 1: 0.9329</i> <i>augmented image 2: 0.9146</i> <i>augmented image 3: 0.9476</i> <i>augmented image 4: 0.8873</i> <i>augmented image 5: 0.8882</i> </p>	