

**RESTORASI FOTO AKIBAT KERUSAKAN DIGITAL MENGGUNAKAN
DEEP LEARNING BERBASIS *GENERATIVE ADVERSARIAL NETWORKS*
(GANs)**

LAPORAN TUGAS AKHIR

Laporan ini Disusun untuk Memenuhi Salah Satu Syarat Memperoleh Gelar Sarjana
Strata 1 (S1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri
Universitas Islam Sultan Agung Semarang



DISUSUN OLEH:

MINAN ABDILLAH

NIM 32602100065

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG**

2025

***PHOTO RESTORATION FROM DIGITAL DAMAGE USING DEEP
LEARNING BASED ON GENERATIVE ADVERSARIAL NETWORKS (GANs)***

FINAL PROJECT

*Proposed to complete the requirement to obtain a bachelor's degree (S1) at
Informatics Engineering Departement of Industrial Technology Faculty
Sultan Agung Islamic University*



ARRANGED BY:

MINAN ABDILLAH

32602100065

***MAJORING OF INFORMATICS ENGINEERING
INDUSTRIAL TECHNOLOGY FACULTY
SULTAN AGUNG ISLAMIC UNIVERSITY
SEMARANG***

2025

LEMBAR PENGESAHAN TUGAS AKHIR

RESTORASI FOTO AKIBAT KERUSAKAN DIGITAL MENGGUNAKAN DEEP LEARNING BERBASIS GENERATIVE ADVERSARIAL NETWORKS (GANs)

Minan Abdillah
NIM 32602100065

Telah dipertahankan di depan tim penguji ujian sarjana tugas akhir
Program Studi Teknik Informatika
Universitas Islam Sultan Agung
Pada tanggal : 11-06-2025

TIM PENGUJI UJIAN SARJANA :

Mustafa, ST, MM, M. Kom

NIK. 210610040

(Ketua Penguji)

Badie'ah, ST, M. Kom

NIK. 210615044

(Anggota Penguji)

Ir. Sri Mulyono, M. Eng

NIK. 210616049

(Pembimbing)

Semarang, 11-06-2025

Mengetahui,

Kaprodi Teknik Informatika
Universitas Islam Sultan Agung

Moch. Taufik, ST, MIT

NIK. 210604034

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Minan Abdillah

NIM : 32602100065

Judul Tugas Akhir : *RESTORASI FOTO AKIBAT KERUSAKAN DIGITAL MENGGUNAKAN DEEP LEARNING BERBASIS GENERATIVE ADVERSARIAL NETWORKS (GANs).*

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 15-07-2025

Yang Menyatakan,



Minan Abdillah

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Minan Abdillah
NIM : 32602100065
Program Studi : Teknik Informatika
Fakultas : Teknologi industri
Alamat Asal : Desa Manjang, Kec. Jaken, Kab. Pati

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : ***RESTORASI FOTO AKIBAT KERUSAKAN DIGITAL MENGGUNAKAN DEEP LEARNING BERBASIS GENERATIVE ADVERSARIAL NETWORKS (GANs).***

Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti *Non-Eksklusif* untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiatisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan Agung.

Semarang, 15-07-2025

enyatakan

Abdillah

KATA PENGANTAR

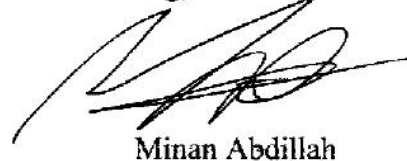
Dengan mengucapkan syukur alhamdulillah atas kehadiran Allah SWT yang telah memberikan rahmat dan karunianya kepada penulis, sehingga dapat menyelesaikan Tugas Akhir dengan judul “Restorasi Foto Akibat Kerusakan Digital Menggunakan *Deep Learning* Berbasis *Generative Adversarial Networks (GANs)*” ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana (S-1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.

Tugas Akhir ini disusun dan dibuat dengan adanya bantuan dari berbagai pihak, materi maupun teknis, oleh karena itu saya selaku penulis mengucapkan terima kasih kepada:

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, S.H., M.H yang mengizinkan penulis menimba ilmu di kampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Novi Marlyana, S.T., M.T.
3. Dosen pembimbing I penulis Ir. Sri Mulyono, M. Eng yang telah meluangkan waktu dan memberi ilmu. Serta memberikan banyak nasehat dan saran.
4. Orang tua penulis yang telah mengizinkan untuk menyelesaikan laporan ini,
5. Dan kepada semua pihak yang tidak dapat saya sebutkan satu persatu.

Dengan segala kerendahan hati, penulis menyadari masih terdapat banyak kekurangan dari segi kualitas atau kuantitas maupun dari ilmu pengetahuan dalam penyusunan laporan, sehingga penulis mengharapkan adanya saran dan kritikan yang bersifat membangun demi kesempurnaan laporan ini dan masa mendatang.

Semarang, 04-08-2025



Minan Abdillah

DAFTAR ISI

COVER.....	i
LEMBAR PENGESAHAN TUGAS AKHIR	iii
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH.....	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	x
DAFTAR TABEL.....	xii
ABSTRAK.....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	2
1.3 Pembatasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Manfaat	3
1.6 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	5
2.1 Tinjauan Pustaka	5
2.2 Dasar Teori.....	8
2.2.1 Restorasi Citra.....	8
2.2.2 Kerusakan Digital.....	9
2.2.3 <i>Deep Learning</i>	9
2.2.4 <i>Image-to-Image Translation (I2I)</i>	11
2.2.5 <i>Generative Adversarial Networks (GANs)</i>	14

2.2.6	Pix2Pix Sebagai Model	18
2.2.7	<i>Generator</i> (U-Net)	20
2.2.8	<i>Discriminator</i> (PatchGAN).....	21
2.2.9	Parameter.....	22
BAB III METODOLOGI PENELITIAN.....		24
3.1	Metode Penelitian.....	24
3.1.1	Studi Literatur	25
3.1.2	Pengumpulan Data	25
3.1.3	<i>Preprocessing</i> Data.....	26
3.1.4	Pelatihan Model	27
3.1.5	Evaluasi Model	29
3.1.6	<i>Deployment</i> Model.....	31
3.2	Analisa Kebutuhan.....	32
3.3	Penggunaan Sistem	34
3.4	Perancangan <i>User Interface</i>	35
3.4.1	Halaman Awal Sistem.....	35
3.4.2	Tampilan Hasil Sistem.....	36
BAB IV HASIL DAN ANALISIS PENELITIAN		38
4.1	Hasil Pengumpulan Data.....	38
4.2	Hasil <i>Preprocessing</i> Data.....	41
4.2.1	Implementasi Proses <i>Preprocessing</i>	41
4.2.2	Visualisasi Hasil <i>Preprocessing</i>	44
4.3	Hasil Pemodelan	46

4.3.1	Ringkasan Model <i>Generator</i> (U-Net)	46
4.3.2	Ringkasan Model <i>Discriminator</i> (PatchGAN).....	50
4.3.3	Analisis Komparatif <i>Generator</i> dan <i>Discriminator</i>	52
4.4	Hasil Pelatihan	53
4.4.1	Perbandingan <i>Loss</i>	53
4.4.2	Waktu Pelatihan	55
4.5	Hasil Evaluasi	56
4.5.1	Evaluasi SSIM.....	56
4.5.2	Evaluasi PSNR.....	58
4.6	Hasil Implementasi.....	61
4.6.1	Implementasi Tipe 1 (Ringan/Mudah).....	62
4.6.2	Implementasi Tipe 2 (Sedang)	63
4.6.3	Implementasi Tipe 3 (Berat/Sulit).....	64
BAB V KESIMPULAN DAN SARAN		65
5.1	Kesimpulan	65
5.2	Saran.....	65
DAFTAR PUSTAKA.....		66

DAFTAR GAMBAR

Gambar 2. 1 Arsitektur GAN	14
Gambar 2. 2 Arsitektur pix2pix (Adiyaman <i>dkk.</i> , 2024).....	18
Gambar 2. 3 Arsitektur <i>Generator</i> (U-Net) (Weng and Zhu, 2021).....	20
Gambar 2. 4 Arsitektur <i>Discriminator</i> (PatchGAN) (Kushwaha <i>dkk.</i> , 2023).....	21
Gambar 3. 1 Tahap Penelitian	24
Gambar 3. 2 <i>Flowchart</i> alur pengumpulan data.....	26
Gambar 3. 3 Proses <i>Training</i> GAN	27
Gambar 3. 4 <i>Flowchart deployment</i>	31
Gambar 3. 5 <i>Flowchart</i> alur kerja sistem.....	34
Gambar 3. 6 Tampilan halaman awal sistem	35
Gambar 3. 7 Tampilan hasil sistem.....	36
Gambar 4. 1 Dataset citra wajah asli dan rusak	38
Gambar 4. 2 Pengurutan normal_data.....	42
Gambar 4. 3 Pengurutan rusak data	42
Gambar 4. 4 <i>Pixel</i> BGR	43
Gambar 4. 5 <i>Pixel</i> RGB	43
Gambar 4. 6 <i>Pixel Grayscale</i>	43
Gambar 4. 7 Normalisasi RGB	44
Gambar 4. 8 Normalisasi <i>Grayscale</i>	44
Gambar 4. 9 Citra hasil <i>preprocessing</i>	45
Gambar 4. 10 <i>Genertaor Loss</i> vs <i>Discriminator Loss</i>	53
Gambar 4. 11 Waktu pelatihan.....	55
Gambar 4. 12 SSIM	57
Gambar 4. 13 PSNR.....	59
Gambar 4. 14 Halaman utama.....	61
Gambar 4. 15 Hasil restorasi citra tipe 1	62
Gambar 4. 16 Hasil restorasi citra tipe 2.....	63



DAFTAR TABEL

Table 4. 1 Dataset tipe citra rusak.....	40
Table 4. 2 Ringkasan <i>modelling Generator</i>	49
Table 4. 3 Ringkasan <i>modelling Discriminator</i>	51



ABSTRAK

Kerusakan digital pada citra dapat mengakibatkan hilangnya informasi visual penting, sehingga diperlukan metode pemulihan yang mampu mengembalikan struktur dan detail citra secara optimal. Penelitian ini bertujuan untuk merancang sistem restorasi citra menggunakan pendekatan *Generative Adversarial Networks* (GANs) dengan arsitektur Pix2Pix, yang menggabungkan U-Net sebagai *Generator* dan PatchGAN sebagai *Discriminator*. Data pelatihan bersumber dari dataset CelebA-HQ *resized* yang telah dimodifikasi untuk merepresentasikan tiga kategori tingkat kerusakan, yaitu ringan, sedang, dan berat, dengan ukuran citra 256×256 *pixel*. Proses pelatihan berlangsung selama 100 epoch, dengan penyesuaian rasio pembaruan antara *Generator* dan *Discriminator* pada setiap fase pelatihan guna menjaga kestabilan konvergensi model. Evaluasi kinerja dilakukan menggunakan metrik *Structural Similarity Index Measure* (SSIM) dan *Peak Signal-to-Noise Ratio* (PSNR), di mana model mencapai nilai SSIM tertinggi sebesar 0,9226 dan PSNR sebesar 32,6895 dB. Hasil ini menunjukkan kemampuan model dalam menghasilkan citra restorasi yang menyerupai citra asli, baik dari segi struktur maupun kualitas visual. Pengujian pada berbagai tingkat kerusakan menunjukkan bahwa model bekerja secara optimal pada kerusakan ringan dan sedang, serta tetap memberikan hasil yang layak pada kerusakan berat. Sistem ini juga telah diterapkan dalam bentuk aplikasi web yang ramah pengguna, sehingga memudahkan proses unggah, restorasi, dan pengunduhan citra. Kesimpulannya, pendekatan Pix2Pix berbasis GAN terbukti efektif untuk tugas restorasi citra dan memiliki potensi untuk dikembangkan pada kasus serupa di masa depan.

Kata Kunci : Restorasi Citra, *Generative Adversarial Networks*, Pix2Pix, SSIM, PSNR

ABSTRACT

Digital damage in photographs can result in the loss of critical visual information, necessitating a restoration method capable of optimally recovering both structure and detail. This study aims to design a photo restoration system using the *Generative Adversarial Networks* (GANs) approach with the Pix2Pix architecture, which combines a U-Net as the Generator and a PatchGAN as the Discriminator. The training data is sourced from the resized CelebA-HQ dataset, which has been modified to represent three levels of damage severity: mild, moderate, and severe, with image dimensions of 256×256 pixels. The training process was conducted over 100 epochs, with an adjusted update ratio between the Generator and Discriminator at each training phase to maintain model convergence stability. Performance evaluation was carried out using the *Structural Similarity Index Measure* (SSIM) and *Peak Signal-to-Noise Ratio* (PSNR) metrics, with the model achieving a highest SSIM score of 0.9226 and a PSNR of 32.6895 dB. These results demonstrate the model's capability to produce restored images that closely resemble the original, both in terms of structure and visual quality. Testing across different damage levels showed optimal performance on mild and moderate damage, while still yielding acceptable results on severe damage. The system has also been implemented as a user-friendly web application, facilitating easy image uploading, restoration, and downloading. In conclusion, the Pix2Pix-based GAN approach has proven effective for photo restoration tasks and holds potential for further development in similar applications in the future.

Keywords : Photo Restoration, *Generative Adversarial Networks*, Pix2Pix, SSIM, PSNR

BAB I

PENDAHULUAN

1.1 Latar Belakang

Fotografi memainkan peran penting dalam mendokumentasikan berbagai aspek kehidupan, mulai dari sejarah dan budaya hingga keperluan identifikasi dan medis. Meski demikian, citra digital sering kali mengalami penurunan kualitas akibat kompresi yang berlebihan, resolusi rendah, dan artefak dari pemrosesan citra. Galteri *dkk.*, (2019) menjelaskan bahwa artefak akibat kompresi, seperti *blocking*, dapat menyebabkan distorsi visual dan kehilangan detail penting, yang pada gilirannya mengurangi kualitas citra secara keseluruhan. Selain itu, K. Zhang *dkk.*, (2019) mengungkapkan bahwa *noise* dan resolusi rendah juga mempengaruhi ketajaman dan kejelasan citra. Karena itu, teknologi kecerdasan buatan dibutuhkan untuk memulihkan kualitas citra dan menjaga informasi yang ada di dalamnya.

Berbagai teknik restorasi citra telah dikembangkan, mulai dari metode interpolasi sederhana hingga image inpainting. Namun, menurut Triningsi *dkk.*, (2024), metode konvensional seperti interpolasi bilinear dan bicubic seringkali kurang mampu menjaga detail tekstur dan struktur citra dengan baik, sehingga hasil restorasi tidak memuaskan. Sebagai solusi, metode berbasis *Generative Adversarial Networks* (GANs) mulai banyak digunakan karena kemampuan mereka untuk menghasilkan citra dengan kualitas lebih alami dan detail yang lebih terjaga (Cao *dkk.*, 2023). Teknologi ini telah diaplikasikan dalam berbagai bidang, termasuk pemulihan citra medis dan forensik (Maharani and Salim, 2024).

Seiring perkembangan teknologi, fokus pada jenis citra tertentu menjadi penting mengingat kebutuhan khusus di berbagai bidang. Salah satu jenis citra yang sering mengalami masalah akibat kerusakan digital adalah citra wajah. Karena struktur wajah yang kompleks dan detail halusnya sangat penting untuk

keperluan seperti identifikasi, analisis ekspresi, serta dokumentasi medis dan forensik, dibutuhkan metode restorasi yang mampu mengembalikan kualitas dan informasi tersebut secara akurat.

Salah satu model GAN yang banyak digunakan untuk restorasi citra, terutama pada wajah, adalah Pix2pix. Liu *dkk.*, (2023) menunjukkan bahwa model ini mampu meningkatkan nilai SSIM dari 0,5565 menjadi 0,5930 dan PSNR dari 20,3110 dB menjadi 21,7335 dB dalam proses restorasi. Selain itu, He *dkk.*, (2024) mengembangkan model berbasis Pix2pix dengan menggabungkan *loss* perseptual dan *style loss* serta menambahkan residual block untuk memperbaiki kualitas pelatihan dan hasil restorasi.

Dengan demikian, pemanfaatan teknologi *deep learning* berbasis GAN dapat menjadi solusi efektif dalam mengatasi kerusakan digital pada citra, terutama dalam menjaga kualitas dan detail penting pada citra wajah. Evaluasi keberhasilan metode ini biasanya dilakukan dengan menggunakan metrik seperti *Peak Signal-to-Noise Ratio* (PSNR) dan *Structural Similarity Index* (SSIM).

1.2 Perumusan Masalah

Rumusan masalah dalam penelitian ini adalah sebagai berikut :

1. Bagaimana merancang dan mengembangkan sistem restorasi citra dengan kerusakan digital, seperti coretan acak digital, serta hilangnya bagian citra, menggunakan *Generative Adversarial Networks* (GANs) untuk meningkatkan kualitas citra?
2. Bagaimana metode *Generative Adversarial Networks* (GANs) dapat mengatasi berbagai bentuk kerusakan digital dengan lebih baik dibandingkan metode konvensional?

1.3 Pembatasan Masalah

Batasan masalah ini bertujuan untuk memudahkan dan menghindari adanya kegiatan di luar sasaran, sehingga dalam pembuatan laporan ini perlu ditentukan

suatu batasan masalah. Batasan masalah tersebut sebagai berikut :

1. Jenis kerusakan yang ditangani meliputi coretan acak digital, distorsi blok, serta hilangnya bagian citra.
2. *Dataset* yang digunakan adalah CelebA-HQ *resized* 256 x 256 yang terdiri dari citra wajah yang diperoleh dari kaggle dengan total 6000 citra, 3000 citra untuk citra asli dan 3000 untuk citra rusak.
3. Penelitian ini berfokus pada *Generative Adversarial Networks* (GANs) terutama pix2pix untuk restorasi citra dengan kerusakan digital, tanpa perbandingan mendalam dengan metode konvensional seperti interpolasi atau image inpainting klasik.
4. Evaluasi dilakukan menggunakan metrik kuantitatif yaitu *Peak Signal-to-Noise Ratio* (PSNR) dan *Structural Similarity Index* (SSIM), dan kualitatif yaitu dengan membandingkan secara langsung citra asli dan citra hasil restorasi.

1.4 Tujuan

Penelitian ini bertujuan untuk mengembangkan sistem restorasi citra yang rusak menggunakan model *Generative Adversarial Networks* (GANs) guna menghasilkan citra dengan kualitas dan kejelasan visual yang baik pada foto yang mengalami kerusakan.

1.5 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat bagi pengembang teknologi pemrosesan citra dalam meningkatkan kualitas citra yang telah mengalami penurunan kualitas. Dengan menerapkan model *Generative Adversarial Networks* (GANs) khususnya Pix2pix, sistem yang dikembangkan mampu menghasilkan citra yang lebih tajam dan alami, sehingga dapat mendukung berbagai kebutuhan yang memerlukan visual dengan kualitas tinggi.

1.6 Sistematika Penulisan

Untuk mempermudah penulisan tugas akhir ini, penulis membuat suatu sistematika yang terdiri dari:

BAB I : PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang pemilihan judul, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI

Bab ini memuat penelitian-penelitian sebelumnya dan dasar teori yang berfungsi sebagai sumber atau alat dalam memahami permasalahan yang berkaitan dengan implementasi *Generative Adversarial Networks* (GANs) khususnya pix2pix.

BAB III : METODOLOGI PENELITIAN

Bab ini mengungkapkan proses tahapan - tahapan penelitian dimulai dari analisa kebutuhan sistem, kemudian perancangan sistem hingga *prototype*.

BAB IV : HASIL DAN ANALISIS SISTEM

Pada bab ini penulis mengungkapkan hasil penelitian yakni hasil dari restorasi citra menggunakan *Generative Adversarial Networks* (GANs).

BAB V : KESIMPULAN DAN SARAN

Bab terakhir memuat kesimpulan isi dari keseluruhan uraian bab-bab sebelumnya dan saran-saran dari hasil yang diperoleh dan diharapkan dapat bermanfaat dalam pengembangan selanjutnya.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Restorasi foto yang mengalami kerusakan digital semakin mendapat perhatian karena perannya dalam memulihkan informasi visual yang hilang. Dalam konteks ini, metode *deep learning* telah muncul sebagai solusi untuk merestorasi detail yang rusak. Sebagai contoh, Maulana *dkk.*, (2020) mengusulkan metode *facial inpainting* berbasis GAN dengan penambahan *feature reconstruction loss* yang mampu mempertahankan struktur dan tekstur citra meskipun terdapat kerusakan signifikan pada bagian wajah.

Sementara itu, metode konvensional yang selama ini digunakan untuk restorasi citra masih menghadapi berbagai kendala serius, khususnya dalam hal mempertahankan detail halus dan tekstur asli ketika citra mengalami degradasi yang cukup parah akibat adanya *noise*, kompresi berlebihan, atau artefak visual lainnya. Studi dari Gotamaa & Santiyasaa, (2021) secara jelas menunjukkan bahwa pendekatan tradisional cenderung kurang efektif dalam mengatasi kerusakan struktur visual yang kompleks dan bersifat *non-linear*, sehingga hasil restorasi yang diperoleh sering kali gagal mempertahankan kualitas citra secara keseluruhan dan tidak mampu memenuhi kebutuhan aplikasi yang mengharuskan reproduksi detail dengan tingkat presisi tinggi. Oleh karena itu, kebutuhan akan metode restorasi yang lebih adaptif, canggih, dan mampu mempelajari representasi data secara mendalam semakin mengemuka sebagai solusi untuk mengatasi keterbatasan ini.

Sebagai respon atas berbagai kendala tersebut, teknik *image-to-image translation* yang dikembangkan berbasis *Generative Adversarial Networks* (GANs) muncul sebagai salah satu pendekatan yang sangat potensial karena memungkinkan pemodelan hubungan kompleks secara langsung antara citra input yang rusak dan output hasil restorasi yang diharapkan. Salah satu model

GAN yang populer untuk tugas ini adalah pix2pix, yang memanfaatkan data pasangan citra sebagai dasar pelatihan agar dapat mempelajari pola korelasi dan struktur detail yang hilang secara lebih efektif. Penelitian oleh Salehi & Chalechale (2020) menunjukkan bahwa dengan melakukan penyesuaian parameter secara tepat pada model ini, performa restorasi dapat meningkat secara signifikan, yang diukur melalui peningkatan rata-rata nilai PSNR sebesar 1,8 dB dan nilai SSIM sebesar 0,04 unit, mengindikasikan kemampuan model untuk menghasilkan rekonstruksi yang lebih tajam dan lebih mirip dengan citra asli. Temuan ini sekaligus menggarisbawahi potensi besar dari teknik *image-to-image translation* berbasis GAN, khususnya dalam aplikasi restorasi citra digital yang mengalami berbagai kerusakan.

Hasil restorasi juga sangat dipengaruhi oleh pengembangan arsitektur GAN. Sistem supervisi tambahan diperkenalkan oleh Ding *dkk.*, (2023) dalam struktur model untuk meningkatkan konsistensi hasil, yang menghasilkan peningkatan SSIM sebesar 0,05 unit. Desain inovatif semacam ini sangat krusial untuk menciptakan rekonstruksi visual yang alami dan realistis.

Dalam mendukung ketepatan hasil rekonstruksi, prediksi ketidakpastian pada level *pixel* juga menjadi strategi efektif. Ma (2024) mengembangkan pendekatan berbasis GAN yang memperhitungkan area dengan tingkat ketidakpastian tinggi, sehingga model dapat lebih fokus pada bagian yang paling sulit direkonstruksi, seperti fitur-fitur penting dalam citra wajah.

Meski demikian, penerapan GAN tetap menghadapi tantangan, terutama dalam stabilitas pelatihan dan kontrol terhadap artefak digital. Sriram *dkk.*, (2024) mengidentifikasi bahwa pelatihan yang tidak stabil dan ketergantungan pada data berkualitas tinggi menjadi hambatan yang perlu diatasi agar hasil rekonstruksi konsisten dan bebas dari distorsi tidak alami.

GAN telah digunakan di berbagai bidang restorasi, seperti peningkatan resolusi citra. Menurut Wang *dkk.*, (2023), teknik super-resolusi berbasis GAN dapat meningkatkan PSNR hingga 2,5 dB pada citra optik, yang menunjukkan

fleksibilitas pendekatan ini untuk berbagai jenis data visual. Di sisi lain, K. Liu *dkk.*, (2022) membicarakan potensi teknik *inpainting* berbasis GAN untuk memperbaiki manipulasi digital dan menciptakan bagian citra yang secara visual konsisten, sebuah pendekatan yang sejalan dengan tujuan utama restorasi citra digital.

Selain kualitas visual, aspek konsistensi spasial juga penting dalam rekonstruksi citra. Caruso *dkk.*, (2024) menunjukkan bahwa penerapan algoritma rekonstruksi berbasis *deep learning* pada citra medis mampu meningkatkan kontras dan integritas spasial, sebuah prinsip yang relevan jika diterapkan pada sistem restorasi citra agar hasilnya stabil dan alami.

Dari sisi teoritis, struktur GAN sebagaimana dijelaskan oleh Goodfellow *dkk.*, (2020) memberikan kerangka dasar penting mengenai cara kerja antara *Generator* dan *Discriminator* dalam menghasilkan citra sintetis yang menyerupai distribusi data asli. Pemahaman ini sangat berperan dalam pengembangan metode *image-to-image translation* yang adaptif terhadap berbagai bentuk kerusakan digital.

Berdasarkan berbagai studi yang telah dibahas, pendekatan berbasis *Generative Adversarial Networks* (GANs) melalui teknik *image-to-image translation* menawarkan potensi besar dalam mengembangkan sistem restorasi citra digital yang presisi, natural, dan stabil. Inovasi dalam desain arsitektur, strategi pelatihan, dan pemanfaatan ketidakpastian *pixel* terus mendorong peningkatan kualitas rekonstruksi citra yang mengalami kerusakan kompleks, membuka peluang besar untuk riset dan penerapan lebih lanjut di bidang ini.

2.2 Dasar Teori

2.2.1 Restorasi Foto

Restorasi foto merupakan salah satu cabang penting dalam bidang pemrosesan citra digital yang bertujuan untuk mengembalikan tampilan asli dari citra yang mengalami kerusakan atau degradasi. Kerusakan ini bisa timbul akibat berbagai faktor, seperti gangguan saat pengambilan citra, kesalahan transmisi data, kompresi yang berlebihan, atau kerusakan fisik pada file citra. Dalam dunia digital, bentuk kerusakan yang umum meliputi noise dari sensor kamera, blur akibat gerakan, artefak kompresi (seperti pada format JPEG), hingga *pixel* yang hilang akibat korupsi data. Restorasi citra berfokus pada pemulihan detail citra yang hilang atau rusak melalui teknik pemrosesan citra sehingga hasil akhirnya menyerupai kondisi asli sebelum terjadi degradasi. Tujuan utama dari proses ini adalah meningkatkan kualitas visual citra dan mengembalikan informasi yang sebelumnya sulit dikenali, agar citra dapat dimanfaatkan kembali dalam konteks tertentu, baik secara estetika maupun fungsional.

Restorasi citra tidak hanya penting dalam ranah Fotografi umum, tetapi juga memainkan peran signifikan dalam berbagai aplikasi lain seperti pengolahan citra medis, pelestarian arsip sejarah digital, sistem pengawasan, hingga analisis forensik. Dalam beberapa dekade terakhir, perkembangan teknologi telah mendorong munculnya pendekatan restorasi citra yang lebih canggih, termasuk pendekatan berbasis kecerdasan buatan. Meskipun bagian ini belum membahas metode spesifik, pendekatan modern seperti yang berbasis *deep learning* terbukti mampu menghasilkan kualitas restorasi yang mendekati citra aslinya, bahkan dalam kasus kerusakan kompleks. Salah satu studi terbaru menyatakan bahwa restorasi citra merupakan tugas dasar dalam pemrosesan citra yang berfokus pada pemulihan citra berkualitas tinggi dari citra yang rusak atau bising, serta memainkan peran penting dalam meningkatkan kualitas visual untuk berbagai aplikasi lanjutan (Senthil Anandhi and Jaiganesh, 2025)

2.2.2 Kerusakan Digital

Kerusakan digital pada citra terjadi ketika data citra terganggu selama proses akuisisi, kompresi, penyimpanan, ataupun transmisi, sehingga menurunkan kualitas visual dan informasi yang terkandung di dalamnya. Beberapa bentuk kerusakan digital yang paling umum meliputi scribble yang seolah seperti coretan, area terpotong, dan artefak kompresi seperti blok-blok, serta hilangnya *pixel* atau korupsi data yang dapat muncul akibat kesalahan penulisan pada media penyimpanan maupun gangguan jaringan saat pengiriman file. Selain itu, distorsi warna, *quantization error*, dan *banding (color banding)* juga sering ditemui pada citra dengan rentang dinamis yang terbatas. Menurut Singh dkk., (2022) yang mempelajari strategi mengembalikan informasi asli dari citra yang telah terdegradasi, menekankan bahwa *noise* dapat mengaburkan tepi dan detail halus, sehingga memerlukan teknik khusus untuk memisahkan sinyal citra dari gangguan acak tersebut.

2.2.3 Deep Learning

Deep learning sebagai cabang *machine learning*, memungkinkan pemrosesan citra melalui jaringan saraf tiruan seperti *convolutional neural networks* (CNN) yang secara otomatis mempelajari fitur citra dari data mentah untuk tugas *image-to-image translation* (I2I). Dalam konteks I2I, *deep learning* digunakan untuk memetakan citra dari satu domain, seperti citra yang rusak akibat distorsi, goresan, atau degradasi warna ke domain lain, yaitu citra yang telah direstorasi, sehingga mendukung aplikasi seperti restorasi citra (Alotaibi, 2020). Arsitektur *deep learning*, seperti *U-Net*, memungkinkan model untuk menangkap pola hierarkis dalam data citra, yang penting untuk menghasilkan output berkualitas tinggi dalam restorasi citra. Proses ini bergantung pada struktur jaringan saraf yang terdiri dari lapisan-lapisan yang bekerja secara berurutan untuk mengolah informasi citra secara mendalam.

Setiap lapisan jaringan memproses input dari lapisan sebelumnya melalui operasi linier dan non-linier secara bertahap. Proses ini disebut *forward propagation*, dan pada lapisan ke- l secara umum dapat dituliskan sebagai:

$$z^{(l)} = W^{(l)}a^{(l-1)} + b^{(l)}, \quad a^{(l)} = \sigma(z^{(l)}), \quad (1)$$

di mana $W^{(l)}$ dan $b^{(l)}$ masing-masing adalah bobot dan bias, $a^{(l-1)}$ adalah output dari lapisan sebelumnya, dan σ merupakan fungsi aktivasi seperti ReLU atau Sigmoid. Untuk data berupa citra, jaringan menggunakan operasi konvolusi, yang didefinisikan sebagai:

$$(I * K)[i, j] = \sum_m \sum_n I[i + m, j + n] K[m, n], \quad (2)$$

di mana I adalah input feature map dan K adalah kernel konvolusi. Fungsi kerugian yang paling umum digunakan dalam tugas restorasi citra adalah *Mean Squared Error (MSE)*, yang menghitung rata-rata kuadrat selisih antara *pixel* hasil prediksi dan *pixel* ground truth:

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (Y[i, j] - y'[i, j])^2 \quad (3)$$

Keterangan:

- MSE : Mean Squared Error (MSE), digunakan untuk mengukur seberapa besar kesalahan prediksi.
- M, N : Ukuran gambar (tinggi dan lebar).
- $Y[i, j]$: Nilai *pixel* target (ground truth) pada posisi (i, j) .
- $y'[i, j]$: Nilai *pixel* hasil prediksi dari model pada posisi (i, j) .
- $(Y[i, j] - y'[i, j])^2$: Selisih kuadrat antara target dan prediksi pada *pixel* ke- (i, j) .

Menurut Trigka & Dritsas (2025), arsitektur *Deep learning* seperti *Convolutional Neural Network (CNN)* mampu secara efektif mengekstraksi fitur spasial melalui lapisan konvolusi dan pooling. Untuk tugas restorasi citra yang kompleks, arsitektur seperti *U-net* banyak digunakan karena struktur *encoder-*

decoder-nya yang dilengkapi dengan *skip connections*, yang mempertahankan detail spasial pada saat restorasi. Selain *U-net*, terdapat juga arsitektur lanjutan seperti *ResNet* dan model berbasis attention yang semakin banyak digunakan.

Di sisi lain, Karavarsamis *dkk.*, (2022) menyoroti aplikasi *Deep learning* dalam kondisi nyata yang menantang, seperti restorasi citra akibat hujan, salju, dan kabut. Penelitian mereka menunjukkan bahwa jaringan ringan dan efisien seperti *encoder-decoder CNN*, *residual networks*, dan model dengan mekanisme perhatian sangat efektif digunakan dalam sistem waktu nyata yang diterapkan di perangkat terbatas. Studi tersebut menekankan pentingnya efisiensi komputasi tanpa mengorbankan kualitas visual, terutama dalam aplikasi pemantauan dan penyelamatan berbasis citra.

2.2.4 Image-to-Image Translation (I2I)

Image-to-Image Translation (I2I) merupakan suatu pendekatan dalam bidang visi komputer yang bertujuan untuk mentransformasikan suatu citra dari domain sumber X ke domain target Y , sambil mempertahankan konten semantik utama. Secara matematis, proses ini dinyatakan sebagai pemetaan fungsi:

$$G: X \rightarrow Y$$

yang berarti model *Generator* G belajar menghasilkan citra $y' = G(x)$ yang menyerupai citra nyata $y \in Y$, berdasarkan input $x \in X$. Tujuan utama dari I2I adalah untuk menciptakan citra yang realistis dalam domain target sambil mempertahankan informasi penting dari domain sumber (Salehi and Chalechale, 2020). Untuk mekanisme sendiri terdapat dua pendekatan utama yaitu *supervised* dan *unsupervised* yang digunakan dalam pengembangan model I2I sebagai berikut:

1. *Supervised* I2I (dengan pasangan data)

Pada pendekatan terawasi, model dilatih menggunakan pasangan citra (x, y) dari domain sumber dan target. Dua komponen utama dari fungsi *loss* adalah:

a. *Adversarial Loss*

$$L_{Adv}(G, D) = E_{x,y}[\log D(x, y)] + E_x \left[\log \left(1 - D(x, G(x)) \right) \right] \quad (4)$$

Keterangan:

- G : *Generator*, yang menghasilkan data palsu $G(x)$ dari input x .
- D : *Discriminator*, yang menilai apakah pasangan (x, y) nyata atau palsu.
- x : Input (misalnya gambar atau data awal).
- y : Target asli (misalnya gambar target atau label sebenarnya).
- $D(x, y)$: Output diskriminator untuk pasangan nyata.
- $D(x, G(x))$: Output diskriminator untuk pasangan palsu (hasil dari *Generator*).
- E : Ekspektasi (rata-rata) terhadap seluruh data.

Penjelasan :

- Bagian pertama $E_{x,y}[\log D(x, y)]$ mendorong diskriminator D untuk mengenali pasangan nyata sebagai asli.
- Bagian kedua $E_x \left[\log \left(1 - D(x, G(x)) \right) \right]$ mendorong G untuk menghasilkan citra $G(x)$ yang sulit dibedakan dari nyata oleh D .

b. *L1 Loss (Pixel-wise Loss)*

$$L_1(G) = E_{x,y}[\|y - G(x)\|_1] \quad (5)$$

Keterangan:

- $L_1(G)$: Fungsi L1 loss untuk *Generator* G .
- x : Input gambar (misalnya sketsa, citra grayscale, dsb).
- y : Gambar target atau ground truth (citra yang sebenarnya).
- $G(x)$: Output dari *Generator*, prediksi citra yang dihasilkan dari input x .
- $\|y - G(x)\|_1$: Jarak absolut rata-rata (mean absolute error) antara

gambar prediksi dan gambar target.

- $E_{x,y}$: Ekspektasi atau rata-rata terhadap seluruh pasangan data dalam dataset.

Loss ini menghitung rata-rata selisih absolut antara *pixel* citra target y dan hasil $G(x)$, yang membantu menjaga kesesuaian struktur dan tekstur pada level *pixel*.

c. Total *Loss* atau *Generator Loss* I2I

$$L_{G(total)} = L_{Adv} + \lambda L_1 \quad (6)$$

Fungsi total *loss* menggabungkan komponen *adversarial* dan *L1 loss*, dengan bobot λ untuk mengontrol pengaruh dari *L1* terhadap keseluruhan pembelajaran.

2. *Unsupervised* I2I (tanpa pasangan data)

Untuk kondisi di mana data berpasangan tidak tersedia, digunakan pendekatan tidak terawasi seperti CycleGAN. Sistem ini menggunakan dua fungsi pemetaan:

$$G: X \rightarrow Y, \quad F: Y \rightarrow X$$

a. *Cycle-Consistency Loss*

$$L_{cyc} = E_x[||F(G(x)) - x||_1] + E_y[||G(F(y)) - y||_1] \quad (7)$$

Penjelasan:

- Bagian pertama memastikan bahwa citra yang diubah ke domain target lalu dikembalikan ke domain sumber tetap menyerupai aslinya.
- Bagian kedua melakukan hal serupa dari domain target ke sumber.

b. Total *Loss Unsupervised*

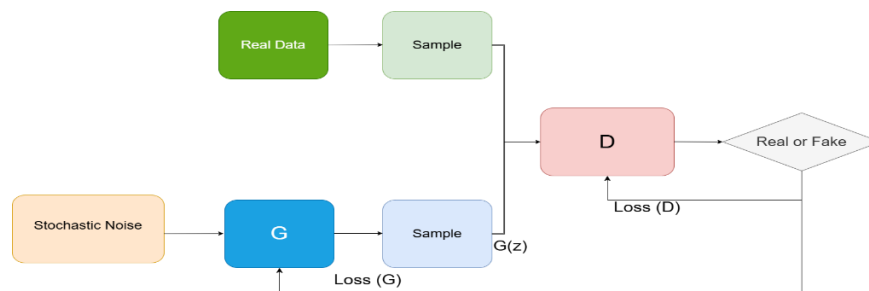
$$L_{total} = L_{GAN}(G, D_y + L_{GAN}(F, D_x) + L_{cyc}) \quad (8)$$

Fungsi total *loss* menggabungkan dua komponen GAN dan *cycle-consistency loss*, memastikan bahwa model belajar melakukan transformasi yang realistis dan reversibel meskipun tanpa pasangan data eksplisit (Y. Zhang dkk., 2022).

Image-to-Image Translation merupakan salah satu tugas penting dalam pengolahan citra berbasis *deep learning*. Pendekatan ini memungkinkan transformasi citra antar domain dengan mempertahankan semantik dan detail visual. Baik dalam skenario terawasi maupun tidak terawasi, I2I memainkan peran penting dalam berbagai aplikasi nyata, salah satunya adalah dalam restorasi citra. Keberhasilan metode ini sangat dipengaruhi oleh desain fungsi *loss* yang efisien dan stabilitas proses pelatihan.

2.2.5 Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) adalah salah satu arsitektur jaringan saraf generatif yang diperkenalkan sebagai pendekatan inovatif untuk menghasilkan data baru yang menyerupai distribusi data asli. Menurut Goodfellow *dkk.*, (2020), GAN terdiri dari dua jaringan yang saling berkompetisi yaitu *Generator* yang menciptakan data sintetis dan *Discriminator* yang membedakan antara data nyata dan buatan. Keduanya dilatih secara simultan dalam kerangka permainan dua pemain, yang memungkinkan *Generator* menghasilkan data yang semakin realistis. Dalam *deep learning*, GAN menjadi komponen penting untuk pembelajaran representasi kompleks, khususnya dalam tugas *image-to-image translation*, yaitu proses mentransformasikan citra dari satu domain ke domain lain. Model seperti pix2pix memanfaatkan arsitektur GAN untuk mempelajari pemetaan langsung antar domain citra secara *end-to-end*.



Gambar 2. 1 Arsitektur GAN

Pada gambar 2.1 merupakan arsitektur dari *Generative Adversarial Networks* (GAN) di atas, input berupa *noise* acak dari ruang laten $z \sim p_z(z)$ diubah oleh *Generator* menjadi data sintetis $G(x)$. *Discriminator* kemudian menerima dua jenis data yaitu data asli $x \sim p_{data}(x)$ dan data hasil *Generator*, lalu memberikan probabilitas apakah data tersebut asli atau palsu. Proses ini berlangsung secara iteratif dan membentuk kompetisi yang saling meningkatkan performa kedua jaringan.

Tujuan pelatihan GAN diformulasikan sebagai permainan *minimax* antara *Generator* (G) dan *Discriminator* (D), dengan fungsi objektif sebagai berikut:

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log (1 - D(G(z)))] \quad (9)$$

disini:

- $D(x)$ adalah probabilitas bahwa input x merupakan data asli.
- $G(z)$ adalah hasil dari *Generator* berdasarkan *noise* z .
- p_{data} adalah distribusi data nyata.
- $p_z(z)$ adalah distribusi *noise* acak (umumnya *Gaussian* atau *Uniform*).

Generator berusaha meminimalkan nilai fungsi di atas dengan cara membuat *Discriminator* tidak mampu membedakan hasil buaatannya. Sebaliknya, *Discriminator* berusaha memaksimalkan fungsi tersebut dengan membedakan secara akurat antara data asli dan palsu.

Adapun beberapa komponen dari *Generative Adversarial Networks* (GAN) yaitu *Generator*, *Discriminator*, dan beberapa komponen lainnya dapat dijelaskan sebagai berikut:

1. *Generator*

Generator adalah jaringan saraf yang memetakan *noise* acak dari ruang laten ke ruang data sintetis. Tujuannya adalah menghasilkan data $G(z)$ yang secara statistik dan visual menyerupai data nyata. Arsitektur *Generator* biasanya terdiri dari beberapa lapisan *fully connected* atau *convolutional transpose* layer tergantung pada jenis data yang digunakan. Fungsi kerugian

Generator sering ditulis sebagai:

$$L_G = -E_{z \sim p_z(z)} [\log D(G(z))] \quad (10)$$

Generator dikenai penalti jika *Discriminator* dengan mudah mengenali bahwa $G(z)$ adalah palsu. Oleh karena itu, semakin andal *Discriminator*, semakin besar tantangan bagi *Generator* untuk menghasilkan data yang realistis.

2. *Discriminator*

Discriminator berperan sebagai klasifikator biner yang menerima input berupa data nyata x maupun data sintetis $G(z)$, dan mengoutput-kan probabilitas bahwa input tersebut asli. Arsitekturnya biasanya terdiri dari lapisan *convolutional* dan *fully connected*, tergantung pada kompleksitas data. Fungsi kerugian *Discriminator* dituliskan sebagai:

$$L_D = -E_{x \sim p_{data}} [\log D(x)] - E_{z \sim p_z} [\log (1 - D(G(z)))] \quad (11)$$

dengan fungsi ini, *Discriminator* akan belajar meningkatkan kemampuan membedakan input palsu dan nyata, yang secara tidak langsung juga mendorong *Generator* untuk lebih baik.

3. Komponen Tambahan

Selain dua jaringan utama, beberapa komponen tambahan penting dalam GAN meliputi:

- a. Ruang Laten (*Latent Space*): Merupakan vektor acak berdimensi rendah yang menjadi masukan bagi *Generator*. Variasi pada ruang laten berpengaruh besar terhadap keanekaragaman dan kualitas data sintetis yang dihasilkan. Latent space biasanya mengikuti distribusi normal standar atau distribusi seragam.
- b. Fungsi Regularisasi dan Varian GAN: Dalam praktiknya, pelatihan GAN sering kali menghadapi masalah seperti ketidakstabilan gradien dan *mode collapse*. Oleh karena itu, varian GAN seperti *Wasserstein GAN* (WGAN), *Least Squares GAN* (LSGAN), dan WGAN-GP diperkenalkan

dengan fungsi kerugian dan teknik regularisasi baru. Misalnya, WGAN-GP menambahkan penalti gradien sebagai berikut:

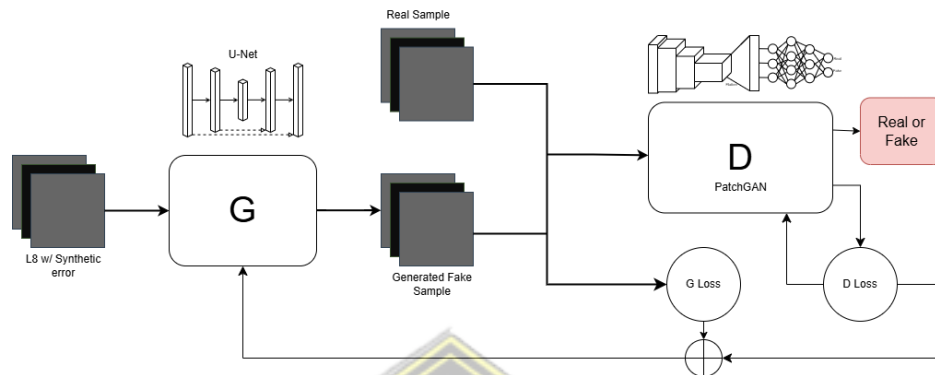
$$L_{GP} = \lambda E_{x' \sim p_{x'}} [(\|\nabla_{x'} D(x')\|_2 - 1)^2] \quad (12)$$

penalti ini menjaga agar fungsi *Discriminator* memenuhi syarat *Lipschitz continuity* dan membantu stabilisasi pelatihan (Cao dkk., 2023).

Secara keseluruhan, *Generative Adversarial Network* (GAN) merupakan pendekatan revolusioner dalam pembelajaran mesin yang memungkinkan pembangkitan data sintetis berkualitas tinggi melalui proses pelatihan dua jaringan saraf yang saling berkompetisi. Dengan *Generator* yang berfokus pada penciptaan data baru dan *Discriminator* yang berperan sebagai pengevaluasi, GAN dapat menghasilkan data yang semakin realistis seiring proses pelatihan. Pendekatan ini telah banyak diadopsi dalam berbagai aplikasi seperti pemrosesan citra, peningkatan resolusi, dan restorasi citra, serta menjadi dasar bagi banyak arsitektur generatif modern.

Meski memiliki potensi besar, GAN juga dihadapkan pada tantangan seperti ketidakstabilan pelatihan dan *mode collapse*. Namun, berbagai perbaikan telah dikembangkan untuk mengatasi hal tersebut, salah satunya melalui varian arsitektur dan teknik regularisasi. Dalam konteks tugas akhir ini, GAN dipilih sebagai dasar metode karena kemampuannya dalam mempelajari dan merekonstruksi pola visual kompleks, sehingga sangat sesuai untuk tugas restorasi citra digital akibat kerusakan. Pemahaman teoritis yang kuat terhadap GAN menjadi bekal penting dalam merancang dan mengimplementasikan model generatif yang efektif di bab-bab selanjutnya.

2.2.6 Pix2Pix Sebagai Model



Gambar 2. 2 Arsitektur pix2pix (Adiyaman dkk., 2024)

Pix2pix merupakan salah satu varian model dari *Generative Adversarial Networks* (GAN) berbasis metode *supervised* yang banyak digunakan dalam bidang *image-to-image translation* (I2I). Model ini dirancang untuk melakukan konversi citra antar dua domain dengan memanfaatkan data yang tersusun dalam pasangan (*paired Dataset*) seperti yang dapat dilihat pada gambar 2.2. Konsep yang diusung oleh pix2pix mewakili pendekatan *end-to-end* dalam pembelajaran pemetaan langsung dari citra masukan ke citra target. Setiap model I2I *supervised* yang bertujuan mentransformasikan citra dari satu domain ke domain lain dapat mengadopsi struktur dasar yang diperkenalkan oleh pix2pix (Pang dkk., 2022). Dalam implementasinya, pix2pix melibatkan dua jaringan saraf utama yang dilatih secara bersamaan dan bersifat kompetitif, yaitu *Generator* dan *Discriminator*. *Generator* bertugas menghasilkan citra sintetis yang menyerupai citra target, sementara *Discriminator* bertindak sebagai penilai yang memverifikasi keaslian citra tersebut dengan membandingkannya terhadap citra referensi.

Arsitektur *Generator* pada pix2pix mengadopsi struktur *U-net*, yang terdiri dari dua bagian utama yaitu *encoder* dan *decoder*, serta dilengkapi dengan *skip-connections*. *Encoder* berfungsi mengidentifikasi dan mengekstraksi fitur penting dari citra masukan melalui proses down-sampling, sedangkan *decoder* bertanggung jawab merekonstruksi citra keluarannya melalui proses up-

sampling. Keberadaan *skip-connections* membantu mempertahankan informasi spasial dari citra resolusi tinggi agar hasil yang dihasilkan tetap memiliki ketajaman dan detail yang akurat. Untuk menilai hasil dari *Generator*, *Discriminator* menggunakan pendekatan PatchGAN, yang mengevaluasi keaslian citra berdasarkan area-area kecil secara lokal. Selain itu, untuk menjaga agar struktur dan warna hasil citra tetap menyerupai referensinya, pix2pix menggunakan fungsi kerugian *L1 loss*, yang berperan dalam menjaga kesesuaian visual antara citra hasil dan target (M. Liu dkk., 2023).

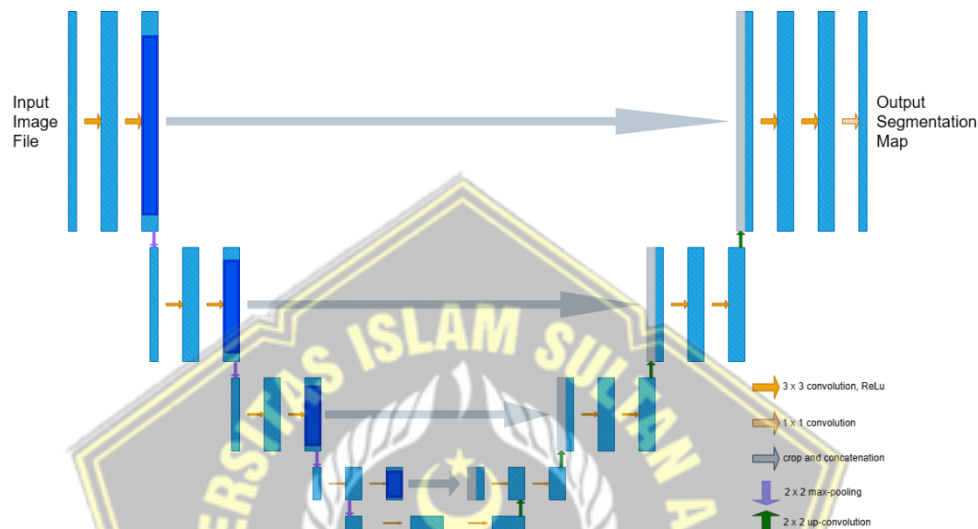
Pix2pix dapat diterapkan dalam proses restorasi citra yang mengalami kerusakan digital, dengan melatih model agar mampu mengubah citra rusak menjadi versi yang telah diperbaiki. Pelatihan dilakukan dengan pendekatan *supervised* menggunakan pasangan data berupa citra yang rusak sebagai input dan citra asli sebagai target. Adapun tahapan umum dalam proses restorasi citra menggunakan pix2pix meliputi:

1. Penyusunan *Dataset* berpasangan yang terdiri dari citra rusak dan citra asli.
2. Proses ekstraksi fitur oleh *encoder* yang mengambil informasi penting meski terdapat kerusakan.
3. Restorasi citra oleh *decoder*, yang diperkuat dengan *skip-connections* untuk menjaga detail visual.
4. Evaluasi hasil restorasi oleh *Discriminator* berbasis PatchGAN yang memeriksa keaslian citra secara lokal.
5. Optimisasi jaringan menggunakan kombinasi *L1 loss* dan *adversarial loss* guna meningkatkan kualitas dan realisme citra hasil.
6. Menghasilkan citra yang telah diperbaiki, di mana model mampu memperbaiki gangguan visual seperti coretan digital, bagian yang hilang, maupun distorsi sehingga menghasilkan citra yang mendekati bentuk aslinya.

Dengan urutan proses tersebut, pix2pix terbukti efektif dalam melakukan perbaikan citra digital yang rusak, karena kemampuannya dalam mempelajari

representasi visual secara langsung dan menghasilkan keluaran yang akurat baik secara semantik maupun tampilan visual.

2.2.7 Generator (U-Net)



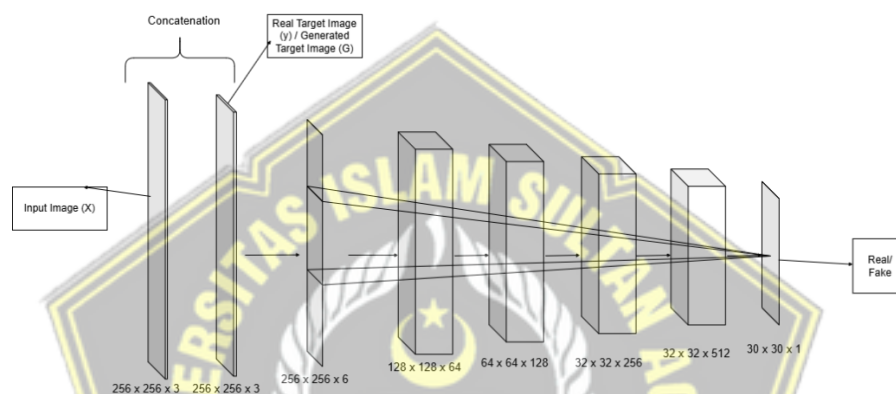
Gambar 2. 3 Arsitektur *Generator* (U-Net) (Weng and Zhu, 2021)

U-net adalah arsitektur jaringan yang secara khusus dirancang untuk segmentasi citra, proses pemisahan objek atau klasifikasi setiap *pixel* dalam citra ke dalam beberapa kelas seperti yang ditunjukkan pada Gambar 2.3. Teknik ini didasarkan pada *Convolutional Neural Network* (CNN), dengan berfokus pada ekstraksi fitur spasial dari citra dengan menggunakan lapisan-lapisan konvolusi. CNN biasanya efektif dalam mendeteksi fitur penting dalam citra, seperti tepi atau tekstur, melalui penggunaan kernel yang bergerak di atas citra *input* (Siddique dkk., 2021). *U-net* menggunakan struktur simetris berbentuk "U" yang terdiri dari "*contracting path*" dan "*expanding path*".

1. *Contracting path* bertujuan untuk mengecilkan ukuran citra sambil menangkap fitur.
2. *Expanding path* memperluas citra kembali ke ukuran asli sambil menggabungkan informasi fitur dari lapisan-lapisan sebelumnya.

Pendekatan ini memungkinkan *U-net* untuk mempertahankan resolusi tinggi pada hasil segmentasi untuk mendeteksi detail kecil dan mempertahankan informasi spasial. Dalam praktiknya, *U-net* memiliki kemampuan untuk menyesuaikan segmentasi terhadap struktur yang kompleks dan berbeda-beda.

2.2.8 Discriminator (PatchGAN)



Gambar 2. 4 Arsitektur *Discriminator* (PatchGAN) (Kushwaha dkk., 2023)

PatchGAN merupakan varian dari *Discriminator* dalam arsitektur GAN yang mengevaluasi realisme citra berdasarkan area lokal (patch kecil) daripada keseluruhan citra yang dapat dilihat pada gambar 2.4. Pendekatan ini terbukti efektif untuk menilai dan meningkatkan kualitas tekstur serta ketajaman detail pada hasil yang dihasilkan oleh *Generator*. Dengan memfokuskan evaluasi pada bagian kecil dari citra, PatchGAN mendorong hasil keluaran yang lebih realistis secara lokal.

Dalam struktur dasarnya, PatchGAN menerima dua buah citra, yakni citra input dari domain sumber dan citra hasil prediksi *Generator*. Kedua citra ini digabungkan secara kanal (*channel-wise*) dan menjadi input bagi jaringan *Discriminator*. Selanjutnya, input ini diproses melalui lima lapisan konvolusi berturut-turut. Tiga lapisan pertama memiliki kernel berukuran 4×4 dengan *stride* 2 dan *padding* 1, sementara dua lapisan terakhir menggunakan *stride* 1 agar

dimensi spasial tetap. Setiap lapisan diikuti oleh fungsi aktivasi Leaky ReLU dengan nilai *negative slope* sebesar 0,2, dan lapisan akhir menggunakan fungsi aktivasi sigmoid untuk menghasilkan nilai probabilitas antara 0 dan 1.

Seperti dijelaskan oleh Kushwaha *dkk.*, (2023), konfigurasi arsitektur ini menghasilkan output dalam bentuk grid dengan dimensi sekitar 16×16 untuk input berukuran 256×256 . Masing-masing elemen dalam *grid* tersebut mewakili klasifikasi terhadap satu patch citra lokal dengan *receptive field* sekitar 70×70 *pixel*. Hal ini memungkinkan PatchGAN untuk melakukan evaluasi yang lebih spesifik terhadap detail citra, dibandingkan *Discriminator* konvensional yang hanya menghasilkan satu nilai global.

Keunggulan utama PatchGAN terletak pada efisiensinya serta kemampuannya menghasilkan hasil yang tajam. Tanpa penggunaan lapisan *fully-connected*, model menjadi lebih ringan dan cepat dalam pelatihan. Selain itu, karena tidak menggunakan pooling, hubungan spasial antar *pixel* tetap terjaga, yang sangat penting dalam tugas-tugas seperti pewarnaan citra, restorasi, dan segmentasi.

2.2.9 Parameter

Dalam *Image-to-Image Translation* untuk restorasi citra berbasis GAN, penentuan parameter kunci sangat memengaruhi performa model. Berikut ringkasan parameter utama tanpa menampilkan rumus lengkap:

1. Fungsi *Loss* dan Bobot Utama

Loss yang biasanya digunakan adalah total *loss* yaitu gabungan antara *adversarial loss* dan *pixel-wise loss* (misal L1) dengan satu bobot utama (λ). Nilai $\lambda = 10$ umum dipakai untuk menjaga keseimbangan antara keaslian tekstur dan struktur keseluruhan citra (Ren *dkk.*, 2024).

2. *Optimizer* dan Pengaturan *Learning Rate*

Adam dengan $\beta_1 = 0.5$ dan $\beta_2 = 0.999$ meningkatkan stabilitas pada pelatihan GAN, sedangkan *learning rate* biasanya diatur 2×10^{-4} hingga 1×10^{-4} ,

terkadang menggunakan *decay* setelah *epoch* tertentu untuk haluskan konvergensi (Hoyez dkk., 2022).

3. Arsitektur Lapisan dan Dimensi *Feature*

Arsitektur lapisan dan dimensi *feature* yang digunakan adalah sebagai berikut:

- a. *Generator* dengan *encoder-decoder* 4–6 level dan *skip connections* untuk mempertahankan detail. Filter awal 64, kernel size 3×3 , stride 2.
- b. *Discriminator* dengan 5–6 lapis *convolutional*, tanpa *fully connected* layer.

4. *Batch Size* dan *Epoch*

Batch size biasanya 1–4 untuk input resolusi tinggi, hingga 16 jika memori GPU mencukupi. Jumlah *epoch* yang umum digunakan dalam pelatihan adalah 100.

5. Regularisasi dan Normalisasi

Dropout (rate 0.3 – 0.5) digunakan di beberapa layer *decoder* untuk mengurangi *overfitting*, sedangkan *instance normalization* mempermudah adaptasi gaya citra, dan *spectral normalization* atau *gradient penalty* ditambahkan di *Discriminator* untuk stabilitas tambahan.

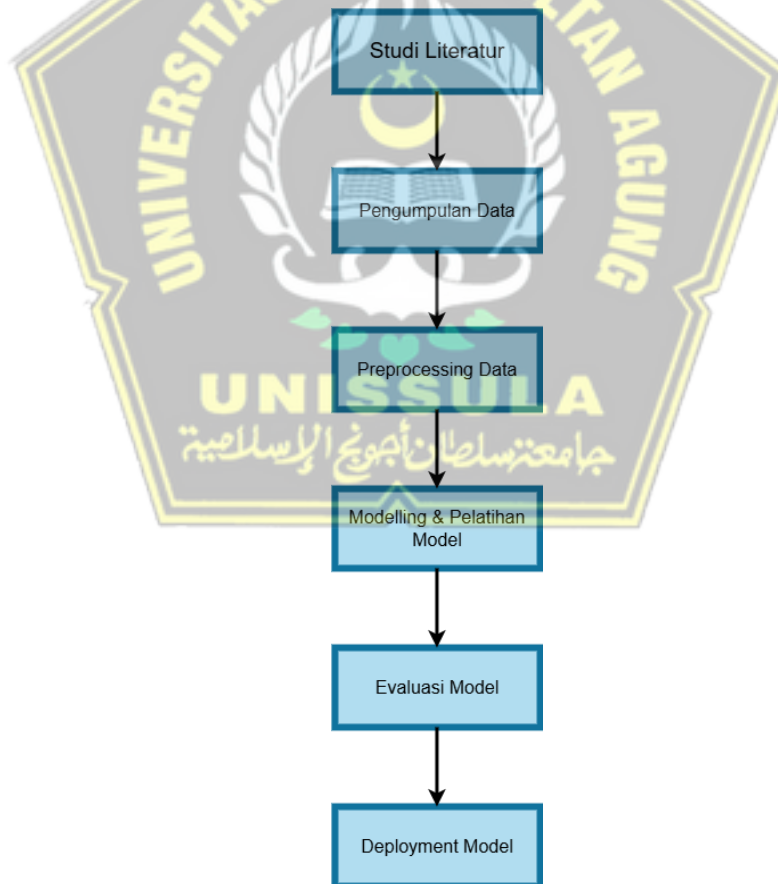
Parameter yang telah disebutkan sebelumnya masih dapat disesuaikan lebih lanjut dengan memperhatikan karakteristik khusus dari dataset serta tujuan utama proses restorasi. Penyesuaian ini bertujuan untuk menghasilkan keluaran yang tampak lebih alami dan realistis, sekaligus mengurangi kemungkinan munculnya artefak atau gangguan visual lainnya. Melalui tuning parameter yang optimal, kualitas hasil restorasi dapat ditingkatkan secara signifikan, baik dari sisi ketajaman detail, keselarasan tekstur, maupun keseluruhan tampilan visual, sehingga mampu memenuhi standar estetika dan teknis yang diharapkan.

BAB III

METODOLOGI PENELITIAN

3.1 Metode Penelitian

Pada tahap ini dilakukan serangkaian proses untuk membangun sistem restorasi citra berbasis *deep learning* dengan model GAN yang terdiri dari *Generator* berarsitektur *U-net* dan *Discriminator* PatchGAN. Sistem ini dirancang untuk memproses citra yang mengalami kerusakan dan memulihkannya agar menyerupai kondisi aslinya. Tahapan meliputi kajian literatur, pengumpulan *Dataset*, *preprocessing*, pelatihan model, evaluasi hasil, dan *deployment* model yang semuanya terdapat di *flowchart* pada gambar 3.1:



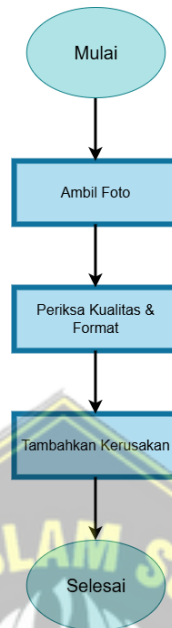
Gambar 3. 1 Tahap Penelitian

3.1.1 Studi Literatur

Pada tahap ini dilakukan pencarian dan telaah literatur terkait metode restorasi citra digital dengan fokus pada teknologi *deep learning*, terutama yang menggunakan pendekatan *Generative Adversarial Network* (GAN) dengan pix2pix yang akan dijadikan sebagai *base model*. Tujuannya adalah memperoleh pemahaman yang mendalam tentang arsitektur jaringan, strategi pelatihan, jenis kerusakan yang umum ditangani, serta metrik evaluasi yang relevan untuk mengukur performa model. Studi ini menjadi dasar dalam memilih pendekatan dan teknologi yang akan diimplementasikan.

3.1.2 Pengumpulan Data

Dataset utama yang digunakan dalam penelitian ini adalah CelebA-HQ *Resized*, yaitu kumpulan citra wajah manusia dengan resolusi 256×256 *pixel* yang telah melalui proses prapemrosesan dan standarisasi kualitas. *Dataset* ini dipilih karena mengandung variasi ekspresi, pencahayaan, dan orientasi wajah yang cukup kaya, sehingga sangat sesuai untuk melatih model restorasi yang perlu menggeneralisasi beragam kondisi visual. Untuk kebutuhan pelatihan model, citra-citra dari *Dataset* ini dimodifikasi dengan menyisipkan kerusakan buatan (simulatif) menggunakan beberapa teknik berbeda, agar sistem dapat belajar mengoreksi berbagai jenis degradasi citra.



Gambar 3. 2 *Flowchart* alur pengumpulan data

Pada gambar 3. 2 adalah *flowchart* proses pengumpulan data yang akan digunakan untuk dataset sistem restorasi citra. Dimulai dari mencari dataset citra dari *platform* Kaggle, kemudian dilanjutkan memeriksa kualitas dan format pada dataset, dan terakhir ada menambahkan kerusakan pada citra.

3.1.3 *Preprocessing* Data

Preprocessing data dilakukan untuk memastikan seluruh citra memiliki format yang seragam dan siap digunakan dalam proses pelatihan model. Dataset yang digunakan dalam penelitian ini adalah CelebA-HQ Resized dengan resolusi 256×256 *pixel*, yang terdiri dari pasangan citra input (yang telah mengalami kerusakan digital) dan citra target (citra asli sebagai referensi pemulihan). Setiap citra diubah menjadi representasi numerik agar dapat diolah oleh model, baik dalam format RGB maupun grayscale sesuai kebutuhan.

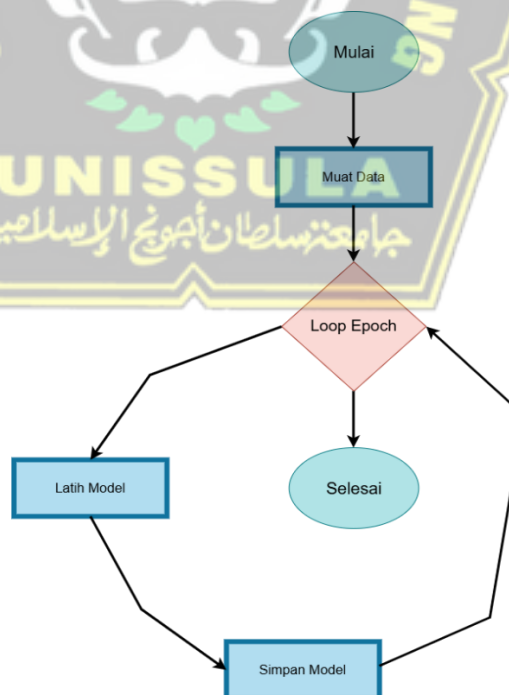
Nilai *pixel* pada setiap citra dinormalisasi ke dalam rentang $[0, 1]$ guna menjaga kestabilan pelatihan dan memudahkan model dalam mempelajari pola-pola visual yang ada. Selain itu, setiap citra diperiksa untuk memastikan ukuran dan jumlah saluran warnanya sesuai dengan yang diharapkan. Citra yang tidak

memenuhi kriteria akan dilewati agar tidak mengganggu proses pelatihan. Untuk mencegah kelebihan beban memori, jumlah citra yang digunakan dibatasi hingga maksimum 3000 gambar.

Setelah proses normalisasi dan validasi selesai, seluruh dataset diorganisasikan ke dalam batch berukuran 8 citra per iterasi. Pengelompokan dalam batch ini dilakukan untuk meningkatkan efisiensi pelatihan dan menjaga kestabilan proses optimisasi. Dengan preprocessing yang konsisten dan sistematis ini, data yang digunakan dalam penelitian dipastikan siap dan optimal untuk mendukung pelatihan model Generative Adversarial Network (GAN).

3.1.4 Pelatihan Model

Tahap ini meliputi proses pelatihan model GAN dengan arsitektur pix2pix. Proses dimulai dari pemodelan, kemudian dilakukan pelatihan selama beberapa epoch dengan optimasi bobot melalui perhitungan fungsi *loss*. Setelah pelatihan selesai, model disimpan untuk digunakan pada tahap berikutnya. Ilustrasi proses ini ditunjukkan pada gambar 3.3.



Gambar 3. 3 Proses *Training* GAN

1. Pemodelan

Model yang dibangun dalam penelitian ini adalah pix2pix sebagai base model yang terdiri dari dua komponen utama, yaitu *Generator (U-net)* dan *Discriminator (PatchGAN)*, yang keduanya diatur untuk saling berkompetisi dalam kerangka kerja *Generative Adversarial Network (GAN)*:

a. *Generator (U-net)*

Generator dikembangkan dengan arsitektur *U-net*, yang memiliki struktur *encoder-decoder* berlapis-lapis. Bagian *encoder* terdiri dari delapan blok downsampling yang bertugas mengekstraksi fitur spasial dari citra input secara bertahap, sementara bagian *decoder* berisi tujuh blok *upsampling* yang merekonstruksi citra ke resolusi aslinya. *Skip connection* dipasang secara strategis antara layer di *encoder* dan *decoder* untuk mempertahankan detail penting dari citra input selama proses transformasi. *Blok output Generator* dirancang untuk menghasilkan citra dengan tiga kanal warna (RGB), agar hasilnya dapat dibandingkan langsung dengan citra *ground truth*.

b. *Discriminator (PatchGAN)*

Discriminator dibangun berdasarkan konsep PatchGAN, yang mengevaluasi realisme citra dengan membaginya menjadi beberapa patch berukuran kecil, kemudian menentukan probabilitas apakah masing-masing patch adalah asli atau palsu. Arsitektur *Discriminator* terdiri dari serangkaian layer konvolusional yang menangani gabungan antara citra input (yang rusak) dan citra hasil prediksi (atau citra asli). *Discriminator* menghasilkan keluaran berupa peta nilai dua dimensi yang merepresentasikan tingkat keaslian di setiap area lokal citra.

2. Fungsi *Loss* dan *Optimizer*

Pelatihan model ini melibatkan dua jenis fungsi *loss* utama:

- a. *Adversarial Loss*, yang bertugas memandu *Generator* agar mampu menghasilkan citra yang semakin mirip dengan citra asli dan mampu menipu *Discriminator*. Teknik label *smoothing* diterapkan untuk menstabilkan proses pelatihan dengan sedikit memodifikasi nilai target dari label "real".
- b. *L1 Loss*, yang menghitung rata-rata selisih absolut antara citra hasil prediksi dan citra *ground truth*. *Loss* ini berperan penting dalam memastikan citra hasil prediksi tetap setia secara struktural dan visual terhadap citra asli.

Model dioptimalkan menggunakan algoritma Adam, dengan penyesuaian laju pembelajaran yang berbeda antara *Generator* dan *Discriminator*. Proses pelatihan dilakukan selama 100 epoch, dengan strategi *update* di mana setiap *batch* melalui satu kali pembaruan *Discriminator*, diikuti oleh tiga kali pembaruan *Generator*, guna mendorong *Generator* untuk berkembang lebih cepat dalam menghadapi tantangan yang diberikan oleh *Discriminator*. *Checkpoint* model disimpan secara berkala setiap 5 - 20 epoch untuk memastikan hasil pelatihan dapat dipantau dan dipulihkan jika diperlukan.

3.1.5 Evaluasi Model

Evaluasi dilakukan melalui dua pendekatan utama:

1. Evaluasi Kuantitatif: menggunakan metrik PSNR (*Peak Signal-to-Noise Ratio*) dan SSIM (*Structural Similarity Index Measure*), yang masing-masing mengukur seberapa dekat nilai *pixel* prediksi terhadap *ground truth*, dan seberapa mirip struktur dan tekstur antar citra. Rata-rata nilai PSNR dan SSIM dihitung untuk seluruh batch setiap epoch, dan ditampilkan dalam bentuk grafik untuk memantau kemajuan pelatihan.

- a. PSNR (*Peak Signal-to-Noise Ratio*)

$$PSNR = 10 \log_{10} \left(\frac{MAX^2}{MSE} \right) \quad (13)$$

di mana MAX adalah nilai maksimum *pixel* dalam citra (biasanya 255 untuk citra 8-bit). Semakin tinggi nilai PSNR, semakin kecil kesalahan atau distorsi pada citra hasil terhadap citra aslinya. PSNR yang bernilai di atas 30 dB umumnya dianggap menunjukkan kualitas visual yang baik (Susanto *dkk.*, 2024).

- b. SSIM (*Structural Similarity Index Measure*)

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (14)$$

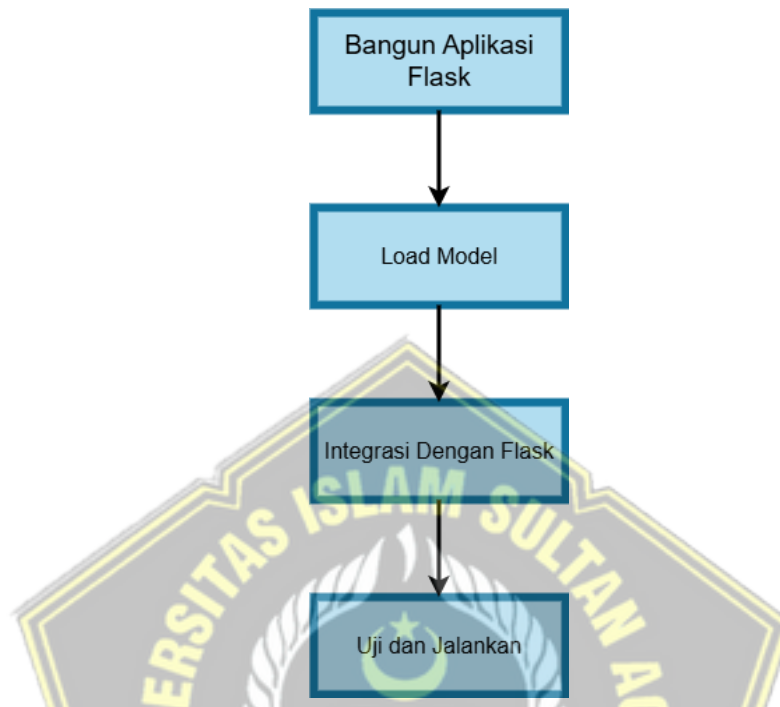
Keterangan:

- μ_x, μ_y : rata-rata intensitas *pixel* citra x dan y.
- σ_x^2, σ_y^2 : variansi dari masing-masing citra.
- σ_{xy} : kovariansi antara dua citra.
- C_1, C_2 : konstanta untuk menstabilkan perhitungan.

Nilai SSIM berada dalam rentang -1 hingga 1, dengan nilai 1 menandakan dua citra sangat identik secara struktural (Mudeng *dkk.*, 2022).

2. Evaluasi Kualitatif: dilakukan dengan menampilkan perbandingan visual antara citra rusak dan citra hasil restorasi. Visualisasi ini digunakan untuk mengamati kualitas restorasi secara langsung oleh pengamat manusia, serta membantu mengidentifikasi potensi artefak atau kesalahan prediksi yang tidak terdeteksi oleh metrik kuantitatif.

3.1.6 Deployment Model



Gambar 3. 4 *Flowchart deployment*

Gambar 3.4 adalah *flowchart* untuk proses *deployment* yang merupakan proses pembuatan *user interface* berupa web. Setelah model selesai dilatih dan dievaluasi, model *Generator* diekspor ke dalam format .h5 (HDFS). Model ini kemudian diintegrasikan ke dalam aplikasi berbasis *Flask* dengan endpoint yang memungkinkan pengguna mengunggah file citra untuk diproses. Setiap citra yang diterima akan divalidasi dan disimpan sementara, lalu diproses melalui tahapan *preprocessing* seperti konversi ke *grayscale* dan normalisasi agar sesuai dengan input model. Model kemudian memproses citra tersebut untuk menghasilkan citra yang telah direstorasi. Hasil output dikonversi kembali ke format citra standar sebelum dikembalikan kepada pengguna sebagai hasil akhir. Sistem *deployment* dilengkapi dengan logging untuk mencatat setiap aktivitas dan penanganan error untuk menjaga performa aplikasi tetap stabil

3.2 Analisa Kebutuhan

Pada tahap ini penulis menganalisa apa saja kebutuhan selama pembangunan sistem, seperti software, tools, bahasa pemrograman, dan *library* yang digunakan. Berikut adalah apa saja yang digunakan dalam pembangunan sistem :

1. Bahasa Pemrograman

a. Python

Python merupakan bahasa pemrograman utama yang digunakan dalam pengembangan sistem ini. Python dipilih karena memiliki sintaks yang sederhana dan didukung oleh berbagai pustaka *Deep Learning* yang sangat powerful. Dengan fitur-fitur seperti kemudahan dalam pemrograman berorientasi objek, dukungan untuk pengolahan data, serta kemampuan untuk bekerja dengan data besar, Python adalah bahasa yang sangat ideal untuk pengembangan sistem berbasis *Deep Learning*.

2. Perangkat Lunak (*Software*)

a. *Google Colab*

Google Colaboratory (Colab) dimanfaatkan sebagai lingkungan pemrograman interaktif yang berbasis *cloud* untuk menulis, menjalankan, dan menguji kode secara langsung. Platform ini sangat mendukung proses pengembangan dan eksperimen dalam penelitian karena memungkinkan integrasi dengan berbagai pustaka Python serta penyimpanan di *Google Drive*. Dengan antarmuka yang ramah pengguna dan dukungan terhadap eksekusi berbasis GPU, Colab mempermudah peneliti dalam memodifikasi kode, mencatat hasil eksperimen, serta mendokumentasikan proses pengujian secara terstruktur dalam satu dokumen yang interaktif.

b. *Visual Studio Code*

Visual Studio Code dipilih sebagai text editor pada pengembangan aplikasi dalam penelitian ini, *Visual Studio Code* dipilih dikarenakan mendukung banyak bahasa pemrograman dan *framework*, *multi platform*, performa yang sangat cepat, mempunyai banyak *extensions* yang dapat mempermudah proses pengembangan website.

c. Figma

Figma adalah *tools* untuk memfasilitasi *designer* UI dan UX yang berbasis website yang digunakan untuk mendesain sebuah aplikasi. Dalam penelitian ini digunakan untuk membuat *design user interface* pada sistem restorasi citra.

3. *Library dan Framework*

a. *TensorFlow*

TensorFlow adalah *framework open-source* yang digunakan untuk membangun, melatih, dan menerapkan model *Deep Learning*. *TensorFlow* dipilih karena efisiensinya dalam menangani perhitungan matriks dan operasi tensor, serta kemampuan untuk melakukan pelatihan model menggunakan CPU/GPU.

b. OpenCV

OpenCV (*Open Source Computer Vision Library*), yang diakses melalui modul *cv2* pada Python, digunakan sebagai pustaka utama untuk pengolahan citra digital dalam penelitian ini. OpenCV menyediakan berbagai fungsi dan algoritma yang efisien untuk menangani tugas-tugas seperti pembacaan citra, konversi format warna, deteksi tepi, segmentasi objek, serta transformasi spasial. Dengan dukungan terhadap berbagai format file dan kemampuannya dalam pemrosesan *real-time*, OpenCV sangat membantu dalam tahap praproses data citra sebelum digunakan dalam pelatihan dan pengujian model *deep learning*. Fleksibilitas dan kemudahan integrasinya dengan

pustaka lain seperti NumPy dan TensorFlow menjadikannya pilihan ideal dalam pengembangan sistem berbasis visi computer.

c. *NumPy*

NumPy digunakan untuk manipulasi *array* dan matriks numerik. Pustaka ini sangat penting untuk pengolahan data numerik, yang sering digunakan dalam model *Deep Learning* untuk pemrosesan *batch* data.

d. *Matplotlib*

Matplotlib digunakan untuk visualisasi hasil eksperimen dan model. Visualisasi ini sangat penting untuk menganalisis proses pelatihan, seperti mengcitra grafik *Loss* dan akurasi selama pelatihan model, serta untuk memverifikasi performa model.

e. *TQDM*

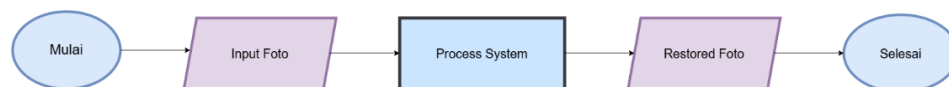
TQDM digunakan untuk menampilkan progress bar pada iterasi pelatihan model. Ini memudahkan pengembang untuk memonitor kemajuan pelatihan, terutama ketika bekerja dengan dataset yang besar.

f. *Time*

Modul *time* pada Python digunakan untuk mengukur durasi pelatihan model. Dengan mencatat waktu mulai dan selesai, peneliti dapat mengetahui total waktu yang dibutuhkan dalam proses pelatihan. Informasi ini berguna untuk mengevaluasi efisiensi dan membandingkan performa antar model atau konfigurasi.

3.3 Penggunaan Sistem

Pada tahap ini dilakukan analisa untuk menentukan alur kerja penggunaan sistem restorasi citra yang akan dilakukan oleh *user* dalam bentuk *flowchart*. Alur kerja sistem dapat dilihat pada citra 3.5.



Gambar 3. 5 *Flowchart* alur kerja sistem

Gambar 3. 5 adalah *flowchart* restorasi citra yang mengcititakan proses sistem saat digunakan oleh *user*. Pada citra 3. 5 memiliki beberapa tahapan, sebagai berikut :

1. *User* membuka aplikasi kemudian *user* akan melihat tampilan awal website restorasi citra.
2. *User* menekan tombol *choose file* untuk memasukkan citra rusak ke sistem.
3. Kemudian sistem akan memproses citra rusak tersebut dengan memperbaiki kerusakan menggunakan model yang sudah dilatih dan diintegrasikan dengan sistem.
4. *User* akan mendapatkan *output* dari sistem berupa citra yang sudah di restorasi dari citra rusak yang telah *user* masukkan ke sistem.

3.4 Perancangan *User Interface*

3.4.1 Halaman Awal Sistem



Gambar 3. 6 Tampilan halaman awal sistem

Gambar 3.6 menampilkan rancangan antarmuka pengguna (*user interface*) pada halaman awal sistem restorasi citra. Pada halaman ini, pengguna diminta

untuk mengunggah citra yang akan direstorasi dengan menekan tombol *Choose File*, yang kemudian akan membuka direktori penyimpanan lokal untuk memilih berkas citra. Setelah citra berhasil diunggah, sistem secara otomatis akan memproses citra tersebut menggunakan model *Generative Adversarial Networks* (GANs) dengan arsitektur *pix2pix*, dan menghasilkan keluaran berupa citra hasil restorasi. Proses ini dirancang agar berjalan secara otomatis tanpa memerlukan intervensi manual dari pengguna, sehingga meningkatkan efisiensi dan kemudahan dalam penggunaan sistem.

3.4.2 Tampilan Hasil Sistem



Gambar 3. 7 Tampilan hasil sistem

Gambar 3.7 menampilkan antarmuka sistem setelah pengguna berhasil mengunggah citra rusak dan menekan tombol *Restore Image* untuk memulai proses restorasi. Pada tahap ini, sistem secara otomatis menampilkan dua citra secara berdampingan, yaitu citra asli yang mengalami kerusakan dan hasil citra yang telah direstorasi menggunakan model *Generative Adversarial Networks*

(GANs) dengan arsitektur *pix2pix*. Tampilan ini memungkinkan pengguna untuk membandingkan kondisi sebelum dan sesudah proses restorasi secara visual, sehingga mempermudah dalam menilai kualitas hasil yang diperoleh. Selain itu, sistem menyediakan fitur *Download Restored Image* yang memungkinkan pengguna mengunduh citra hasil restorasi ke perangkat lokal, sehingga hasil pemrosesan dapat disimpan dan digunakan sesuai kebutuhan. Antarmuka ini dirancang untuk memberikan pengalaman pengguna yang intuitif dan efisien, serta mendukung alur kerja yang terotomatisasi dalam proses restorasi citra berbasis kecerdasan buatan.



BAB IV

HASIL DAN ANALISIS PENELITIAN

4.1 Hasil Pengumpulan Data



Gambar 4. 1 Dataset citra wajah asli dan rusak

Pada gambar 4.1 berikut menunjukkan hasil pengumpulan data yang digunakan dalam penelitian ini. Dataset yang digunakan adalah CelebA-HQ *resized* (256 x 256). Hasil pengumpulan data ini, citra-citra wajah dari dataset telah diberi berbagai bentuk kerusakan digital. Kerusakan ini tampak jelas pada citra dan mencerminkan kondisi nyata yang harus dihadapi oleh model dalam melakukan restorasi. Setiap bentuk kerusakan menghasilkan tantangan visual yang berbeda, yang akan berdampak pada kompleksitas proses restorasi oleh model GAN.

Adapun hasil penerapan kerusakan digital pada citra ditunjukkan dalam beberapa bentuk berikut:

1. Coretan Acak (*Scribble Noise*)

Terlihat adanya goresan acak berwarna putih yang melintang pada area wajah dan sekitarnya. Coretan ini menutupi sebagian informasi penting dari wajah seperti kontur pipi dan area mata, sehingga menyulitkan proses restorasi visual.

2. Distorsi Blok (*Block Distortion*)

Tampak beberapa bagian wajah bergeser secara tidak beraturan dalam bentuk blok. Hal ini menyebabkan ketidaksesuaian antara bagian wajah satu dengan yang lain, seperti pergeseran pada garis rambut atau struktur mata.

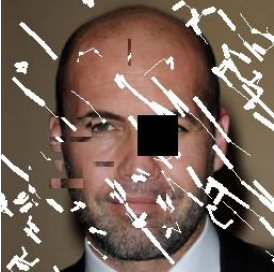
3. Pemotongan Area (*Region Masking*)

Pada jenis kerusakan ini, beberapa area pada citra terlihat sepenuhnya gelap (hitam), menutupi fitur penting wajah seperti mata, pipi, atau dahi. Hal ini menandakan adanya bagian citra yang benar-benar hilang dan harus direstorasi oleh model.

Citra-citra hasil pengumpulan data ini menunjukkan kondisi nyata dari input yang akan diproses oleh model. Citra asli yang utuh berperan sebagai acuan atau target, sedangkan citra yang telah rusak digunakan sebagai input pada proses pelatihan model GAN. Variasi jenis kerusakan pada dataset memberikan citraan bahwa model akan dilatih untuk menghadapi berbagai tantangan dalam proses restorasi, mulai dari gangguan ringan hingga kehilangan informasi yang parah.

Tingkat kerusakan pada citra dibagi menjadi tiga kategori berdasarkan letak dan sejauh mana kerusakan tersebut memengaruhi bagian penting dari wajah. Pembagian ini bertujuan untuk memberikan citraan tentang tingkat kompleksitas yang harus dihadapi model dalam proses restorasi. Klasifikasi lengkap mengenai tingkat kesulitan kerusakan dapat dilihat pada Tabel 4.1.

Table 4. 1 Dataset tipe citra rusak

Citra	Deskripsi
 Tipe 1 : Ringan/Mudah	Kerusakan yang terjadi pada bagian citra yang tidak terlalu penting atau tidak berpengaruh langsung pada fitur wajah utama. Biasanya berupa gangguan ringan pada latar belakang atau area sekitar wajah seperti rambut, pakaian, atau area kosong. Kerusakan ini tidak mengganggu struktur wajah secara signifikan sehingga model restorasi dapat memperbaikinya dengan relatif mudah dan hasil restorasi cenderung akurat dan konsisten.
 Tipe 2 : Sedang	Kerusakan yang muncul pada bagian tubuh yang berdekatan dengan wajah, seperti leher, pipi, atau bagian samping kepala, namun tidak mengenai fitur utama wajah seperti mata, hidung, atau mulut. Kerusakan ini cukup menantang karena mempengaruhi bagian citra yang berdekatan dengan fitur penting, sehingga model harus mampu menjaga konsistensi tekstur dan bentuk agar hasil restorasi terlihat alami dan tidak terjadi artefak visual.
 Tipe 3 : Berat/Sulit	Kerusakan yang menutupi atau menghilangkan bagian utama dari wajah seperti mata, hidung, mulut, atau area sekitar yang sangat menentukan ekspresi wajah dan identitas individu. Tingkat kesulitan restorasi sangat tinggi karena model harus mampu merestorasi detail yang sangat

Citra	Deskripsi
	kompleks dan kritikal, sekaligus mempertahankan kesesuaian dengan konteks wajah secara keseluruhan. Kesalahan kecil pada bagian ini dapat membuat hasil restorasi menjadi tidak realistis dan mengurangi kualitas visual secara signifikan..

Klasifikasi ini juga menjadi acuan dalam mengevaluasi performa model GAN pada bab selanjutnya. Restorasi pada kategori mudah biasanya tidak menimbulkan perubahan signifikan karena informasi utama wajah masih utuh. Namun, pada kategori sedang dan sulit, kemampuan model diuji untuk mengisi bagian yang hilang atau rusak dengan hasil yang realistis dan konsisten terhadap konteks wajah.

4.2 Hasil *Preprocessing* Data

Tahap *preprocessing* dilakukan sebelum citra dimasukkan ke dalam model, untuk memastikan konsistensi bentuk data, pengurangan *noise* yang tidak diinginkan, serta menyesuaikan citra agar nilai *pixel*-nya terstandarisasi. Dalam konteks restorasi citra, terutama dengan dataset yang mengandung kerusakan digital, *preprocessing* membantu memperjelas informasi penting yang akan dipelajari model untuk melakukan restorasi.

4.2.1 Implementasi Proses *Preprocessing*

Preprocessing dilakukan menggunakan bahasa pemrograman Python dengan bantuan pustaka OpenCV dan NumPy. Berikut adalah beberapa *preprocessing* yang digunakan adalah sebagai berikut:

1. Pengurutan File Secara Alfanumerik

Fungsi `sorted_alphanumeric` memastikan urutan file citra yang konsisten dan berurutan, agar tidak terjadi kekacauan saat proses batch atau

saat data diambil secara berulang.

```
=== Daftar file di normal_data sebelum pengurutan ===
['16464.jpg', '07112.jpg', '25659.jpg', '12283.jpg', '19882.jpg', '00559.jpg', '25190.jpg']

=== Daftar file di normal_data sesudah pengurutan alfanumerik ===
['00014.jpg', '00015.jpg', '00019.jpg', '00020.jpg', '00025.jpg', '00030.jpg', '00033.jpg']
```

Gambar 4. 2 Pengurutan normal_data

Pada Gambar 4.2 sebelum dilakukan pengurutan, file dalam folder normal_data muncul dalam urutan acak seperti ['16464.jpg', '07112.jpg', '25659.jpg', '12283.jpg']. Ini karena fungsi `os.listdir()` mengembalikan daftar file sesuai urutan sistem file, bukan berdasarkan urutan angka atau nama. Setelah diterapkan fungsi `sorted_alphanumeric`, daftar file diurutkan berdasarkan angka secara logis (bukan alfabetis biasa), sehingga hasilnya menjadi ['00014.jpg', '00015.jpg', '00019.jpg', '00020.jpg']. Pengurutan ini penting agar pemrosesan data (misalnya pencocokan antara gambar normal dan rusak) terjadi secara konsisten dan berurutan.

```
== Daftar file di rusak_data sebelum pengurutan ==
'07493_combined.jpg', '03163_combined.jpg', '27185_combined.jpg'

== Daftar file di rusak_data sesudah pengurutan alfanumerik ==
'00014_combined.jpg', '00015_combined.jpg', '00019_combined.jpg'
```

Gambar 4. 3 Pengurutan rusak data

Di folder rusak_data yang terdapat pada Gambar 4.3, nama file awal seperti ['07493_combined.jpg', '03163_combined.jpg', '27185_combined.jpg'] juga tidak tersusun secara numerik meskipun mengandung angka. Dengan menggunakan fungsi `sorted_alphanumeric`, file-file ini disusun berdasarkan nilai angka yang terkandung dalam nama file, menghasilkan urutan seperti ['00014_combined.jpg', '00015_combined.jpg', '00019_combined.jpg']. Ini sangat berguna terutama jika file rusak ingin dipasangkan atau dibandingkan dengan file dari normal_data, karena urutan numerik akan memudahkan pencocokan berdasarkan ID gambar.

2. Konversi Warna

Citra yang awalnya menggunakan format BGR (*default* OpenCV) dikonversi ke format RGB yang lebih umum dan sesuai untuk kebanyakan

model *deep learning*.

```
=== Piksel Sebelum Pemrosesan (BGR) ===
BGR (uint8): [129 151 203]
```

Gambar 4. 4 *Pixel BGR*

Pada Gambar 4.4 Nilai [129 151 203] menunjukkan nilai warna *pixel* pada koordinat (0,0) dalam format BGR (Blue, Green, Red). OpenCV secara default membaca gambar dalam urutan BGR, bukan RGB. Artinya, pada posisi tersebut, intensitas biru adalah 129, hijau 151, dan merah 203. Nilai ini masih dalam format 8-bit unsigned integer (uint8), yaitu rentang 0 hingga 255.

```
=== Piksel Setelah Konversi ke RGB ===
RGB (uint8): [203 151 129]
```

Gambar 4. 5 *Pixel RGB*

Setelah konversi dari BGR ke RGB, urutan nilai warna berubah menjadi [203 151 129] seperti yang terlihat pada Gambar 4.5, di mana nilai merah menjadi 203, hijau tetap 151, dan biru menjadi 129. Ini penting karena sebagian besar pustaka visualisasi (seperti matplotlib) menggunakan format RGB, bukan BGR. Nilai *pixel* tetap dalam bentuk uint8.

3. Konversi ke *Grayscale*

Jika parameter `is_grayscale=True`, citra dikonversi ke format *grayscale* dengan menambahkan satu dimensi sebagai channel tunggal, sehingga bentuk citra menjadi (256, 256, 1).

```
=== Piksel Setelah Konversi ke Grayscale ===
Grayscale (uint8): 164
```

Gambar 4. 6 *Pixel Grayscale*

Pada Gambar 4.6 hasil konversi ke *grayscale* menghasilkan nilai *pixel* tunggal 164. Ini merupakan hasil dari kombinasi linear dari saluran warna RGB untuk menghasilkan intensitas abu-abu, biasanya menggunakan

rumus luminance ($0.299R + 0.587G + 0.114B$). Nilai 164 berada dalam skala uint8, masih dalam rentang 0–255, dan mewakili kecerahan *pixel*.

4. Normalisasi Nilai *Pixel*

Seluruh *pixel* citra diubah nilainya menjadi rentang 0 hingga 1 dengan membagi nilai *pixel* dengan 255.0. Normalisasi ini penting agar pelatihan model menjadi lebih stabil dan konvergen lebih cepat, untuk contoh hasil samplanya dapat dilihat pada Gambar 4.7 dan Gambar 4.8.

```
=== Pikel Setelah Normalisasi RGB ===
RGB Ternormalisasi (float32): [0.79607844 0.5921569 0.5058824 ]
```

Gambar 4. 7 Normalisasi RGB

Nilai [0.79607844 0.5921569 0.5058824] adalah hasil normalisasi dari nilai RGB sebelumnya. Setiap nilai dibagi dengan 255 untuk mengubah skala dari 0–255 menjadi 0–1. Ini umum dilakukan dalam pemrosesan citra dan deep learning agar jaringan saraf lebih cepat dan stabil saat belajar dari data gambar.

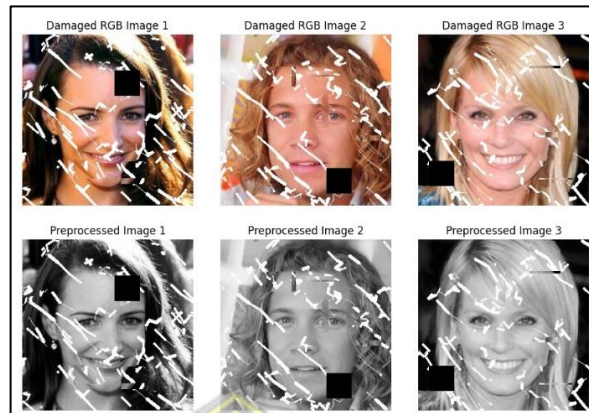
```
=== Pikel Setelah Normalisasi Grayscale ===
Grayscale Ternormalisasi (float32): 0.6431372761726379
```

Gambar 4. 8 Normalisasi *Grayscale*

Nilai 0.6431372761726379 adalah hasil dari membagi nilai grayscale 164 dengan 255. Ini dilakukan untuk merubah skala intensitas menjadi antara 0 dan 1. Normalisasi ini membuat data lebih seragam dan lebih cocok untuk digunakan dalam pelatihan model pembelajaran mesin atau analisis numerik lebih lanjut.

4.2.2 Visualisasi Hasil *Preprocessing*

Untuk memastikan proses *preprocessing* berjalan dengan baik dan data sudah siap untuk digunakan dalam pelatihan model, dilakukan visualisasi citra hasil *preprocessing*.



Gambar 4. 9 Citra hasil *preprocessing*

Visualisasi hasil *preprocessing* ditampilkan pada gambar 4.9 yang memperlihatkan pasangan citra input rusak (berwarna) dan hasil *preprocessing* (*grayscale*). Citra bagian atas menunjukkan citra yang telah mengalami kerusakan digital seperti coretan acak, pemotongan area, dan distorsi blok. Citra-citra ini tetap dalam ukuran asli tanpa dilakukan proses *resize*, dan digunakan untuk menguji kemampuan model dalam mengatasi berbagai tingkat kerusakan visual. Sementara itu, citra bagian bawah merupakan hasil *preprocessing* berupa konversi ke *grayscale* dan normalisasi nilai *pixel* ke rentang $[0, 1]$. Proses ini dilakukan untuk menyederhanakan representasi data sekaligus menjaga kestabilan saat proses pelatihan model GAN.

Hasil visualisasi menunjukkan bahwa struktur spasial utama dari citra tetap terjaga setelah *preprocessing*. Meskipun warna dihilangkan, fitur penting seperti bentuk wajah, posisi mata, serta garis tepi objek masih terlihat jelas, yang menjadi indikator bahwa *preprocessing* tidak menghilangkan informasi penting dalam citra. Dengan hasil visual seperti ini, model diharapkan dapat belajar merestorasi citra secara optimal, terutama dalam konteks restorasi citra yang mengalami kerusakan parah. Gambar 4.9 menjadi bukti bahwa data masukan telah diproses secara konsisten dan siap digunakan pada tahap pelatihan selanjutnya.

4.3 Hasil Pemodelan

Pada tahap ini, dilakukan perancangan dan pembangunan arsitektur model *deep learning* yang digunakan untuk proses restorasi citra yang mengalami kerusakan digital. Model yang digunakan berbasis pendekatan pix2pix, sebuah arsitektur *Generative Adversarial Network* (GAN) yang dirancang untuk tugas *image-to-image translation*. Dalam pendekatan ini, dua model utama dikembangkan dan dilatih secara bersamaan, yaitu *Generator* dan *Discriminator*. *Generator* bertugas merestorasi citra rusak menjadi citra yang tampak utuh dan realistis, sedangkan *Discriminator* bertugas membedakan apakah citra hasil restorasi merupakan citra asli atau buatan.

Kedua model diimplementasikan menggunakan *framework* TensorFlow dan dibangun sesuai spesifikasi arsitektural yang telah dirancang sebelumnya. Dataset yang digunakan telah diproses sehingga seluruh input memiliki ukuran tetap 256×256 *pixel*, dengan format *grayscale* (1 channel) untuk input dan RGB (3 channel) untuk target output. Output dari modeling ini berupa struktur detail dari kedua arsitektur lengkap dengan jumlah parameter serta hubungan antar-layer berdasarkan hasil pemanggilan `model.summary()`.

4.3.1 Ringkasan Model *Generator* (U-Net)

Generator dalam sistem ini dibangun menggunakan arsitektur U-Net, yakni salah satu bentuk Convolutional Neural Network (CNN) yang secara luas digunakan dalam tugas segmentasi dan restorasi citra. Arsitektur ini memiliki struktur simetris encoder-decoder yang diperkuat dengan mekanisme skip connection, sehingga informasi spasial resolusi tinggi dari encoder dapat langsung ditransmisikan ke decoder. Hal ini penting untuk mempertahankan detail citra yang sering kali hilang saat proses downsampling. Arsitektur U-Net pada sistem ini menerima input berupa citra grayscale berukuran $256 \times 256 \times 1$ dan menghasilkan output berupa citra RGB $256 \times 256 \times 3$. Model ini terdiri dari delapan blok downsampling (encoder) dan delapan blok upsampling (decoder),

termasuk output layer sebagai bagian akhir dari proses upsampling. Pada bagian encoder digunakan fungsi aktivasi LeakyReLU, sedangkan pada bagian decoder digunakan aktivasi ReLU. Layer output menggunakan fungsi aktivasi tanh untuk memastikan bahwa nilai *pixel* berada dalam rentang $[-1, 1]$.

Jumlah delapan blok downsampling dan delapan blok upsampling dipilih berdasarkan pertimbangan struktural dan efisiensi resolusi. Dalam proses konvolusi dengan kernel 4×4 dan stride 2, setiap operasi downsampling akan mengurangi dimensi spasial citra sebesar setengahnya. Maka, dimensi citra akan turun berturut-turut dari $256 \times 256 \rightarrow 128 \times 128 \rightarrow$ hingga akhirnya menjadi 1×1 pada downsampling ke-8. Resolusi 1×1 ini dipertahankan sebagai bottleneck yang menyimpan representasi fitur global dari seluruh citra. Untuk mencapai kembali dimensi 256×256 , diperlukan delapan tahap transposed convolution dengan stride 2, termasuk tahap terakhir (output layer), sehingga proses upsampling pun dilakukan sebanyak delapan kali. Struktur simetris ini memastikan efisiensi pemrosesan dan kestabilan selama pelatihan, sekaligus menjaga keseimbangan antara representasi lokal dan global dalam fitur yang diekstrak.

Pemrosesan gambar di dalam *Generator* U-Net berlangsung melalui tahapan sebagai berikut:

1. Citra input dimasukkan sebagai citra grayscale berukuran $256 \times 256 \times 1$.
2. Gambar diproses melalui delapan blok downsampling, yang masing-masing terdiri dari:
 - Operasi konvolusi (Conv2D) dengan kernel 4×4 dan stride 2.
 - Batch normalization (kecuali pada layer pertama).
 - Fungsi aktivasi LeakyReLU.
3. Setiap blok downsampling mengurangi resolusi gambar dan meningkatkan jumlah saluran fitur (channel), misalnya dari $256 \times 256 \times 1$ menjadi

$128 \times 128 \times 64$, dan seterusnya hingga menjadi $1 \times 1 \times 512$.

4. Setelah tahap bottleneck ($1 \times 1 \times 512$), dilakukan proses upsampling melalui delapan blok decoder, yang masing-masing terdiri dari:
 - Transposed convolution (Conv2DTranspose) dengan kernel 4×4 dan stride 2.
 - Batch normalization (kecuali layer output).
 - Fungsi aktivasi ReLU (kecuali layer output).
 - *Skip connection* (pada tujuh blok pertama): output dari layer encoder dengan resolusi yang sama dikoneksikan ke input layer decoder.
5. Ukuran gambar diperbesar bertahap hingga kembali ke dimensi awal, dengan output akhir berukuran $256 \times 256 \times 3$.
6. Layer terakhir menggunakan fungsi aktivasi tanh untuk menghasilkan citra RGB dengan nilai *pixel* pada rentang $[-1, 1]$.

Output `model.summary()` menunjukkan bahwa total parameter pada *Generator* mencapai lebih dari 54 juta parameter, menjadikannya komponen utama dari sistem yang memiliki kapasitas belajar paling tinggi. Rincian struktur arsitektur *Generator* ditunjukkan pada Tabel 4.2.

Table 4. 2 Ringkasan *modelling Generator*

No	Layer (Type)	Output Shape	Jumlah Parameter
1	Input Layer	(256, 256, 1)	0
2	Downsampling #1	(128, 128, 64)	1.024
3	Downsampling #2	(64, 64, 128)	131.584
4	Downsampling #3	(32, 32, 256)	525.312
5	Downsampling #4	(16, 16, 512)	2.099.200
6	Downsampling #5	(8, 8, 512)	4.196.352
7	Downsampling #6	(4, 4, 512)	4.196.352
8	Downsampling #7	(2, 2, 512)	4.196.352
9	Downsampling #8	(1, 1, 512)	4.196.352
10	Upsampling #1	(2, 2, 512)	4.196.352
11	Upsampling #2	(4, 4, 512)	8.390.656
12	Upsampling #3	(8, 8, 512)	8.390.656
13	Upsampling #4	(16, 16, 512)	8.390.656
14	Upsampling #5	(32, 32, 256)	4.195.328
15	Upsampling #6	(64, 64, 128)	1.049.088
16	Upsampling #7	(128, 128, 64)	262.400
17	Output Layer	(256, 256, 3)	6.147
Total Parameter			54.423.811
Trainable			54.412.931
Non Trainable			10.880

Dari total parameter sebanyak 54,4 juta, hampir seluruhnya (lebih dari 99%) bersifat *trainable*. Ini berarti model memiliki kapasitas besar untuk belajar dan melakukan pemetaan *non-linear* yang kompleks dari input citra rusak menuju citra hasil restorasi. Dengan desain seperti ini, *Generator* berperan sebagai komponen utama dari sistem restorasi citra yang bertugas melakukan

pemetaan kompleks dari input citra grayscale yang rusak menuju citra RGB hasil restorasi. Struktur U-Net yang menggabungkan kelebihan encoder-decoder serta skip connection menjadikannya efektif dalam menghasilkan output citra berkualitas tinggi, dengan ketajaman detail dan warna yang akurat.

4.3.2 Ringkasan Model *Discriminator* (PatchGAN)

Discriminator dalam sistem ini dibangun berdasarkan arsitektur PatchGAN, yaitu jaringan diskriminator yang mengevaluasi keaslian citra secara lokal, bukan keseluruhan gambar. Alih-alih memberikan satu nilai global, PatchGAN menghasilkan peta prediksi (score map) yang masing-masing elemennya merepresentasikan validitas dari patch kecil pada citra (sekitar 70×70 pixel). Pendekatan ini terbukti lebih efektif untuk tugas restorasi citra karena menekankan pada realisme detail lokal seperti tekstur kulit, garis wajah, dan kontur halus lainnya.

Arsitektur *Discriminator* ini menerima dua input:

1. Citra rusak (grayscale) berukuran $256 \times 256 \times 1$.
2. Citra target (RGB) berukuran $256 \times 256 \times 3$.

Kedua input ini kemudian digabungkan secara kanal (channel-wise) menggunakan operasi Concatenate, menghasilkan tensor berukuran $256 \times 256 \times 4$. Tensor ini kemudian dilewatkan ke dalam beberapa blok Conv2D berturut-turut, dengan jumlah filter yang meningkat hingga 512. Tidak digunakan layer fully connected, sehingga struktur jaringan bersifat sepenuhnya konvolusional (fully convolutional), yang membuatnya lebih ringan dan cepat dilatih.

Setiap operasi konvolusi dilakukan dengan kernel 4×4 dan stride 2, disertai padding. Fungsi aktivasi yang digunakan adalah LeakyReLU. Batch normalization diterapkan setelah beberapa lapisan untuk menstabilkan pelatihan. Pada akhir jaringan, dilakukan konvolusi $4 \times 4 \times 512 \rightarrow 1$ tanpa aktivasi sigmoid karena diproses lebih lanjut dalam perhitungan loss. Hasil akhir berupa output tensor dengan dimensi $30 \times 30 \times 1$, yaitu peta probabilitas keaslian dari tiap patch

citra.

Pemrosesan citra dalam *Discriminator* berlangsung melalui langkah-langkah berikut:

1. Dua buah citra (grayscale dan RGB) digabungkan menjadi tensor $256 \times 256 \times 4$.
2. Tensor dilewatkan ke Conv2D bertingkat dengan kernel 4×4 dan stride 2.
3. Resolusi gambar menurun setiap layer: $256 \rightarrow 128 \rightarrow 64 \rightarrow 32$, dan seterusnya.
4. Jumlah filter meningkat untuk menangkap fitur kompleks lokal: $64 \rightarrow 128 \rightarrow 256 \rightarrow 512$.
5. Pada tahap akhir, dilakukan padding dan konvolusi menjadi tensor $30 \times 30 \times 1$.
6. Setiap elemen 30×30 mewakili prediksi terhadap satu patch citra berukuran kira-kira 70×70 pixel.
7. Output digunakan sebagai dasar evaluasi keaslian citra hasil restorasi oleh *Generator*.

Struktur lengkap dari arsitektur *Discriminator* beserta perhitungan jumlah parameter pada setiap layer secara rinci ditunjukkan pada Tabel 4.3 berikut:

Table 4. 3 Ringkasan *modelling Discriminator*

No	Layer (Type)	Output Shape	Jumlah Parameter
1	Input <i>Grayscale</i>	(256, 256, 1)	0
2	Input RGB	(256, 256, 3)	0
3	Concatenate	(256, 256, 4)	0
4	Conv2D #1 (64 filter)	(128, 128, 64)	4.096
5	Conv2D #2 (128 filter)	(64, 64, 128)	131.584
6	Conv2D #3 (256 filter)	(32, 32, 256)	525.312
7	Zero Padding	(34, 34, 256)	0
8	Conv2D #4 (512 filter)	(31, 31, 512)	2.097.152

9	BatchNormalization	(31, 31, 512)	2.048
10	LeakyReLU	(31, 31, 512)	0
11	Zero Padding	(33, 33, 512)	0
12	Output Conv2D (1 filter)	(30, 30, 1)	8.193
Total Parameter			2.768.385
Trainable			2.766.593
Non Trainable			1.792

Meskipun jumlah parameter pada *Discriminator* jauh lebih kecil dibandingkan *Generator* yaitu sekitar 2,7 juta berbanding 54 juta, perannya sangat penting sebagai evaluator yang memberikan umpan balik kepada *Generator* dalam kerangka kerja Generative Adversarial Network (GAN). Dengan pendekatan PatchGAN, *Discriminator* tidak menilai citra secara keseluruhan, melainkan mengevaluasi keaslian patch-patch kecil (sekitar 70×70 pixel) melalui output berupa peta skor berukuran $30 \times 30 \times 1$. Strategi ini memungkinkan model lebih fokus pada detail lokal seperti tekstur kulit dan kontur wajah, yang sangat krusial dalam tugas restorasi citra. Selain itu, arsitekturnya yang sepenuhnya konvolusional tanpa lapisan fully connected menjadikannya efisien secara komputasi dan fleksibel terhadap variasi resolusi input, sementara penggunaan batch normalization dan padding membantu menjaga stabilitas proses pelatihan.

4.3.3 Analisis Komparatif *Generator* dan *Discriminator*

Jika dilihat dari total parameter, *Generator* memiliki kompleksitas jauh lebih tinggi dibandingkan *Discriminator*, yaitu 54 juta berbanding 2,7 juta parameter. Hal ini mencerminkan peran *Generator* sebagai komponen utama yang bertugas menghasilkan citra baru yang realistis dan menyerupai target. *Discriminator*, di sisi lain, berperan sebagai evaluator yang relatif ringan, namun cukup sensitif untuk mendeteksi ketidaksesuaian dalam hasil restorasi *Generator*.

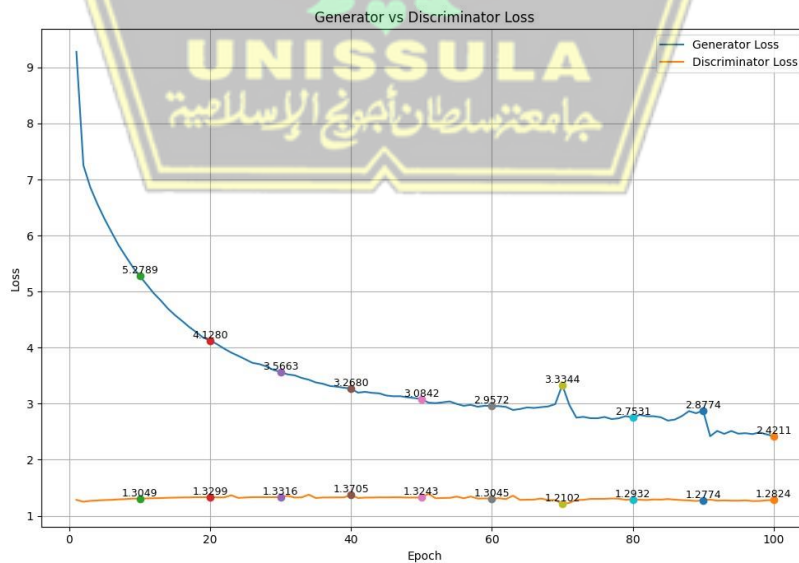
Sebagaimana ditunjukkan pada Tabel 4.2 dan Tabel 4.3, perbedaan kompleksitas ini tercermin dari kedalaman layer, dimensi output antar-layer, serta jumlah parameter *trainable* pada masing-masing arsitektur. Kombinasi dari dua arsitektur ini memungkinkan sistem untuk melakukan pembelajaran *end-to-end* terhadap konversi dari citra rusak ke citra utuh yang menyerupai citra *ground truth*.

4.4 Hasil Pelatihan

Setelah proses pemodelan selesai, tahap selanjutnya adalah melatih model GAN menggunakan dataset wajah berukuran 256×256 *pixel*. Pelatihan dilakukan selama 100 epoch dengan *Loss Binary Crossentropy* untuk *Discriminator* serta kombinasi GAN *Loss* dan L1 *Loss* untuk *Generator*. Proses ini dibagi ke dalam tiga fase berdasarkan rasio frekuensi pelatihan antara *Generator* dan *Discriminator*.

4.4.1 Perbandingan Loss

Perkembangan nilai *Generator Loss* dan *Discriminator Loss* sepanjang proses pelatihan ditunjukkan pada Gambar 4.10 berikut:



Gambar 4. 10 Genertaor Loss vs Discriminator Loss

Berdasarkan gambar 4.10, dinamika pelatihan menunjukkan tren yang jelas dan dapat dibagi ke dalam beberapa fase:

1. Epoch 1–60 (Rasio 1:1):

Pada fase awal, *Generator* dan *Discriminator* dilatih secara seimbang (rasio 1:1). Selama periode ini, *Generator Loss* mengalami penurunan yang konsisten dan signifikan, dengan fluktuasi yang relatif kecil. Hal ini menunjukkan bahwa model berhasil belajar dengan stabil dan efektif selama fase awal pelatihan.

2. Epoch 61–70 (Rasio 1:1):

Memasuki fase ini, *Generator Loss* mulai menunjukkan peningkatan bertahap, yang berpotensi mengindikasikan ketertinggalan *Generator* terhadap *Discriminator*. Tren ini juga berdampak pada penurunan nilai evaluasi.

3. Epoch 71–90 (Rasio 2:1):

Untuk mengatasi tren tersebut, dilakukan penyesuaian strategi pelatihan dengan menaikkan rasio pelatihan *Generator* menjadi 2:1, di mana *Generator* dilatih dua kali lebih sering dibandingkan *Discriminator*. Perubahan ini berhasil menurunkan kembali nilai *Generator Loss*, meskipun terjadi beberapa fluktuasi dan sedikit kenaikan *Loss* mendekati akhir fase ini.

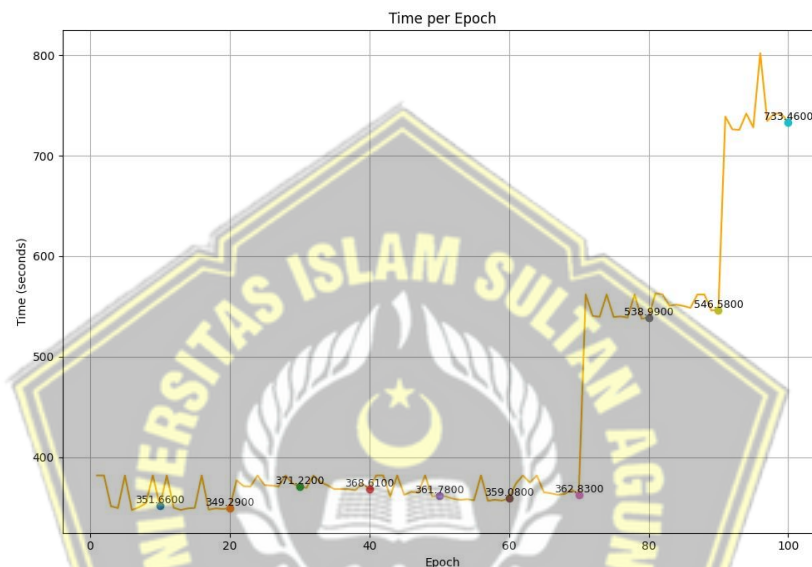
4. Epoch 91–100 (Rasio 3:1):

Untuk mengurangi fluktuasi dan mendorong konvergensi lebih lanjut, rasio pelatihan kembali ditingkatkan menjadi 3:1. Strategi ini memberikan hasil positif dengan penurunan *Generator Loss* yang cukup signifikan dan stabil hingga akhir pelatihan. Nilai evaluasi pun turut mengalami peningkatan.

Secara keseluruhan, perubahan rasio pelatihan *Generator* terbukti efektif dalam menjaga kestabilan pelatihan dan meningkatkan performa model serta mampu menurunkan *Generator Loss* dengan cukup baik.

4.4.2 Waktu Pelatihan

Perubahan rasio pelatihan *Generator* turut berdampak pada durasi pelatihan per Epoch. Meningkatnya jumlah langkah update *Generator* per Epoch menyebabkan beban komputasi bertambah. Rata-rata waktu pelatihan per Epoch pada tiap fase ditunjukkan pada gambar 4.11 berikut:



Gambar 4. 11 Waktu pelatihan

1. Epoch 1–70 (Rasio 1:1):

Waktu pelatihan per Epoch berkisar antara 350 hingga 385 detik. Ini merupakan fase pelatihan paling ringan secara waktu karena kedua model dilatih secara seimbang.

2. Epoch 71–90 (Rasio 2:1):

Dengan peningkatan frekuensi pelatihan *Generator*, durasi Epoch meningkat ke 535 hingga 565 detik.

3. Epoch 91–100 (Rasio 3:1):

Fase terakhir mencatat durasi tertinggi, yaitu 725 hingga 805 detik per Epoch, seiring *Generator* dilatih tiga kali lebih banyak per batch.

Peningkatan waktu pelatihan ini merupakan konsekuensi dari strategi pelatihan yang agresif, dan terbukti efektif karena menghasilkan penurunan *Loss* yang signifikan serta perbaikan nilai evaluasi.

4.5 Hasil Evaluasi

Setelah model dilatih selama 100 epoch, dilakukan evaluasi performa terhadap *Generator* dengan menggunakan dua metrik kuantitatif utama, yaitu *Structural Similarity Index Measure* (SSIM) dan *Peak Signal-to-Noise Ratio* (PSNR). Evaluasi dilakukan terhadap hasil citra pada data uji untuk mengetahui sejauh mana hasil prediksi model mendekati citra ground truth, baik dari segi struktur maupun kualitas visual.

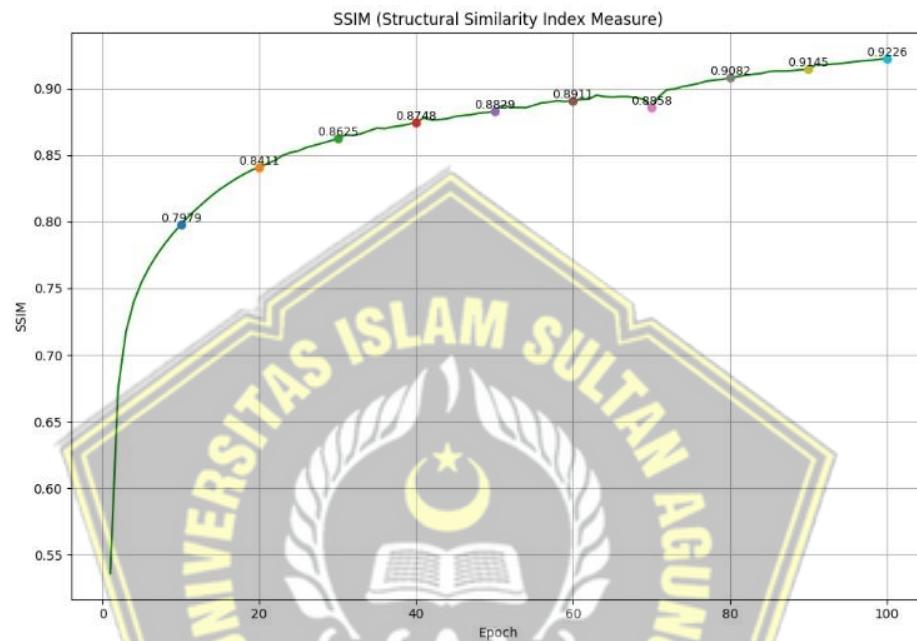
SSIM digunakan untuk mengukur kemiripan struktural antara dua citra, sedangkan PSNR digunakan untuk mengevaluasi kualitas sinyal terhadap *noise*. Kedua metrik dihitung secara periodik setiap 10 epoch guna memantau perkembangan model secara bertahap, dan disajikan dalam bentuk grafik.

4.5.1 Evaluasi SSIM

Structural Similarity Index Measure (SSIM) mengukur kualitas citra dari segi struktur, luminansi, dan kontras. Nilai SSIM berada dalam rentang 0 hingga 1, di mana nilai yang lebih tinggi menunjukkan kemiripan yang lebih besar

terhadap citra asli. Metrik ini sangat penting dalam konteks restorasi citra wajah, karena struktur objek seperti bentuk mata, mulut, dan kontur wajah sangat menentukan keberhasilan visualisasi.

Grafik perkembangan SSIM disajikan pada gambar 4.12 berikut:



Gambar 4. 12 SSIM

Pada awal pelatihan, nilai SSIM (*Structural Similarity Index*) model masih tergolong masih cukup rendah, yaitu 0.7979 pada epoch 10, yang menunjukkan bahwa struktur citra hasil masih belum menyerupai target. Namun, seiring pelatihan berlangsung, model menunjukkan peningkatan stabil dan progresif, yang dapat dirangkum sebagai berikut:

1. Epoch 20: SSIM meningkat menjadi 0.8411, menunjukkan bahwa model mulai memahami struktur dasar dari citra target.
2. Epoch 30: Nilai SSIM terus naik menjadi 0.8625, memperlihatkan perbaikan lebih lanjut dalam reproduksi pola visual.
3. Epoch 40: SSIM mencapai 0.8748, menandakan bahwa model semakin mampu mendekati ground truth secara struktural.
4. Epoch 50–60: Nilai SSIM berturut-turut menjadi 0.8829 dan 0.8911,

mencerminkan hasil yang semakin menyerupai target, dan menjadi titik penting dalam evaluasi performa model.

Namun, pada epoch 70, terjadi penurunan SSIM menjadi 0.8858, yang lebih rendah dibandingkan epoch sebelumnya. Penurunan ini mengindikasikan adanya gangguan dalam pelatihan, kemungkinan disebabkan oleh dinamika dari *adversarial training* atau ketidakseimbangan sementara dalam proses pembelajaran. Meskipun penurunannya tidak ekstrem, hal ini tetap penting dicermati sebagai bagian dari evaluasi performa.

Setelah penurunan tersebut, SSIM kembali meningkat cukup signifikan dikarenakan peningkatan pelatihan pada *Generator*:

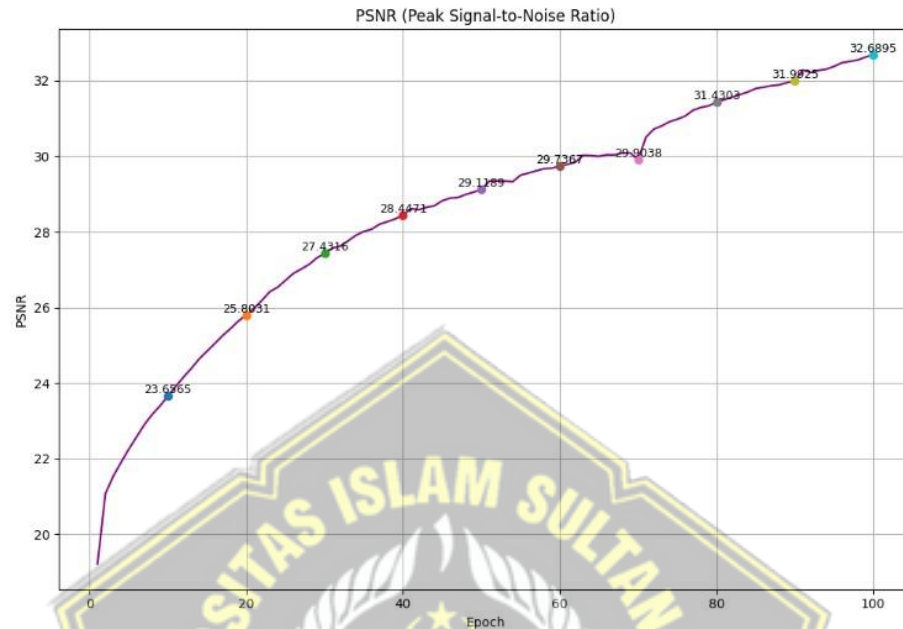
1. Epoch 80: SSIM meningkat menjadi 0.9082.
2. Epoch 90: Meningkat lagi menjadi 0.9145.
3. Epoch 100: Mencapai nilai tertinggi yaitu 0.9226.

Kenaikan nilai SSIM di atas 0.9 pada akhir pelatihan menunjukkan bahwa model telah berhasil mengatasi ketidakstabilan sebelumnya dan mampu menghasilkan citra yang secara struktural sangat mirip dengan citra asli. Ini mengindikasikan kemampuan tinggi model dalam mempertahankan struktur serta detail penting dalam proses restorasi citra.

4.5.2 Evaluasi PSNR

Peak Signal-to-Noise Ratio (PSNR) digunakan untuk mengukur seberapa baik kualitas citra hasil dibandingkan dengan citra asli, terutama dalam hal kejernihan dan tingkat *noise*. Semakin tinggi nilai PSNR, maka semakin kecil kesalahan *pixel* antara dua citra, yang menunjukkan bahwa hasil restorasi semakin mendekati citra referensi. PSNR dinyatakan dalam satuan desibel (dB), dan sering digunakan sebagai indikator numerik untuk menilai kualitas restorasi citra dalam berbagai aplikasi pengolahan citra digital.

Grafik perkembangan nilai PSNR dapat dilihat pada Gambar 4.13 berikut:



Gambar 4. 13 PSNR

Pada awal pelatihan, nilai PSNR (*Peak Signal-to-Noise Ratio*) berada di 23.6565 dB pada epoch 10, yang mencerminkan bahwa hasil rrestorasi citra masih mengandung banyak *noise* dan kesalahan *pixel* yang tinggi. Ini wajar terjadi di awal pelatihan karena model masih dalam tahap awal belajar mengenali pola-pola visual dari data dan belum mampu menghasilkan representasi yang akurat dari citra target.

Seiring dengan berjalannya proses pelatihan, model mulai belajar dari *loss function* dan *feedback* dari *Discriminator* (pada arsitektur GAN), sehingga hasil citra menjadi semakin baik secara numerik. Hal ini ditunjukkan oleh peningkatan bertahap nilai PSNR sebagai berikut:

1. Epoch 20 – 25.8031 dB: Menunjukkan bahwa model sudah mulai mengurangi *noise*, dan hasil citra mulai menyerupai struktur dasar dari citra asli.
2. Epoch 30 – 27.4316 dB dan Epoch 40 – 28.4471 dB: Menunjukkan peningkatan akurasi *pixel* dan stabilnya proses pembelajaran model.

3. Epoch 50 – 29.1189 dB dan Epoch 60 – 29.7367 dB: Merupakan fase di mana model telah memahami pola penting dalam data dan mampu menghasilkan citra dengan *noise* yang relatif rendah.

Pada epoch 70, meskipun SSIM mengalami sedikit penurunan (yang berarti struktur global sedikit terganggu), nilai PSNR justru naik ke 29.9038 dB. Ini menandakan bahwa kesalahan numerik antar *pixel* masih terus menurun, walaupun dari segi struktur (tekstur, tepi, atau pola visual) sempat mengalami sedikit penurunan. Hal ini mungkin disebabkan oleh ketidakseimbangan sementara antara *Generator* dan *Discriminator*, atau oleh model yang terlalu fokus pada pencocokan *pixel* (*pixel-wise matching*), bukan struktur.

Setelah itu, PSNR terus meningkat cukup signifikan:

1. Epoch 80: Mencapai 31.4303 dB.
2. Epoch 90: Naik menjadi 31.9925 dB.
3. Epoch 100: Mencapai puncak di 32.6895 dB.

Nilai-nilai ini menunjukkan bahwa model telah memasuki fase konvergensi, yaitu model tidak hanya mampu mempertahankan stabilitas pelatihan, tetapi juga secara konsisten menghasilkan citra dengan kualitas visual tinggi dan kesalahan *pixel* yang sangat minim. Puncaknya pada epoch 100, dengan PSNR mencapai 32.6895 dB, ini mengindikasikan bahwa model telah berhasil mempelajari representasi visual dengan sangat baik dan mampu merestorasi citra yang sangat dekat dengan *ground truth*.

Secara keseluruhan, tren peningkatan PSNR menunjukkan bahwa model memiliki kemampuan progresif dalam mengurangi *noise* dan kesalahan numerik antar *pixel*, yang sangat penting dalam restorasi citra atau *image-to-image translation*. Konsistensi kenaikan PSNR juga menandakan stabilitas pelatihan yang baik, dan bahwa strategi pelatihan yang digunakan baik dari sisi arsitektur maupun hyperparameter telah bekerja dengan efektif.

4.6 Hasil Implementasi

Setelah tahap pemodelan dan pelatihan selesai, selanjutnya implementasi sistem restorasi citra ke dalam *platform* website dengan menggunakan *framework flask*. Berikut ini merupakan hasil implementasi sistem :

RESTORASI FOTO AKIBAT KERUSAKAN DIGITAL MENGUNAKAN DEEP LEARNING BERBASIS GENERATIVE ADVERSARIAL NETWORKS (GANs)

Upload Damaged Image

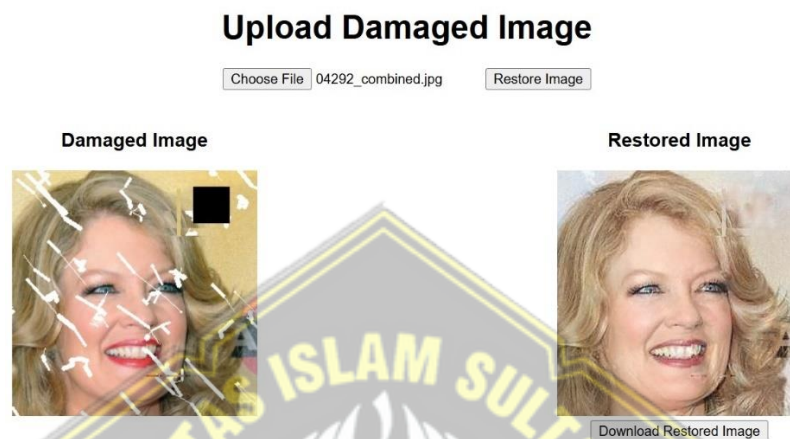


Gambar 4. 14 Halaman utama

Pada Gambar 4. 14 merupakan tampilan halaman utama yang akan ditampilkan pengguna saat pertama kali menjalankan sistem. Pada halaman ini terdapat judul utama "RESTORASI CITRA AKIBAT KERUSAKAN DIGITAL MENGGUNAKAN *DEEP LEARNING* BERBASIS *GENERATIVE ADVERSARIAL NETWORKS* (GANs)" yang menjelaskan fungsi utama aplikasi, yaitu restorasi citra akibat kerusakan digital menggunakan *Deep Learning* dengan model GAN yaitu pix2pix. Pada halaman ini juga terdapat tombol "Choose file" yang dapat digunakan untuk mengunggah citra dengan format JPG, JPEG dan PNG, lalu ada juga tombol "Restore Image" yang digunakan untuk memulai proses restorasi citra setelah file citra rusak berhasil di unggah.

4.6.1 Implementasi Tipe 1 (Ringan/Mudah)

RESTORASI FOTO AKIBAT KERUSAKAN DIGITAL MENGUNAKAN DEEP LEARNING BERBASIS GENERATIVE ADVERSARIAL NETWORKS (GANs)



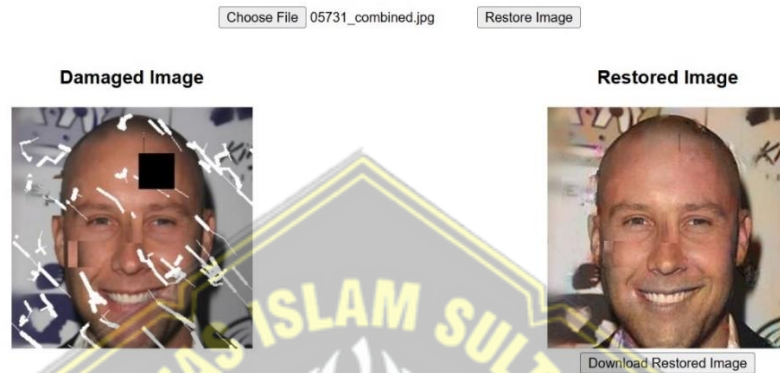
Gambar 4. 15 Hasil restorasi citra tipe 1

Pada Gambar 4.15 ditampilkan hasil penerapan model pada citra tipe 1, yaitu citra dengan tingkat kerusakan yang relatif ringan. Pada jenis kerusakan ini, bagian yang hilang (*region masking*) muncul di luar area wajah, sehingga tidak mengganggu fitur-fitur penting seperti mata, hidung, dan mulut yang memiliki peran besar dalam proses restorasi wajah. Kondisi ini memberikan keuntungan bagi model karena informasi utama wajah tetap utuh, memungkinkan proses rekonstruksi berlangsung lebih efektif dan akurat. Berdasarkan hasil yang ditunjukkan, sebagian besar area citra berhasil direkonstruksi dengan baik, dengan tingkat keberhasilan diperkirakan mencapai sekitar 95%. Meskipun masih terdapat sedikit kerusakan digital di beberapa area kecil, hal tersebut tidak memberikan dampak signifikan terhadap tampilan keseluruhan citra. Citra tetap terlihat utuh, natural, dan menyatu dengan baik, menunjukkan bahwa model sangat efektif dalam menangani kerusakan ringan.

4.6.2 Implementasi Tipe 2 (Sedang)

RESTORASI FOTO AKIBAT KERUSAKAN DIGITAL MENGUNAKAN DEEP LEARNING BERBASIS GENERATIVE ADVERSARIAL NETWORKS (GANs)

Upload Damaged Image



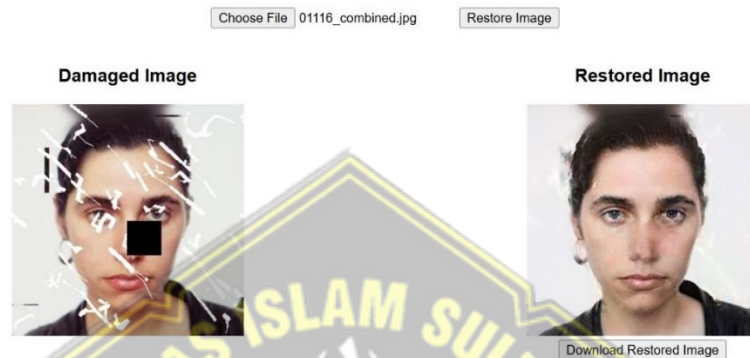
Gambar 4. 16 Hasil restorasi citra tipe 2

Pada Gambar 4.16 ditampilkan hasil implementasi model terhadap citra tipe 2, yaitu citra dengan tingkat kerusakan sedang. Pada kasus ini, Bagian hilang (*region masking*) mulai memasuki area wajah, namun masih berada di luar bagian-bagian vital seperti mata, hidung, dan mulut yang merupakan fitur penting dalam identifikasi wajah. Meskipun kerusakan sudah mulai mengenai area yang lebih sensitif dibandingkan tipe 1, model masih mampu melakukan proses restorasi dengan cukup baik karena informasi utama dari wajah masih dapat dikenali. Proses pemulihan citra berjalan cukup optimal, dan hasilnya menunjukkan bahwa sebagian besar area yang terdampak berhasil diperbaiki. Tingkat keberhasilan restorasi diperkirakan mencapai sekitar 93%. Namun, masih terdapat sedikit kerusakan digital yang tersisa, terutama di area dekat telinga kanan, yang menunjukkan bahwa model mulai menghadapi tantangan lebih besar saat kerusakan mendekati atau mengenai area wajah. Meskipun demikian, hasil akhir tetap menunjukkan kualitas visual yang baik dan citra tampak menyatu secara keseluruhan.

4.6.3 Implementasi Tipe 3 (Berat/Sulit)

RESTORASI FOTO AKIBAT KERUSAKAN DIGITAL MENGUNAKAN DEEP LEARNING BERBASIS GENERATIVE ADVERSARIAL NETWORKS (GANs)

Upload Damaged Image



Gambar 4. 17 Hasil restorasi citra tipe 3

Pada Gambar 4.17 ditampilkan hasil implementasi model pada citra tipe 3, yaitu dengan tingkat kerusakan berat. Bagian hilang (*region masking*) menutupi sebagian besar area penting, termasuk setengah hidung, salah satu fitur utama wajah. Meski belum sempurna, model mampu menggeneralisasi bentuk hidung dengan cukup baik. Kerusakan lain yang belum diperbaiki secara sempurna juga tampak pada bagian rambut akibat distorsi blok. Tingkat keberhasilan restorasi diperkirakan 85–90%, menunjukkan kemampuan model yang cukup baik dalam menangani kerusakan berat, meskipun akurasi dan detail menurun dibanding tipe kerusakan ringan.

Secara keseluruhan, kualitas restorasi dipengaruhi oleh tingkat kerusakan citra. Pada kerusakan ringan, hasil restorasi hampir sempurna karena fitur wajah utama tetap utuh. Untuk kerusakan sedang dan berat, meski detail mulai menurun, model masih mampu memberikan hasil yang cukup baik. Ini menunjukkan kemampuan adaptif model terhadap berbagai kondisi kerusakan. Selain itu, tersedia tombol “*Download Restored Image*” yang memudahkan pengguna mengunduh hasil restorasi.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian, telah berhasil dikembangkan sistem restorasi citra akibat kerusakan digital menggunakan pendekatan deep learning berbasis *Generative Adversarial Networks* (GAN) dengan arsitektur Pix2Pix, yang menggabungkan U-Net sebagai *Generator* dan PatchGAN sebagai *Discriminator*. Model ini dilatih menggunakan dataset CelebA-HQ *resized* (256 x 256) yang telah dimodifikasi untuk mensimulasikan tiga tingkat kerusakan digital, dan menunjukkan performa terbaik pada epoch ke-100 dengan nilai SSIM sebesar 0.9226 dan PSNR sebesar 32.6895 dB. Nilai tersebut menunjukkan bahwa model mampu menghasilkan citra hasil restorasi dengan kemiripan struktural yang tinggi dan kualitas visual yang mendekati citra aslinya. Evaluasi visual juga mendukung hasil tersebut, di mana detail penting berhasil dipulihkan secara meyakinkan pada berbagai tingkat kerusakan. Sistem ini telah diimplementasikan dalam bentuk aplikasi web yang interaktif dan mudah digunakan, serta memiliki potensi untuk diterapkan secara luas dalam bidang restorasi citra berbasis kecerdasan buatan.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, saran untuk pengembangan sistem restorasi citra akibat kerusakan digital ke depannya adalah memperluas variasi dataset dengan jenis citra dan bentuk kerusakan yang lebih beragam, agar model memiliki kemampuan generalisasi yang lebih baik. Selain itu, disarankan meningkatkan kualitas dan jumlah data latih serta mengoptimalkan struktur model untuk menghasilkan restorasi yang lebih detail, akurat, dan stabil pada berbagai kondisi kerusakan citra digital.

DAFTAR PUSTAKA

- Adiyaman, H., Emre Varul, Y., Bakirman, T., and Bayram, B. (2024). Stripe Error Correction for Landsat-7 Using Deep Learning. *PFG - Journal of Photogrammetry, Remote Sensing and Geoinformation Science*. <https://doi.org/10.1007/s41064-024-00306-x>
- Alotaibi, A. (2020). Deep generative adversarial networks for image-to-image translation: A review. *Symmetry*, 12(10), 1–26. <https://doi.org/10.3390/sym12101705>
- Cao, M., Feng, H., Xiao, H., Computer, S., and Information Technology, S. (2023). An Improved GAN-Based Image Restoration Method for Imaging Logging Images. *Applied Sciences (Switzerland)*, 13(16). <https://doi.org/10.3390/app13169249>
- Caruso, D., De Santis, D., Del Gaudio, A., Guido, G., Zerunian, M., Polici, M., Valanzuolo, D., Pugliese, D., Persechino, R., Cremona, A., Barbato, L., Caloisi, A., Iannicelli, E., and Laghi, A. (2024). Low-dose liver CT: image quality and diagnostic accuracy of deep learning image reconstruction algorithm. *European Radiology*, 34(4), 2384–2393. <https://doi.org/10.1007/s00330-023-10171-8>
- Ding, F., Shen, Z., Zhu, G., Kwong, S., Zhou, Y., and Lyu, S. (2023). ExS-GAN: Synthesizing Anti-Forensics Images via Extra Supervised GAN. *IEEE Transactions on Cybernetics*, 53(11), 7162–7173. <https://doi.org/10.1109/TCYB.2022.3210294>
- Galteri, L., Seidenari, L., Bertini, M., and Bimbo, A. Del. (2019). Deep Universal Generative Adversarial Compression Artifact Removal. *IEEE Transactions on Multimedia*, 21(8), 2131–2145. <https://doi.org/10.1109/TMM.2019.2895280>
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11), 139–144. <https://doi.org/10.1145/3422622>
- Gotamaa, S., and Santiyasaa, I. W. S. (2021). Image Restoration Menggunakan Metode Lucy-Richardson Pada Citra Underwater. *Resolusi : Rekayasa Teknik Informatika Dan Informasi*, 1(3), 179–187. <https://djournals.com/resolusi/article/view/128>

- He, Y., Zhao, D., Ren, Z., Qin, H., and Jiang, P. (2024). *基于改进 pix2pix 框架的人脸图像修复模型*.
- Hoyez, H., Schockaert, C., Rambach, J., Mirbach, B., and Stricker, D. (2022). *Unsupervised Image-to-Image Translation : A Review*. 1–27.
- Karavarsamis, S., Gkika, I., Gkitsas, V., Konstantoudakis, K., and Zarpalas, D. (2022). A Survey of Deep Learning-Based Image Restoration Methods for Enhancing Situational Awareness at Disaster Sites: The Cases of Rain, Snow and Haze. *Sensors*, 22(13). <https://doi.org/10.3390/s22134707>
- Kushwaha, V., Shukla, P., and Nandi, G. C. (2023). Generating quality grasp rectangle using Pix2Pix GAN for intelligent robot grasping. *Machine Vision and Applications*, 34(1). <https://doi.org/10.1007/s00138-022-01362-2>
- Liu, K., Li, J., and Hussain Bukhari, S. S. (2022). Overview of Image Inpainting and Forensic Technology. *Security and Communication Networks*, 2022. <https://doi.org/10.1155/2022/9291971>
- Liu, M., Su, X., Yao, X., Hao, W., and Zhu, W. (2023). Lensless Image Restoration Based on Multi-Stage Deep Neural Networks and Pix2pix Architecture. *Photonics*, 10(11). <https://doi.org/10.3390/photonics10111274>
- Ma, C. (2024). Uncertainty-Aware GAN for Single Image Super Resolution. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(5), 4071–4079. <https://doi.org/10.1609/aaai.v38i5.28201>
- Maharani, M., and Salim, T. A. (2024). *Strategi Pemerintah menggunakan AI : Studi Kasus Restorasi Warisan Budaya (Koleksi Citra Prajurit) untuk Peringatan 70 Tahun Gencatan Senjata Korea Selatan* *Government Strategies Using AI : A Case Study of Cultural Heritage Restoration (Soldier Photo Col. 15(2)*, 117–130.
- Maulana, A., Fatichah, C., Suciati, N., Teknik Informatika, D., and Surabaya, N. (2020). Facial Inpainting Pada Citra Wajah Unaligned Menggunakan Generative Adversarial Network Dengan Feature Reconstruction Loss. *JUTI: Jurnal Ilmiah Teknologi Informasi*, 18(2), 171–178.

- Mudeng, V., Kim, M., and Choe, S. W. (2022). Prospects of Structural Similarity Index for Medical Image Analysis. *Applied Sciences (Switzerland)*, 12(8). <https://doi.org/10.3390/app12083754>
- Pang, Y., Lin, J., Qin, T., and Chen, Z. (2022). Image-to-Image Translation: Methods and Applications. *IEEE Transactions on Multimedia*, 24, 3859–3881. <https://doi.org/10.1109/TMM.2021.3109419>
- Ren, H., Sun, K., Zhao, F., and Zhu, X. (2024). Dunhuang murals image restoration method based on generative adversarial network. *Heritage Science*, 12(1). <https://doi.org/10.1186/s40494-024-01159-8>
- Salehi, P., and Chalechale, A. (2020). Pix2Pix-based Stain-to-Stain Translation: A Solution for Robust Stain Normalization in Histopathology Images Analysis. *Iranian Conference on Machine Vision and Image Processing, MVIP, 2020-Febru.* <https://doi.org/10.1109/MVIP49855.2020.9116895>
- Senthil Anandhi, A., and Jaiganesh, M. (2025). An enhanced image restoration using deep learning and transformer based contextual optimization algorithm. *Scientific Reports*, 15(1), 1–18. <https://doi.org/10.1038/s41598-025-94449-5>
- Siddique, N., Paheding, S., Elkin, C. P., and Devabhaktuni, V. (2021). U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications. *IEEE Access*, 9, 82031–82057. <https://doi.org/10.1109/ACCESS.2021.3086020>
- Singh, A., Kushwaha, S., Alarfaj, M., and Singh, M. (2022). Comprehensive Overview of Backpropagation Algorithm for Digital Image Denoising. *Electronics (Switzerland)*, 11(10). <https://doi.org/10.3390/electronics11101590>
- Sriram, A., Vatsa, A., Kumar, A., Vats, S., and Kumar, A. (2024). Challenges and Opportunities in Malignant Image Reconstruction Using GAN: A Review. *2024 IEEE Integrated STEM Education Conference, ISEC 2024, March.* <https://doi.org/10.1109/ISEC61299.2024.10664911>
- Susanto, A., Sinaga, D., Utomo, I., and Mulyono, W. (2024). PSNR and SSIM Performance Analysis of Schur Decomposition for Imperceptible Steganography. *11(3)*, 803–810. <https://doi.org/10.15294/sji.v11i3.9561>

- Trigka, M., and Dritsas, E. (2025). A Comprehensive Survey of Deep Learning Approaches in Image Processing. *Sensors*, 25(2). <https://doi.org/10.3390/s25020531>
- Triningsi, M., Ama, T., Himamunanto, A. R., Setyawan, C., Informatika, P. S., and Immanuel, U. K. (2024). *Peningkatan Resolusi Citra dengan Menggunakan Metode GAN untuk Aplikasi Peningkatan Citra*. 5(3), 1287–1296.
- Wang, X., Sun, L., Chehri, A., and Song, Y. (2023). A Review of GAN-Based Super-Resolution Reconstruction for Optical Remote Sensing Images. *Remote Sensing*, 15(20), 1–34. <https://doi.org/10.3390/rs15205062>
- Weng, W., and Zhu, X. (2021). INet: Convolutional Networks for Biomedical Image Segmentation. *IEEE Access*, 9, 16591–16603. <https://doi.org/10.1109/ACCESS.2021.3053408>
- Zhang, K., Zuo, W., Zhang, L., and Computing, D. (2019). Deep plug-and-play super-resolution for arbitrary blur kernels. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019-June*, 1671–1681. <https://doi.org/10.1109/CVPR.2019.00177>
- Zhang, Y., Li, W., Wang, D., and Feng, S. (2022). Unsupervised Image Translation Using Multi-Scale Residual GAN. *Mathematics*, 10(22). <https://doi.org/10.3390/math10224347>