

BAB III

METODOLOGI PENELITIAN

3.1 Pendahuluan

Pada bab metodologi penelitian ini diuraikan tentang rencana proses penelitian. Rencana proses penelitian ini meliputi unit penelitian, teknik pengumpulan data, metode pengembangan perangkat lunak yang digunakan, dan manajemen proyek penelitian.

3.2 Pengumpulan Data

3.2.1 Jenis Data

Data yang digunakan dalam penelitian ini adalah citra berwarna berukuran 256x256 piksel, dengan format jpg.

3.2.2 Sumber Data














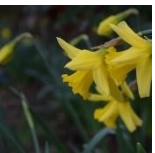

Dataset yang digunakan dalam penelitian ini adalah data sekunder bernama *Linnaeus 5*, yang dirilis oleh Chaladze dan Kalatozishvili (2017). Dataset ini tersedia secara bebas dan diunduh dari situs pixabay.com. Dataset ini telah dirancang untuk aplikasi pembelajaran mesin dengan fokus pada klasifikasi gambar.

Dataset terdiri dari:

- 1.200 citra pelatihan per kelas
- 400 citra pengujian per kelas
- Total: 8.000 citra berwarna dengan resolusi 256x256 piksel

3.2.3 Sampel Gambar dari Dataset

Contoh sampel dari data citra yang digunakan pada penelitian ini dapat dilihat pada Gambar III-1 dibawah ini.

Berry	Bird	Dog	Flower	Others
				
				
				

Gambar III-1 Sampel gambar (Chaladze & Kalatozishvili, 2017)

3.2.4 Metode Pengumpulan Data

Data dikumpulkan dengan cara mengunduh langsung dataset Linnaeus 5 dalam format JPEG dari tautan resmi yang disediakan oleh penulis dataset. Selanjutnya, citra diatur ke dalam direktori yang sesuai dengan label kelas masing-masing (*berry*, *bird*, *dog*, *flower*, *other*) untuk memudahkan *preprocessing* dan pelatihan model.

Dataset ini diakui dan dirujuk berdasarkan publikasi:

- Chaladze, G., & Kalatozishvili, L. (2017). "Linnaeus 5 Dataset for Machine Learning."

3.2.5 Konfigurasi Percobaan

Tabel berikut menyajikan konfigurasi parameter utama yang digunakan saat melakukan pelatihan model:

Tabel III-1 Tabel Hasil Evaluasi Pengujian Model Terbaik

Konfigurasi	Learning Rate	Epoch	Patience	Batch size
1				
2				
3				

Keterangan:

- *Learning Rate*: Mengatur seberapa besar langkah pembaruan bobot (*weight*) saat proses pelatihan.
- *Epoch*: Satu *epoch* merepresentasikan satu kali proses penuh melewati seluruh data latih.
- *Patience*: Jumlah *epoch* tanpa peningkatan kinerja sebelum mekanisme *early stopping* menghentikan pelatihan secara otomatis.

3.2.6 Format Data pengujian

Untuk menilai performa model yang telah dilatih, digunakan tiga metrik evaluasi utama, yaitu *Mean Absolute Error* (MAE), *Structural Similarity Index*

Measure (SSIM), dan *Colorfulness*. Setiap metrik dihitung berdasarkan rata-rata nilai yang diperoleh dari data uji. MAE mengukur rata-rata perbedaan absolut antara piksel hasil pewarnaan dengan piksel *ground truth*, sedangkan SSIM mengevaluasi tingkat kesamaan struktur, luminansi, dan kontras antara citra hasil pewarnaan dan citra referensi. Selain itu, *Colorfulness* digunakan untuk mengukur rata-rata tingkat keragaman warna yang dihasilkan oleh model. Evaluasi ini memberikan analisis kuantitatif terhadap akurasi dan kualitas pewarnaan citra *grayscale* yang dihasilkan oleh model. Hasil pengujian model akan disajikan dalam bentuk tabel berikut:

Tabel III-2 Tabel Hasil Evaluasi Pengujian Model

Konfigurasi	SSIM	MAE	<i>Colorfulness</i> (GT: X)
1			
2			
3			

Keterangan:

- SSIM (*Structural Similarity Index Measure*) semakin mendekati 1 menunjukkan kemiripan struktur, luminansi, dan kontras yang tinggi dengan *ground truth*.
- MAE (*Mean Absolute Error*) semakin kecil menunjukkan hasil pewarnaan yang semakin akurat.
- *Colorfulness* menunjukkan tingkat keragaman dan kejenuhan warna pada citra hasil pewarnaan.

3.2.7 Alat yang Digunakan dalam Penelitian

Penelitian ini dilakukan dengan menggunakan perangkat berikut:

1. Perangkat Keras:

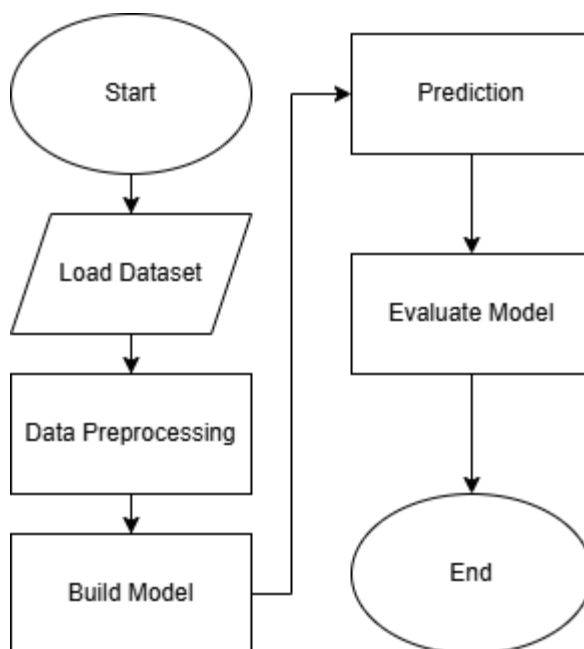
- Prosesor: *AMD Ryzen 7 5700G*
- GPU: *NVIDIA GeForce RTX 3060 12GB*
- RAM: *16GB DDR4 3000MHz*

2. Perangkat Lunak:

- Sistem Operasi: *Windows 11*
- IDE: *Visual Studio Code, Jupyter Notebook*
- Bahasa Pemrograman: *Python*

3.3 Kerangka Kerja dan Pengujian Penelitian

Proses Pengujian penelitian yang dilakukan oleh perangkat lunak dalam penelitian ini dapat dilihat pada alir Gambar III-1 dibawah ini.



Gambar III-2 Tahapan Penelitian

3.3.1 Kerangka Kerja

Kerangka kerja dalam penelitian ini dirancang untuk mengembangkan model pewarnaan citra *grayscale* berbasis *Generative Adversarial Network* (GAN) menggunakan pendekatan Pix2Pix. Kerangka ini terdiri dari beberapa tahapan utama yang disebutkan di Gambar III-2

3.3.1.1 Load Dataset

Tahap pengumpulan data dilakukan dengan memilih dataset yang sesuai untuk tugas pewarnaan citra. Dataset yang digunakan dalam penelitian ini

adalah *Linnaeus 5*, yang terdiri dari citra-citra berwarna dengan resolusi 256x256 piksel. *Dataset* ini dipilih karena keragamannya dalam representasi objek seperti hewan dan tumbuhan, yang memungkinkan pelatihan model pada berbagai pola pewarnaan.

Dataset diperoleh dari sumber terpercaya, dan setiap citra diperiksa kualitasnya untuk memastikan bahwa data bebas dari anomali seperti noise yang berlebihan atau resolusi yang tidak konsisten.

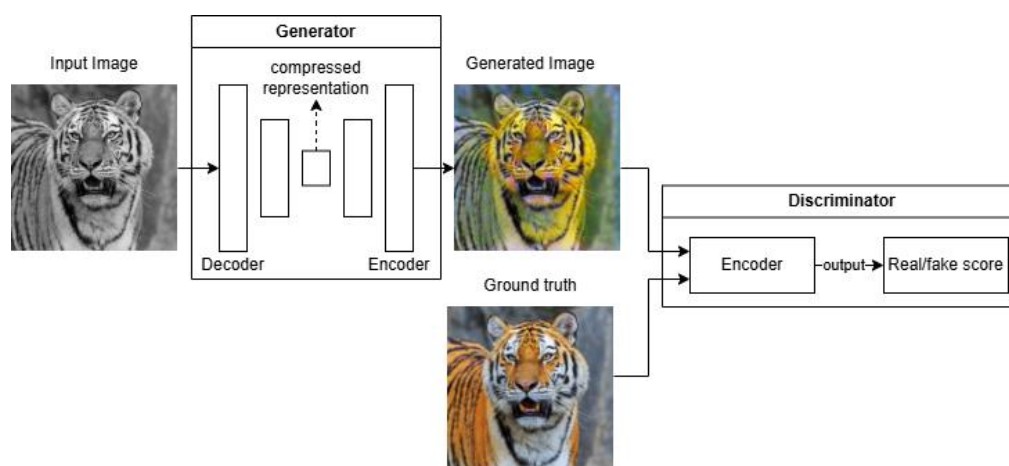
3.3.1.2 *Data Preprocessing*

Tahapan *preprocessing* bertujuan mempersiapkan citra sedemikian rupa sehingga model dapat memanfaatkannya secara optimal. Secara khusus, seluruh proses difokuskan pada perubahan citra dari ruang warna RGB ke CIELAB untuk memisahkan channel *L* (*lightness*) dan *channel a/b* (komponen warna). Tahapan lengkapnya meliputi:

1. Perubahan Ukuran Citra: Seluruh citra di-*resize* menjadi 256×256 piksel agar input model seragam.
2. Normalisasi Data: Tiap nilai piksel dinormalisasi ke rentang [0,1], di mana *channel L* (0–100) dipetakan ke [0,1] dan *channel a/b* (–128 hingga 127) disesuaikan agar berada di sekitar [–1,1].
3. Ekstraksi *Channel Grayscale (L)*: Dari konversi CIELAB, *channel L* berfungsi sebagai input jaringan, sementara *channel a* dan *b* digunakan sebagai target untuk mempelajari pewarnaan.

3.3.1.3 Arsitektur GAN

Pada tahap ini, dilakukan perancangan arsitektur *Generator* dan *Discriminator* dengan pendekatan Pix2Pix dalam rangka membentuk model GAN (*Generative Adversarial Network*) yang berfungsi untuk tugas *image colorization*.



Gambar III-3 Desain *Pix2Pix* GAN

Dalam konteks pewarnaan citra, ruang warna *CIELAB* dimanfaatkan karena memisahkan informasi intensitas (*lightness*) dan informasi kromatisitas (saluran *a* dan *b*). Citra *grayscale* dapat dianggap sebagai representasi kanal *L* (intensitas), sedangkan kanal *a* dan *b* merefleksikan informasi warna yang perlu diprediksi oleh *Generator*.

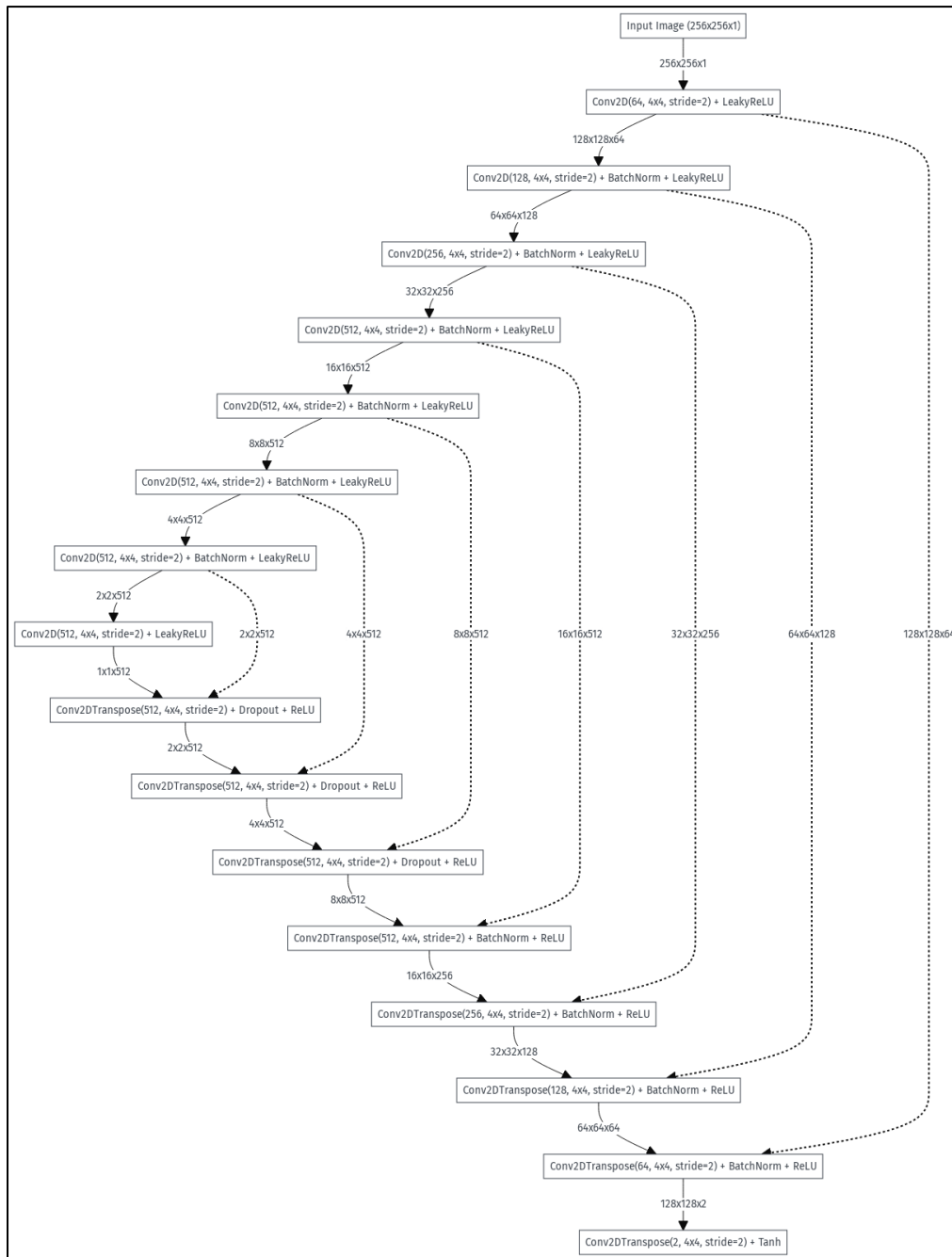
1. *Generator (U-Net)*:

Generator dirancang menggunakan arsitektur *U-Net*, yang terdiri dari:

- *Encoder* untuk mengekstraksi fitur-fitur penting dari citra input (kanal *L*).

- *Decoder* untuk merekonstruksi citra berwarna melalui prediksi kanal a/b .

Skip connections menghubungkan lapisan-lapisan encoder dengan decoder dalam arsitektur U-Net guna memastikan bahwa detail resolusi tinggi pada citra tetap terjaga. Penggunaan skip connections ini sangat penting karena mendistribusikan informasi dari lapisan-lapisan awal ke lapisan-lapisan akhir pada decoder. Dengan demikian, informasi yang lebih tajam dan lebih jelas, seperti tepi dan kontur objek, tidak hilang saat citra direkonstruksi. Hal ini sangat berpengaruh pada kualitas warna yang dihasilkan, karena warna yang dihasilkan oleh Generator menjadi lebih tajam, akurat, dan lebih realistis. Dalam konteks fitur tersebut, model tidak hanya memprediksi nilai warna berdasarkan kanal a dan b , tetapi juga belajar untuk memetakan hubungan antara intensitas pada kanal L (grayscale) dengan informasi warna pada kanal a/b , sambil tetap mempertahankan integritas struktural citra tersebut. Dengan kata lain, meskipun proses pewarnaan dilakukan pada citra grayscale, Generator berusaha untuk memahami ontologi objek pada citra tersebut, seperti bentuk, tekstur, serta perbedaan kontras antara objek yang ada. Melalui mekanisme ini, model secara efisien mentransformasikan citra grayscale menjadi citra berwarna tanpa menghilangkan informasi penting yang ada pada tahap awal citra tersebut. Dengan penerapan skip connections, redesain hasil warna dapat menghasilkan citra yang lebih konsisten dan lebih alami, karena gambar yang dihasilkan mencerminkan baik keakuratan warna maupun kualitas struktur.



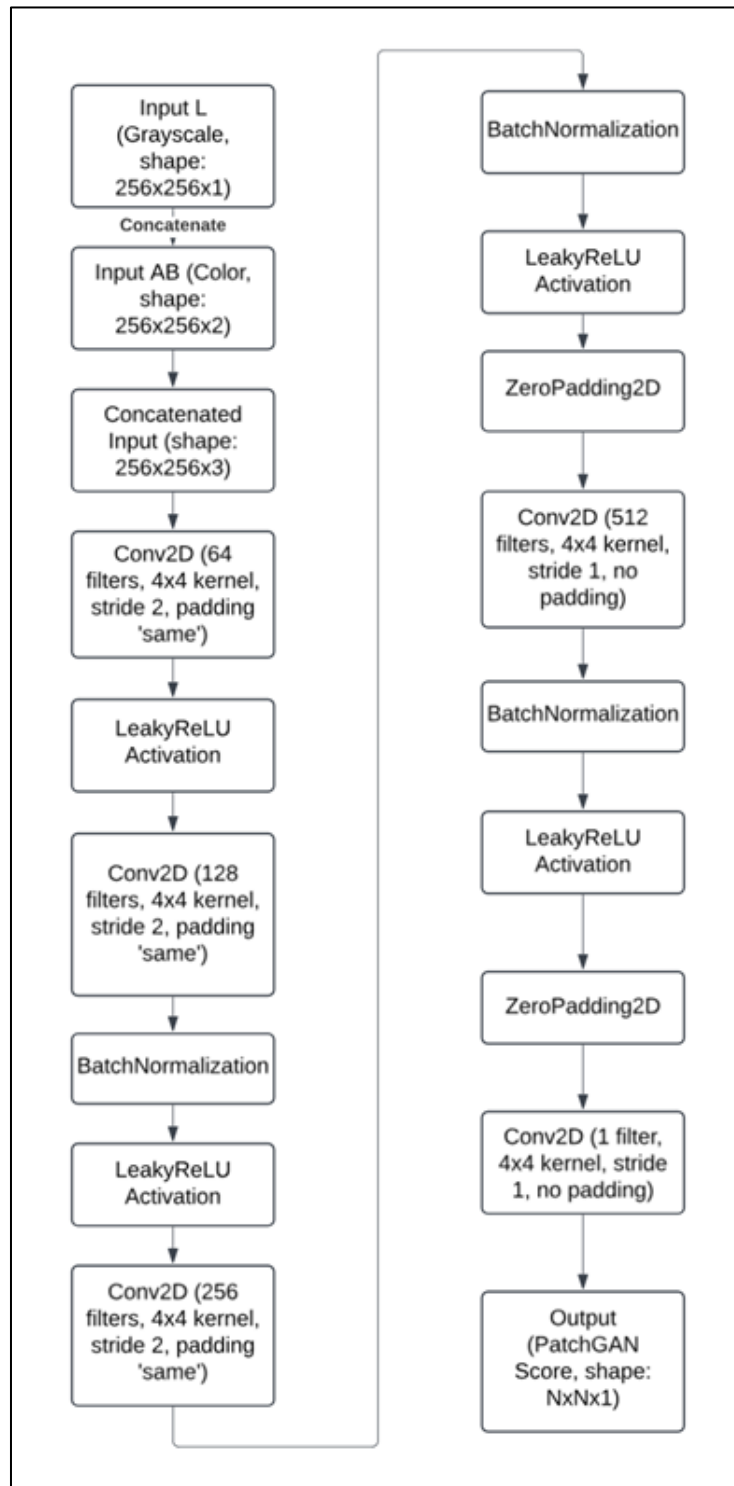
Gambar III-4 Desain Arsitektur Generator

2. *Discriminator (PatchGAN)*

Discriminator menggunakan arsitektur PatchGAN, yang berfokus pada penilaian keaslian hasil pewarnaan dengan membagi citra menjadi patch-patch kecil. Pendekatan ini sangat efektif untuk menangkap detail tekstur dan variasi lokal pada citra, sehingga setiap bagian kecil dapat dievaluasi secara mendalam.

Discriminator berfungsi sebagai klasifikator biner yang menilai apakah setiap patch pada citra berwarna merupakan data nyata (ground truth) atau palsu (output dari Generator). Dengan melakukan evaluasi secara patch-level, Discriminator tidak hanya menilai keseluruhan citra, tetapi juga memeriksa setiap bagian kecil untuk memastikan bahwa fitur-fitur lokal seperti tekstur, tepi, dan pola warna sesuai dengan karakteristik data asli.

Dengan pendekatan patch-level, Discriminator menjadi sangat peka terhadap struktur lokal, seperti tepi, kontur, dan tekstur halus, sehingga dapat memberikan umpan balik yang lebih detail. Umpan balik ini secara langsung mendorong Generator untuk menghasilkan warna yang lebih realistis dan konsisten dengan karakteristik visual aslinya.



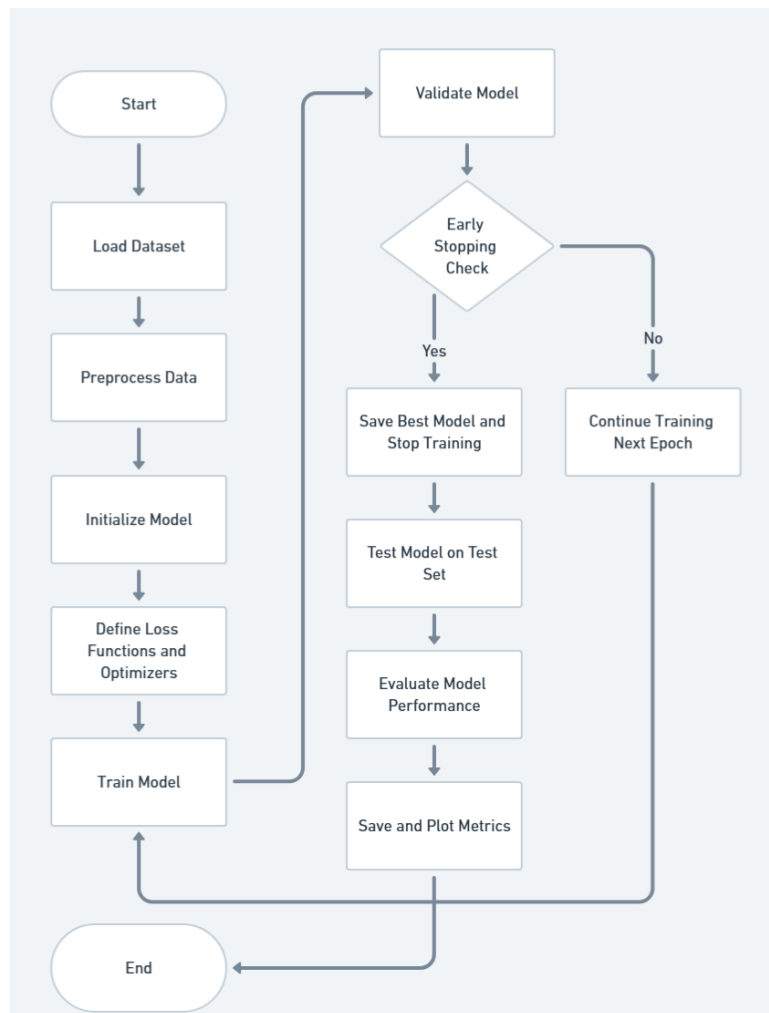
Gambar III-5 Desain Arsitektur *Discriminator*

Perancangan kedua komponen ini melibatkan penggunaan *adversarial loss*, yang mendorong *Generator* untuk menghasilkan citra yang sulit dibedakan oleh *Discriminator*, serta *L1 loss* untuk memastikan *output* tetap selaras dengan citra berwarna asli (*ground truth*). Dengan demikian, *Generator* belajar tidak hanya menipu *Discriminator*, tetapi juga menjaga kemiripan hasil dengan citra berwarna yang sesungguhnya.

3.3.1.4 Melatih Model GAN

Proses pelatihan GAN dilakukan dengan menggunakan dataset *Linnaeus 5*.

Berikut adalah langkah-langkah pelatihannya:



Gambar III-6 Proses *training*

1. *Persiapan Dataset*: Dataset dipecah menjadi data pelatihan dan pengujian dengan 1200 gambar training dan 400 gambar testing per kelas.
2. *Training Loop*: *Generator* menghasilkan citra berwarna dari input *grayscale*, sementara *Discriminator* mengevaluasi citra tersebut dengan membandingkannya dengan data asli.

3. *Epoch*: Pelatihan dilakukan hingga model mencapai kestabilan pada kesetimbangan *Nash*, dengan jumlah *epoch* yang disesuaikan berdasarkan tingkat konvergensi *loss*.

3.3.1.5 Evaluasi Hasil

Model GAN yang telah dilatih dievaluasi menggunakan dua metrik utama:

1. *Mean Absolute Error* (MAE): Mengukur rata-rata perbedaan absolut antara piksel pada citra hasil pewarnaan dan *ground truth*. Nilai MAE yang lebih kecil menunjukkan kesesuaian yang lebih baik.
2. *Structural Similarity Index Measure* (SSIM): Mengukur kemiripan struktur antara citra hasil pewarnaan dengan citra *ground truth*. Nilai SSIM berkisar antara 0 hingga 1, di mana nilai lebih mendekati 1 menunjukkan kesamaan visual yang tinggi.
3. *Colorfulness*: Mengukur perbedaan kekayaan warna antara citra hasil pewarnaan. Nilai *Colorfulness* dihitung sebagai tingkat kekayaan warna prediksi. Nilai yang lebih kecil menunjukkan bahwa warna citra hasil lebih berwarna.

Evaluasi dilakukan pada data pengujian untuk memastikan kemampuan generalisasi model terhadap data baru.

3.3.1.6 Pengembangan Perangkat Lunak

Pada tahap ini, Peneliti mengembangkan perangkat lunak yang bertujuan untuk menjadi antar muka dalam proses pewarnaan citra menggunakan model yang telah dibuat.

3.3.2 Pengujian Penelitian

Pengujian penelitian dilakukan untuk mengevaluasi performa model *Generative Adversarial Network* (GAN) berbasis *Pix2Pix* dalam pewarnaan citra grayscale menggunakan dataset *Linnaeus 5*. Pengujian melibatkan pembagian data menjadi 80% untuk pelatihan dan 20% untuk pengujian, dengan evaluasi hasil pewarnaan pada data uji berdasarkan metrik *Mean Absolute Error* (MAE), *Structural Similarity Index Measure* (SSIM) dan *colorfulness*.

Proses pengujian dilakukan dengan membandingkan citra hasil pewarnaan dengan ground truth untuk menilai akurasi dan kualitas pewarnaan secara kuantitatif. Pengujian juga mencakup analisis visual untuk memeriksa konsistensi warna dan detail citra, memastikan hasil yang realistis dan sesuai dengan persepsi manusia.

3.3.3 Analisis Hasil Pengujian dan Membuat Kesimpulan

Setelah pengujian dilakukan, langkah berikutnya adalah menganalisis hasil pengujian dan menarik kesimpulan. Analisis dilakukan dengan memeriksa tabel hasil pengujian berdasarkan aspek-aspek evaluasi yang telah ditentukan. Langkah ini bertujuan untuk mengevaluasi performa model secara mendalam

yang mampu menghasilkan pewarnaan citra *grayscale* dengan tingkat akurasi dan kualitas visual tertinggi.

3.4 Metode Pengembangan Perangkat Lunak

Pada penelitian ini metode pengembangan perangkat lunak yang digunakan oleh peneliti adalah *Gantt Chart*. Metode pengembangan perangkat lunak pada penelitian ini dapat dilihat pada Tabel III-3.

Tabel III-2 Timeline RUP.

Tahap	Waktu Kegiatan (Dalam Minggu)																							
	Jul				Agu				Sep				Okt				Nov				Des			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Insepsi	■	■	■	■																				
Elaborasi					■	■	■	■																
Observasi									■	■	■	■												
Konstruksi													■	■	■	■								
Transisi																	■	■	■	■				
Penyelesaian																					■	■	■	■

Tahap	Deskripsi	Durasi (Minggu)	Target Output
1. Insepsi	Mengidentifikasi kebutuhan sistem dan pengguna, serta perancangan awal kerangka kerja penelitian.	4	Dokumen kebutuhan dan kerangka kerja penelitian.

2. Elaborasi	Membuat desain sistem, seperti diagram alur kerja (flowchart), use case, dan arsitektur model.	4	Diagram alur kerja, use case, dan rancangan model.
3. Konstruksi	Implementasi model GAN berbasis Pix2Pix, pelatihan model, dan pengembangan perangkat lunak.	4	Model GAN terlatih dan perangkat lunak siap diuji.
4. Transisi	Pengujian sistem dengan dataset pengujian, evaluasi model, dan analisis hasil.	4	Laporan pengujian dan evaluasi performa model.
5. Penyelesaian	Penyusunan laporan akhir dan dokumentasi sistem.	4	Laporan akhir penelitian dan dokumentasi sistem.