

## **BAB II**

### **KAJIAN LITERATUR**

#### **2.1 Pendahuluan**

Bab ini berisi penguraian landasan teori yang relevan dengan permasalahan penelitian, penelitian terdahulu yang berkaitan dengan penelitian yang sedang dikembangkan, serta hipotesis penelitian yang diajukan

#### **2.2 Landasan Teori**

##### **2.2.1 *Generative Adversarial Network***

Pembuatan citra dari sebuah dataset menjadi bidang penelitian yang menarik di *Artificial Intelligence*. Sebuah model yang mampu membuat data baru dari distribusi data awal disebut dengan *generative model*. Alasan kenapa sekarang manusia mulai mengadopsi teknologi pembuatan gambar tersebut adalah karena

dengan adanya teknologi ini mampu mengurangi kebutuhan tenaga kerja manusia dalam pembuatan konten industri seperti *game*. Karena pembuatan gambar secara manual akan menimbulkan biaya yang tinggi dan kelambatan dalam pekerjaan (Nuha & Afiahayati, 2018).

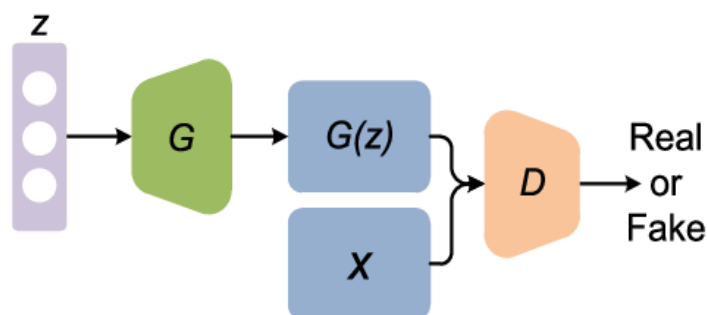
Pada tahun 2014, kerangka kerja *Generative Adversarial Network* (GAN) telah ditemukan oleh Goodfellow dkk., kerangka kerja tersebut memperkirakan *generative model* melalui proses *adversarial* atau permusuhan antara dua *neural network* yang bersaing satu sama lain. Jaringan-jaringan itu disebut *Generator*

dan Diskriminator, yang dapat dianalogikan dengan *Generator* adalah pembuat uang palsu sementara Diskriminator adalah polisi yang mencoba mendeteksi apakah uang itu palsu (dibuat oleh *Generator*) atau nyata (diproduksi oleh data latihan) (Nuha & Afiahayati, 2018).

### 2.2.2 Prinsip GAN

GAN terinspirasi oleh teori permainan yaitu generator  $G$  dan diskriminator  $D$  akan saling berkompetisi satu sama lain untuk mencapai kesetimbangan Nash pada proses pelatihan. *Input* dari *Generator* adalah *random noise vector*  $z$  (biasanya sebuah distribusi seragam atau normal). *Noise* dipetakan ke ruang data baru melalui *generator*  $G$  untuk mendapatkan sampel palsu,  $G(z)$ , yang merupakan vektor *multi-dimensional*. Dan, diskriminator  $D$  adalah sebuah *binary classifier*, dibutuhkan sampel nyata dari dataset dan sampel palsu yang dihasilkan oleh generator  $G$  sebagai *input*, dan output dari diskriminator  $D$  mewakili probabilitas bahwa sampel tersebut adalah nyata daripada sampel palsu. Ketika diskriminator  $D$  tidak dapat menentukan apakah data berasal dari dataset nyata atau generator, keadaan optimal tercapai. Pada

titik ini, kita memperoleh model *generator*  $G$ , yang telah mempelajari distribusi data nyata. Arsitektur GAN diilustrasikan pada Gambar II-1.



**Gambar II-1** Arsitektur GAN (Pan dkk., 2019)

### 2.2.3 Model Pembelajaran dari GAN (*Learning Model*)

Dalam pembelajaran GAN, *Generator* dan diskriminator memiliki fungsi kehilangan (loss functions) mereka sendiri. Dalam hal ini, kami menyebutnya  $J(G)$  dan  $J(D)$ . Diskriminator  $D$  didefinisikan sebagai pengklasifikasi biner, dan fungsinya diwakili oleh cross entropy

$$J(D) = -E_{x \sim p_{\text{data}}(x)} [\log D(x)] - E_{z \sim p_z(z)} [\log (1 - D(G(z)))] \quad (2.1)$$

di mana  $x$  mewakili sampel nyata,  $z$  mewakili *random noise vector*,  $G(z)$  adalah data yang dihasilkan oleh generator, dan  $E$  mewakili ekspektasi.  $D(x)$  menunjukkan probabilitas bahwa  $D$  membedakan  $x$  sebagai data nyata, dan  $D(G(z))$  menunjukkan probabilitas bahwa  $D$  menentukan data yang dihasilkan oleh  $G$ . Tujuan  $D$  adalah untuk menentukan sumber data dengan benar, sehingga ia ingin  $D(G(z))$  mendekati 0, sedangkan tujuan  $G$  adalah untuk mendekatkannya ke 1. Berdasarkan ide ini, terdapat konflik antara dua model

ini (yaitu, *zero-sum game*). Oleh karena itu, fungsi kehilangan dari *generator* dapat diturunkan oleh diskriminator:

$$J(G) = -E_{z \sim p_z(z)} [\log D(G(z))] \quad (2.2)$$

Akibatnya, masalah optimasi GAN ditransformasikan ke dalam game minimax seperti yang ditunjukkan di bawah ini,

$$\min_G \max_D V(D, G) = E_{x \sim p_{\text{data}}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log (1 - D(G(z)))] \quad (2.3)$$

Dalam proses pelatihan, parameter dalam G diperbarui bersama dengan proses pembaruan parameter dalam D. Ketika  $D(G(z)) = 0,5$ , diskriminator tidak dapat menentukan perbedaan antara dua distribusi ini, dan dalam status ini, model akan mencapai solusi optimal global (Pan dkk., 2019).

#### 2.2.4 Model-model yang Diturunkan dari GAN

Karena kekurangan dari model GAN yang asli, berbagai model GAN turunan diusulkan oleh beberapa peneliti, dan model GAN turunan ini dapat diklasifikasikan ke dalam dua kelompok, GAN berbasis optimasi arsitektur, dan GAN berbasis optimasi fungsi objektif, seperti yang ditunjukkan pada Tabel II-1.

**Tabel II-1** Klasifikasi dari model GAN (Pan dkk., 2019)

<i>Architecture Optimization Based GANs</i>	<i>Convolution based GANs</i>	<i>DCGAN</i>
	<i>Condition based GANs</i>	<i>CGANs; InfoGAN; ACGAN</i>

	<i>AutoEncoder based GANs</i>	<i>AAE; BiGAN; ALI; AGE; VAE-GAN</i>
<i>Obective Function Optimization Based GANs</i>	<i>Unrolled GAN; f-GAN; Mode-Regularized GAN; Least-Square GAN; Loss-Sensitive GAN; EBGAN; WGAN; WGAN-GP; WGAN-LP</i>	

### 2.2.5 Citra Grayscale

Citra *grayscale* adalah representasi visual yang hanya terdiri dari skala warna abu-abu, dengan nilai intensitas berkisar dari 0 (hitam) hingga 255 (putih) pada sistem 8-bit. Tidak seperti citra berwarna yang memiliki tiga saluran (RGB), citra *grayscale* hanya memiliki satu saluran intensitas. Representasi ini digunakan secara luas dalam berbagai aplikasi pengolahan citra, seperti deteksi tepi, segmentasi, dan pemrosesan awal untuk analisis data visual (Gonzalez & Woods, 2018).

Citra *grayscale* biasanya diperoleh melalui proses konversi dari citra berwarna dengan menghitung rata-rata atau pembobotan nilai intensitas di setiap saluran warna. Persamaan 2.1 digunakan untuk untuk konversi dari RGB ke grayscale (Schanda, 2007):

$$Y=0.2989 \cdot R+0.5870 \cdot G+0.1140 \cdot B \quad (2.4)$$

Di mana  $Y$  adalah intensitas grayscale, dan  $R$ ,  $G$ , serta  $B$  adalah nilai intensitas dari saluran merah, hijau, dan biru. Pembobotan ini mencerminkan sensitivitas mata manusia terhadap masing-masing saluran warna, dengan hijau memiliki pengaruh terbesar (Schanda, 2007).

## 2.2.6 CIELAB

CIELAB adalah ruang warna yang juga dikenal sebagai “CIE Lab\*” atau “Ruang warna Lab.” Diciptakan pada tahun 1976 oleh Komisi Internasional untuk Penerangan (CIE), ruang warna ini mengekspresikan warna sebagai tiga nilai komponen utama (Schanda, 2007; Wyszecki & Stiles, 2000). Elemen utama ruang warna CIELAB adalah sebagai berikut:

- L\*: Mewakili kecerahan, dengan nilai 0 sebagai hitam sempurna (0% pantulan atau transmisi). Nilai 50% mengindikasikan abu-abu tengah, sedangkan nilai 100 menunjukkan warna putih sempurna, yang mengindikasikan 100% pantulan atau kejernihan yang sempurna.
- a\*: Merepresentasikan kemerahan-kelabu warna. Nilai positif dari a\* mengindikasikan merah, sedangkan nilai negatif mengindikasikan hijau. Nilai 0 adalah netral (tidak ada kecenderungan ke merah atau hijau).
- b\*: Menunjukkan warna kuning-kebiruan. Nilai positif dari b\* menunjukkan kuning, sedangkan nilai negatif menunjukkan biru. Nilai 0 menunjukkan netralitas tanpa kecenderungan ke kuning atau biru.

Nilai a\* dan b\* biasanya berada dalam rentang [-128, 127], memungkinkan representasi warna yang sangat luas dan fleksibel dalam berbagai aplikasi (Sharma, 2002).

### 2.2.6.1 Grayscale ke Ruang Warna CIELAB

Konversi citra *grayscale* ke ruang warna CIELAB adalah langkah penting dalam proses pewarnaan otomatis. Ruang warna CIELAB dirancang

untuk lebih mendekati persepsi visual manusia dibandingkan dengan model seperti RGB. Dalam model CIELAB:

- Komponen  $L^*$  merepresentasikan kecerahan, yang langsung dapat dihubungkan dengan intensitas *grayscale*.
- Komponen  $a^*$  dan  $b^*$ , yang merepresentasikan informasi warna pada spektrum merah-hijau dan kuning-biru, ditambahkan untuk memperkaya informasi warna yang tidak dimiliki oleh citra grayscale.

Proses konversi dari grayscale ke CIELAB melibatkan transformasi nilai intensitas *grayscale* menjadi komponen  $L^*$ , sementara nilai awal untuk  $a^*$  dan  $b^*$  biasanya diinisialisasi sebagai netral (0). Setelah proses pewarnaan, model pewarnaan seperti Pix2Pix conditional GAN akan menghasilkan nilai baru untuk komponen  $a^*$  dan  $b^*$  yang melengkapi citra berwarna (Zhang dkk., 2017).

#### **2.2.6.2 Keunggulan pendekatan Grayscale ke CIELAB**

Karena ruang warna CIELAB dirancang dalam tiga dimensi ( $Lab^*$ ), model ini mampu merepresentasikan warna dalam jumlah yang hampir tak terbatas, memberikan fleksibilitas tinggi dalam berbagai aplikasi visual. Salah satu keunggulan utama CIELAB adalah sifatnya yang disebut perceptual uniformity (keseragaman persepsi), di mana perubahan nilai dalam ruang warna ini mencerminkan perubahan yang konsisten dengan cara manusia memandang warna (Schanda, 2007; Wyszecki & Stiles, 2000).

Hal ini membuat model CIELAB jauh lebih akurat dalam mendekati persepsi visual manusia dibandingkan model warna lain seperti RGB dan CMYK, yang memiliki keterbatasan dalam menginterpretasikan gradasi warna berdasarkan sensitivitas mata manusia. Misalnya, model RGB sering kali tidak mampu mencerminkan variasi kecerahan secara konsisten, sementara CMYK lebih berfokus pada kombinasi untuk pencetakan tanpa mempertimbangkan persepsi warna secara alami (Sharma, 2002). Dengan sifat ini, CIELAB sering digunakan dalam aplikasi pencetakan, pengolahan gambar, dan desain, di mana ketepatan reproduksi warna sangat penting.

### **2.2.7 *Pix2Pix conditional GAN***

Pendekatan *Pix2Pix* yang diperkenalkan oleh Isola dkk. (2017) merupakan model *conditional Generative Adversarial Network* (cGAN) yang dirancang untuk tugas *image-to-image translation*. Berbeda dengan GAN tradisional yang menghasilkan data dari *random noise*, *Pix2Pix* menggunakan pasangan input-output sebagai data pelatihan untuk mempelajari transformasi dari satu domain ke domain lain. Contohnya adalah transformasi dari citra *grayscale* ke citra berwarna, dari sketsa ke gambar realistis, atau dari peta ke gambar satelit (Isola dkk., 2017).

#### **2.2.7.1 Arsitektur Model *Pix2Pix***

Pada pendekatan ini, generator dirancang sebagai jaringan berbasis U-Net, yang terdiri dari lapisan *encoder* dan *decoder*. Lapisan *encoder* bertugas untuk mengekstraksi fitur dari citra input, sementara lapisan *decoder* bertugas



membangun ulang citra dengan menambahkan elemen-elemen dari domain target (Ronneberger dkk., 2015). Model ini memanfaatkan skip connections untuk mempertahankan informasi penting dari lapisan awal selama proses decoding.

Sebaliknya, *discriminator* pada *Pix2Pix* menggunakan arsitektur PatchGAN, yang dirancang untuk mengevaluasi setiap bagian kecil (*patch*) dari citra secara lokal. Pendekatan ini memungkinkan *discriminator* lebih fokus pada pola tekstur lokal daripada keseluruhan gambar, sehingga lebih efektif dalam menghasilkan citra yang realistis (Isola dkk., 2017).

#### **2.2.7.2 Fungsi Kehilangan (*Loss Function*)**

Model *Pix2Pix* menggunakan kombinasi dari dua fungsi kehilangan utama:

1. LSGAN *Loss*: LSGAN (*Least Squares GAN*) merupakan varian dari *Adversarial Loss* yang menggunakan fungsi kuadrat untuk mengukur selisih antara keluaran diskriminator  $D(G(z))$  dan nilai target (0 atau 1), sehingga membantu mengurangi *gradient saturation* yang sering terjadi pada GAN konvensional. Pendekatan ini memacu generator untuk menghasilkan keluaran dengan *fake score* mendekati 1 agar sulit dibedakan oleh diskriminator (Mao dkk., 2017). Dalam *Conditional GAN*, LSGAN *Loss* memfasilitasi pemetaan langsung dari citra *input* (L channel) ke citra berwarna (*A/B channel*) yang semakin mendekati citra rujukan.

2. *L1 Loss*: Fungsi kehilangan ini digunakan untuk mengurangi rata-rata perbedaan absolut antara citra hasil generasi (prediksi warna oleh *generator*) dengan citra target (*ground truth*). Dengan menggunakan *L1 Loss*, *generator* belajar memprediksi warna yang sesuai dengan citra asli, sehingga menghasilkan gambar yang lebih realistis dan akurat. Kombinasi dari *Adversarial Loss* dan *L1 Loss* memungkinkan model *Pix2Pix* tidak hanya menghasilkan warna yang realistis tetapi juga memastikan bahwa pewarnaan tersebut konsisten dan sesuai dengan gambar asli.

#### **2.2.7.3 Aplikasi Pix2Pix**

Pendekatan *Pix2Pix* telah diaplikasikan secara luas dalam berbagai bidang, termasuk:

- Pewarnaan Citra: Mengubah citra *grayscale* menjadi citra berwarna secara otomatis dengan kualitas tinggi (Zhang dkk., 2017).
- Medis: Untuk transformasi citra medis, seperti konversi dari data CT ke MRI, yang membantu dokter dalam diagnosis lebih akurat (Nie dkk., 2018).
- Desain Visual: Menghasilkan gambar realistis dari sketsa atau rancangan awal dalam bidang desain dan seni digital (Wang dkk., 2018).

#### **2.2.7.4 Keunggulan dan Keterbatasan Model Pix2Pix**

Keunggulan utama *Pix2Pix* adalah kemampuannya untuk belajar dari data yang sudah dipasangkan (*paired dataset*), yang memungkinkan model

menghasilkan transformasi yang lebih spesifik dan terarah. Namun, keterbatasan utamanya adalah kebutuhan akan data yang teranotasi dengan baik, yang dapat menjadi tantangan dalam skenario di mana data *paired* sulit diperoleh (Zhu dkk., 2017).

### **2.2.8 Pengukuran Kinerja Model**

Evaluasi hasil pewarnaan citra membutuhkan metrik yang dapat menggambarkan kualitas hasil generasi dengan baik. Dalam penelitian pewarnaan citra *grayscale*, terdapat beberapa metrik yang sering digunakan untuk mengukur kinerja model. Dua metrik utama yang digunakan adalah *Structural Similarity Index Measure* (SSIM) dan *Mean Absolute Error* (MAE). SSIM digunakan untuk mengukur kesamaan struktural antara hasil pewarnaan dengan *ground truth*, mencakup aspek luminansi, kontras, dan struktur visual secara keseluruhan. Sementara itu, MAE digunakan untuk menghitung rata-rata kesalahan absolut antara nilai piksel hasil generasi dan *ground truth*, sehingga memberikan gambaran kuantitatif tentang tingkat akurasi warna.

Selain SSIM dan MAE, *colorfulness* metric telah diusulkan sebagai metrik tambahan untuk mengevaluasi intensitas dan variasi warna dalam hasil pewarnaan. Metrik ini mengukur seberapa hidup (*vivid*) warna yang dihasilkan oleh model, yang relevan untuk menginterpretasikan kualitas visual hasil pewarnaan dari perspektif estetika. Dalam penelitian oleh Žeger dkk. (2021), penggunaan *colorfulness* metric disarankan untuk melengkapi evaluasi kuantitatif, karena metrik ini dapat memberikan wawasan tambahan tentang

performa model dalam menghasilkan warna yang realistis dan menarik secara visual.

Pengukuran kinerja model dengan metrik-metrik ini memberikan evaluasi yang komprehensif, menggabungkan aspek numerik dan visual dalam menilai kualitas hasil pewarnaan citra grayscale.

### 2.2.8.1 *Structural Similarity Index Measure (SSIM)*

SSIM adalah metrik yang digunakan untuk mengukur kesamaan antara dua citra berdasarkan struktur, *luminance*, dan *contrast*. SSIM memiliki rentang nilai dari 0 hingga 1, di mana nilai mendekati 1 menunjukkan bahwa citra hasil generasi sangat mirip dengan *ground truth*.

SSIM dihitung menggunakan rumus berikut:

$$SSIM_{(x,y)} = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (2.5)$$

Di mana:

- $\mu_x$  dan  $\mu_y$ : Rata-rata intensitas dari citra x dan y.
- $\sigma_x^2$  dan  $\sigma_y^2$ : Variansi intensitas dari citra x dan y.
- $\sigma_{xy}$ : Kovarians antara x dan y.
- $C_1$  dan  $C_2$ : Konstanta kecil untuk stabilitas perhitungan.

Metrik SSIM digunakan secara luas karena mampu menangkap kesamaan struktur, yang menjadi aspek penting dalam tugas pewarnaan citra (Zhang dkk., 2017).

### 2.2.8.2 Mean Absolute Error (MAE)

MAE adalah metrik yang menghitung rata-rata kesalahan absolut antara piksel hasil pewarnaan dan *ground truth*. Nilai MAE yang lebih kecil menunjukkan tingkat kesalahan yang rendah, yang berarti hasil pewarnaan mendekati warna asli.

MAE dihitung menggunakan rumus berikut:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (2.6)$$

Di mana:

- $y_i$ : Nilai piksel pada ground truth.
- $\hat{y}_i$ : Nilai piksel pada hasil pewarnaan.
- $N$ : Jumlah piksel dalam citra.

MAE digunakan untuk mengukur akurasi numerik pewarnaan, yang menjadi pelengkap metrik SSIM untuk mengevaluasi kualitas visual dan struktural (Wang dkk., 2004).

### 2.2.8.3 Colorfulness

*Colorfulness metric* digunakan untuk mengukur intensitas dan variasi warna dalam citra digital. Formula yang dikembangkan oleh Hasler dan Suesstrunk (2003) adalah

$$C = \sqrt{\sigma_r^2 + \sigma_b^2} + 0.3 \cdot \sqrt{\mu_r^2 + \mu_b^2} \quad (2.7)$$

, di mana

$$r = R - G \quad (2.8)$$

dan

$$b = B - Y - Y \quad (2.8)$$

mewakili selisih warna antara saluran merah (R), hijau (G), dan biru (B) terhadap luminansi (Y), dengan

$$Y = 0.5 \cdot (R + G) \quad (2.9)$$

Komponen  $\sigma_r$  dan  $\sigma_b$  adalah standar deviasi yang menunjukkan variasi warna, sedangkan  $\mu_r$  dan  $\mu_b$  adalah rata-rata yang menunjukkan intensitas warna. Metrik ini digunakan untuk mengevaluasi hasil pewarnaan citra, terutama untuk memastikan warna yang dihasilkan cukup hidup (*vivid*) dan menarik secara visual.

### 2.3 Penelitian Lain yang Relevan

Pada bagian ini, dirangkum studi sebelumnya yang relevan dengan pengembangan model *Generative Adversarial Networks* (GAN) untuk pewarnaan citra *grayscale*. Penelitian-penelitian ini memberikan wawasan tentang teknik, arsitektur, dan performa model dalam menangani tugas pewarnaan citra secara otomatis.

Salah satu penelitian yang signifikan dilakukan oleh Zhang dkk. (2017), yang memperkenalkan metode *deep learning* untuk pewarnaan citra otomatis. Model ini menggunakan arsitektur *convolutional neural network* (CNN) yang dirancang untuk mempelajari pola pewarnaan dari dataset gambar berwarna.

Penelitian ini berhasil menghasilkan pewarnaan citra yang realistis dan mendekati warna asli dengan struktur visual yang konsisten.

Kemudian, penelitian oleh Isola dkk. (2017) memperkenalkan *Pix2Pix conditional GAN*, model yang dirancang khusus untuk *task image-to-image translation*, termasuk pewarnaan citra *grayscale*. *Pix2Pix* menggunakan pendekatan data yang dipasangkan (*paired dataset*) untuk mempelajari transformasi dari gambar *grayscale* ke citra berwarna. Hasil penelitian ini menunjukkan bahwa *Pix2Pix* mampu mempertahankan detail visual dan menghasilkan pewarnaan yang realistis pada berbagai domain gambar.

Penelitian lain dilakukan oleh Ricky dan Al Rivan (2022), yang memanfaatkan GAN untuk pewarnaan citra menggunakan dataset *Places365*. Model yang dikembangkan dalam penelitian ini berhasil mencapai nilai SSIM (*Structural Similarity Index Measure*) sebesar 68,32% dan MAE (*Mean Absolute Error*) sebesar 10,81. Studi ini menunjukkan keunggulan GAN dalam menghasilkan pewarnaan yang mendekati ground truth dengan kualitas yang baik.

Selain itu, Nie dkk. (2018) menggunakan pendekatan *Pix2Pix* dalam domain citra medis, seperti konversi data CT ke MRI. Penelitian ini memperlihatkan fleksibilitas model *conditional GAN* dalam menghasilkan transformasi citra yang akurat pada domain selain pewarnaan citra umum. Hasil penelitian ini menguatkan bahwa GAN dapat diaplikasikan pada berbagai tugas transformasi citra.

Penelitian oleh Zhu dkk. (2017) memperkenalkan *CycleGAN*, sebuah arsitektur GAN yang dirancang untuk transformasi citra tanpa anotasi data (unpaired dataset). Model ini mampu menghasilkan transformasi gaya dan pewarnaan citra secara konsisten meskipun tidak menggunakan data yang dipasangkan. Pendekatan ini memberikan alternatif yang menarik untuk skenario di mana anotasi data tidak tersedia.

Studi-studi ini menjadi dasar penting dalam pengembangan model GAN, terutama *Pix2Pix conditional GAN*, untuk tugas pewarnaan citra *grayscale*. Hasil-hasil tersebut menunjukkan efektivitas GAN dalam menghasilkan pewarnaan citra yang lebih realistis, efisien, dan akurat dibandingkan metode tradisional.

## **2.4 Kesimpulan**

Kajian literatur ini memberikan gambaran tentang teori dasar, model, serta penelitian terkait dalam pewarnaan citra berbasis GAN. *Generative Adversarial Networks (GAN)*, khususnya model *Pix2Pix conditional GAN*, telah terbukti efektif dalam menghasilkan pewarnaan citra dengan kualitas tinggi. Ruang warna CIELAB digunakan untuk memisahkan komponen luminansi dan warna, yang mendukung proses pewarnaan citra secara lebih akurat. Selain itu, berbagai penelitian terkait mendukung bahwa pendekatan berbasis GAN dapat meningkatkan efisiensi dan kualitas pewarnaan dibandingkan metode tradisional. Berdasarkan kajian ini, penelitian lebih lanjut



dapat difokuskan pada optimalisasi arsitektur model dan penggunaan *dataset* yang lebih beragam untuk meningkatkan hasil pewarnaan.