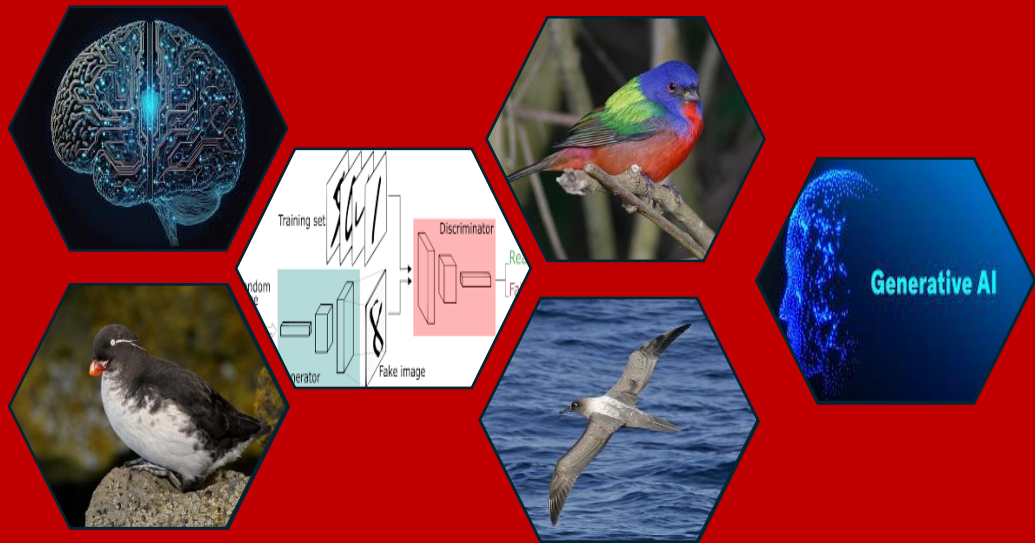


MODEL FASTGAN UNTUK SINTESIS CITRA DALAM DOMAIN ORNITOLOGI



FAKHIRAZZAHRAH RAMADHANI

D121211040



PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

MAKASSAR

2026

BAB I

PENDAHULUAN

1.1 Latar belakang

Perkembangan kecerdasan buatan (*Artificial Intelligence* atau AI) dalam beberapa tahun terakhir telah membawa perubahan signifikan dalam berbagai bidang, salah satunya adalah pembuatan data sintesis melalui AI generatif. AI generatif merupakan cabang AI yang berfokus pada pembuatan data baru seperti gambar, teks, atau suara berdasarkan pola dan distribusi data yang telah dipelajari (Goodfellow et al., 2014). Di bidang pengolahan citra, kemampuan AI untuk menghasilkan gambar yang realistis sangat bermanfaat untuk mendukung berbagai aplikasi, mulai dari industri kreatif hingga penelitian ilmiah yang memerlukan simulasi data dan visualisasi kompleks dalam konteks pembelajaran mesin (Amelia Agustina Herlijanto, 2024).

Generative Adversarial Networks (GAN) adalah model pembelajaran mesin yang terdiri dari dua jaringan saraf yang saling berkompetisi: Generator yang bertugas membuat data sintesis, dan Discriminator yang bertugas membedakan data asli dari data sintesis. Model GAN digunakan untuk menghasilkan data sintesis berkualitas tinggi, khususnya citra. Metode ini pertama kali diperkenalkan oleh Goodfellow et al. (2014) yang memberikan fondasi bagi perkembangan berbagai varian GAN selanjutnya. Meskipun handal, GAN konvensional menghadapi tantangan seperti pelatihan tidak stabil, *mode collapse*, dan kebutuhan data dalam jumlah besar (Arjovsky et al., 2017).

Berbagai varian GAN kemudian dikembangkan untuk mengatasi keterbatasan tersebut. StyleGAN yang diperkenalkan oleh Karras et al. (2019) mengimplementasikan pendekatan *style-based generation* yang memungkinkan kontrol detail citra pada berbagai tingkat resolusi. Namun, StyleGAN masih memiliki keterbatasan berupa kompleksitas arsitektur tinggi, waktu pelatihan yang panjang, dan kebutuhan data yang besar (H. Zhang et al., 2019). Progressive GAN (Tero Karras et al., 2018) dan BigGAN (Brock et al., 2019) juga menunjukkan hasil berkualitas tinggi, namun tetap memerlukan dataset pelatihan berukuran puluhan hingga ratusan ribu gambar, yang menjadi hambatan untuk penerapan pada domain dengan keterbatasan data.

Menjawab berbagai keterbatasan tersebut, Liu et al. (2021) mengembangkan FastGAN, solusi yang lebih efisien untuk sintesis citra dari data terbatas (*few-shot image synthesis*). FastGAN mengintegrasikan teknik *skip-layer excitation* (SLE) (mekanisme yang memfasilitasi aliran informasi multi-skala antar lapisan dengan berbagai resolusi) dan *self-supervised discriminator* (pendekatan yang meningkatkan kemampuan discriminator melalui tugas rekonstruksi tambahan selain klasifikasi asli atau sintesis). Model ini dirancang untuk dapat dilatih menggunakan dataset berukuran ratusan hingga ribuan gambar tanpa mengorbankan kualitas hasil sintesis secara signifikan. Kemampuan ini menjadikannya kandidat ideal untuk domain dengan keterbatasan data seperti ornitologi (Liu et al., 2021; Noguchi & Harada, 2019).

Evaluasi FastGAN dalam penelitian asli mencakup berbagai domain seperti wajah manusia, karya seni, kartun, dan objek dengan dataset berukuran 60 hingga 1000 gambar (Liu et al., 2021). Hasil evaluasi menunjukkan bahwa performa model sangat bervariasi tergantung pada karakteristik domain. Domain dengan struktur konsisten dan variasi terbatas seperti wajah manusia cenderung menghasilkan sintesis berkualitas lebih tinggi, sementara domain dengan variasi visual tinggi dan struktur yang tidak seragam menunjukkan tantangan yang lebih besar. Perbedaan performa ini mengindikasikan bahwa kompleksitas dan karakteristik visual domain memberikan pengaruh signifikan terhadap kualitas sintesis citra yang dihasilkan (Liu et al., 2021).

Meskipun telah dievaluasi pada berbagai domain, penelitian asli FastGAN belum mencakup domain citra biologis, yang memiliki karakteristik unik dan tantangan tersendiri. Salah satu tantangan utama dalam sintesis gambar pada domain biologis adalah menangkap karakteristik visual yang spesifik dan detail morfologis yang kritis (Reed et al., 2016). Model generatif yang tidak dioptimalkan untuk domain ini seringkali menghasilkan citra yang kurang akurat atau kehilangan detail penting yang dibutuhkan untuk merepresentasikan keanekaragaman hayati secara visual (Reed et al., 2016).

Domain ornitologi, khususnya melalui dataset CUB-200-2011 (*Caltech-UCSD Birds-200-2011*), menghadirkan kompleksitas yang berbeda dari domain-domain yang telah dievaluasi sebelumnya (Wah et al., 2011). Dataset ini berisi 11.788 gambar dari 200 spesies burung Amerika Utara dengan rata-rata sekitar 59 gambar per spesies (Wah et al., 2011). Kompleksitas yang dihadirkan meliputi variasi tinggi antar spesies yang berbeda (*high inter-class variation*), detail morfologis halus yang kritis untuk identifikasi spesies (*fine-grained features*) seperti pola bulu, bentuk paruh, dan tekstur sayap, serta kebutuhan akan kewajaran anatomis yang harus dipertahankan dalam sintesis (Wah et al., 2011). Karakteristik ini menciptakan tantangan unik karena model tidak hanya harus mampu menghasilkan citra yang realistis secara visual, tetapi juga harus mempertahankan struktur biologis yang benar untuk setiap spesies.

Dataset CUB-200-2011 telah digunakan secara luas dalam penelitian penglihatan komputer untuk tugas klasifikasi dan pengenalan citra detail tinggi (*fine-grained image recognition*) (Wah et al., 2011). Dataset ini dilengkapi dengan anotasi lengkap termasuk bounding box untuk lokalisasi burung, anotasi 15 bagian tubuh seperti paruh, mata, sayap, dan ekor, serta 312 atribut visual biner (Wah et al., 2011). Keberadaan anotasi komprehensif ini memungkinkan dataset tidak hanya digunakan untuk klasifikasi spesies, tetapi juga potensial untuk evaluasi apakah citra sintesis mampu mempertahankan struktur morfologis burung dengan tepat. Meskipun dataset ini telah digunakan secara ekstensif untuk berbagai tugas penglihatan komputer, penerapannya untuk sintesis citra menggunakan GAN dengan data terbatas seperti FastGAN masih terbatas.

Dengan jumlah gambar yang berada pada spektrum few-shot (rata-rata 59 gambar per spesies), dataset CUB-200-2011 secara teoritis sesuai dengan desain FastGAN (Liu et al., 2021; Wah et al., 2011). Namun, keragaman 200 spesies burung dengan karakteristik morfologis yang berbeda-beda, ditambah dengan kebutuhan untuk mempertahankan

detail *fine-grained* dan kewajaran anatomis, menghadirkan tantangan yang berbeda dari domain-domain yang telah dievaluasi sebelumnya. Domain ornitologi dengan karakteristik multi-spesies dan batasan morfologis ini belum dieksplorasi dalam konteks FastGAN, sehingga performa model pada domain biologis dengan kompleksitas tersebut masih menjadi pertanyaan terbuka.

Penelitian mengenai sintesis citra burung menggunakan *FastGAN* dapat memberikan pemahaman mengenai bagaimana model *few-shot GAN* berperforma pada domain dengan keragaman spesies tinggi dan kebutuhan presisi morfologis. Pemahaman ini penting karena memiliki potensi penerapan di berbagai bidang seperti edukasi, konservasi, serta augmentasi data untuk sistem klasifikasi otomatis. Oleh karena itu, penelitian ini bertujuan untuk mengimplementasikan dan mengevaluasi performa *FastGAN* dalam menghasilkan citra sintesis burung menggunakan dataset CUB-200-2011, serta menilai kualitas hasil sintesisnya secara kuantitatif dan kualitatif.

1.2 Landasan Teori

1.2.1 Kecerdasan Buatan (Artificial Intelligence)

Kecerdasan Buatan (bahasa inggris: Artificial Intelligence atau AI) merupakan cabang ilmu komputer yang berfokus pada perancangan dan pembangunan sistem yang dapat melakukan tugas-tugas yang normalnya memerlukan kecerdasan manusia (Russell & Norvig, 2022). Secara esensial, AI merupakan upaya untuk menciptakan mesin atau perangkat lunak yang dapat meniru kemampuan kognitif manusia seperti pembelajaran, penalaran, pemecahan masalah, persepsi, dan pemahaman bahasa (Russell & Norvig, 2022). AI modern mencakup berbagai pendekatan, mulai dari sistem berbasis aturan (*rule-based systems*) hingga pembelajaran mesin (*machine learning*) yang dapat belajar dari data tanpa diprogram secara eksplisit (Mitchell, 1997).

Dalam bidang pengolahan citra, machine learning berkembang menjadi deep learning, yaitu pendekatan yang memanfaatkan jaringan saraf tiruan berlapis (*deep neural networks*) untuk mengenali pola visual kompleks (Ian Goodfellow et al., 2016; LeCun et al., 2015). Model seperti Convolutional Neural Networks (CNN) telah banyak digunakan untuk klasifikasi dan deteksi objek, sedangkan autoencoder dimanfaatkan untuk rekonstruksi dan representasi fitur (LeCun et al., 2015).

Seiring perkembangan teknologi deep learning, muncul paradigma baru yaitu model generatif, di mana sistem tidak hanya mampu mengenali data, tetapi juga menciptakan data baru dengan pola yang menyerupai data asli. Salah satu model yang merepresentasikan terobosan besar dalam bidang ini adalah Generative Adversarial Networks (GAN), yang diperkenalkan oleh Goodfellow et al. (2014). Model ini menjadi dasar dari banyak arsitektur generatif modern, termasuk StyleGAN dan FastGAN, yang kemudian dikembangkan untuk meningkatkan stabilitas dan kualitas hasil citra.

1.2.2 Generative Adversarial Networks (GAN)

Generative Adversarial Networks (GAN) adalah kelas algoritma pembelajaran mesin generatif yang diperkenalkan oleh Goodfellow et al. (2014) sebagai pendekatan

unsupervised learning yang mampu menghasilkan data sintetik baru dengan distribusi yang menyerupai data asli. Berbeda dari model diskriminatif yang mempelajari pemetaan dari input ke label, model generatif mempelajari distribusi probabilitas data untuk menghasilkan sampel baru yang serupa dengan data pelatihan (Goodfellow et al., 2014). GAN merepresentasikan paradigma baru dalam pembelajaran generatif dengan menggunakan pendekatan *adversarial*, di mana dua model jaringan saraf dilatih secara simultan dalam kerangka permainan minimax (*minimax game*) (Goodfellow et al., 2014).

GAN terdiri dari dua komponen utama yang dilatih secara bersamaan dalam proses *adversarial*:

1. Generator (G): Jaringan saraf yang menerima vektor acak (noise) dengan distribusi prior sederhana (biasanya Gaussian atau uniform) sebagai input dan mengubahnya menjadi citra sintesis (Goodfellow et al., 2014). Proses transformasi ini melibatkan serangkaian operasi non-linear yang memetakan ruang laten berdimensi rendah ke ruang citra berdimensi tinggi (Radford et al., 2015). Tujuan Generator adalah menghasilkan citra yang tidak dapat dibedakan dari citra asli oleh Discriminator (Goodfellow et al., 2014).
2. Discriminator (D): Jaringan saraf yang menerima citra (baik asli maupun sintesis) sebagai input dan mengeluarkan probabilitas skalar antara 0 dan 1 yang menunjukkan apakah citra tersebut asli atau sintesis melalui klasifikasi biner (Goodfellow et al., 2014; Salimans et al., 2016). Discriminator pada dasarnya berfungsi sebagai kritik yang memberikan sinyal pembelajaran kepada Generator (Salimans et al., 2016).

Proses pelatihan GAN berlangsung secara *adversarial*, di mana Generator berusaha “mengecoh” Discriminator dengan menghasilkan citra yang semakin realistis, sementara Discriminator berusaha semakin baik dalam mendeteksi citra sintesis (Goodfellow et al., 2014). Dinamika kompetitif ini menciptakan feedback loop di mana peningkatan kemampuan satu jaringan mendorong peningkatan jaringan lainnya (Arjovsky et al., 2017). Keseimbangan dinamis antara kedua jaringan ini pada akhirnya menghasilkan Generator yang mampu menghasilkan citra berkualitas tinggi yang sulit dibedakan dari citra asli (Goodfellow et al., 2014).

Secara matematis, fungsi objektif GAN dapat dirumuskan sebagai permainan minimax antara Generator dan Discriminator berikut:

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

Di mana x Adalah citra asli dari distribusi data P_{data} , dan z Adalah vektor noise dari distribusi prior P_z (Goodfellow et al., 2014). Generator berusaha meminimalkan fungsi objektif ini sementara Discriminator berusaha memaksimalkannya, menciptakan dinamika kompetitif yang mendorong Generator untuk menghasilkan citra yang semakin realistis (Goodfellow et al., 2014). Dalam praktik, formulasi ini sering dimodifikasi untuk

meningkatkan stabilitas pelatihan, seperti yang diusulkan oleh Salimans dkk. (2016) melalui berbagai teknik pelatihan yang lebih baik.

Meskipun telah terbukti mampu menghasilkan citra yang realistis, GAN konvensional menghadapi beberapa tantangan dalam pelatihan:

1. Ketidakstabilan pelatihan: Keseimbangan antara Generator dan Discriminator sulit dicapai, sering menyebabkan salah satu jaringan mendominasi yang mengakibatkan kegagalan konvergensi atau osilasi yang tidak stabil (Arjovsky et al., 2017; Salimans et al., 2016). Fenomena ini dapat menyebabkan *vanishing gradients* pada Generator ketika Discriminator terlalu kuat, atau sebaliknya, menghasilkan citra berkualitas buruk ketika Generator mendominasi (Arjovsky et al., 2017).
2. *Mode collapse*: Generator hanya menghasilkan variasi terbatas dari citra yang serupa, kehilangan keragaman data asli karena fokus pada mode tertentu dari distribusi data yang mudah “mengecoh” Discriminator (Salimans et al., 2016). Dalam kasus ekstrem, Generator dapat menghasilkan satu atau beberapa sampel yang hampir identik terlepas dari input noise yang berbeda (Metz et al., 2016).
3. Kebutuhan data besar: Model GAN tradisional memerlukan dataset berukuran puluhan hingga ratusan ribu gambar untuk pelatihan yang efektif dan dapat mengalami overfitting pada dataset kecil, menghasilkan citra yang hanya mereplikasi sampel pelatihan (Arjovsky et al., 2017; Brock et al., 2019).
4. Kesulitan evaluasi: Tidak seperti model supervised learning yang memiliki metrik evaluasi jelas, menilai kualitas GAN memerlukan metrik khusus yang tidak selalu berkorelasi sempurna dengan penilaian manusia (Borji, 2018).

Beberapa perkembangan arsitektur Generative Adversarial Networks (GAN) dilakukan untuk mengatasi berbagai keterbatasan pada model konvensional seperti ketidakstabilan pelatihan, *mode collapse*, dan kebutuhan dataset besar.

- a. DCGAN (*Deep Convolutional GAN*) diperkenalkan oleh Radford et al. (2015) dengan mengganti arsitektur multilayer perceptron menjadi konvolusional penuh. Model ini menambahkan *batch normalization* dan fungsi aktivasi ReLU/LeakyReLU, sehingga pelatihan menjadi lebih stabil dan hasil citra lebih tajam. (Radford et al., 2015).
- b. Conditional GAN (cGAN) Varian ini diperkenalkan oleh Mirza dan Osindero (2014) dengan menambahkan informasi kondisional (seperti label kelas) pada Generator dan Discriminator. Pendekatan ini memungkinkan model menghasilkan citra dengan karakteristik tertentu sesuai kondisi yang diberikan. (Mirza & Osindero, 2014; Odena et al., 2016). yang memperkuat kontrol semantik terhadap hasil generasi.
- c. Wasserstein GAN (WGAN) diperkenalkan oleh Arjovsky et al. (2017) dengan mengganti fungsi kerugian konvensional menjadi jarak Wasserstein agar gradien lebih stabil dan mengurangi *mode collapse*. Varian lanjutannya, WGAN-GP (Arjovsky et al., 2017; Gulrajani et al., 2017), menambahkan *gradient penalty* untuk menjaga stabilitas selama pelatihan.

- d. Spectral Normalization GAN (SN-GAN) oleh Miyato et al. (2018) menormalkan bobot Discriminator menggunakan *spectral normalization* guna mengontrol *Lipschitz constant*, meningkatkan stabilitas tanpa mengurangi keragaman citra. (Miyato et al., 2018).
- e. Progressive GAN dikembangkan oleh Karras et al. (2018) dengan melatih model secara bertahap dari resolusi rendah ke tinggi. Strategi *progressive growing* ini memungkinkan sintesis citra beresolusi tinggi hingga 1024×1024 piksel dengan hasil yang lebih halus dan stabil. (Tero Karras et al., 2018).
- f. StyleGAN diperkenalkan oleh Karras et al. (2019) sebagai pengembangan dari Progressive GAN. Model ini memisahkan kontrol *style* dan *content* melalui mekanisme *style-based generation*, sehingga mampu mengatur atribut visual seperti warna dan tekstur secara hierarki (Karras et al., 2018).
- g. StyleGAN2, yang disempurnakan oleh Karras et al. (2020), memperbaiki kelemahan StyleGAN dengan menghapus *progressive growing*, menambahkan *path length regularization*, dan meningkatkan konsistensi spasial. Model ini menghasilkan citra dengan artefak minimal dan kualitas visual yang sangat realistis (Karras et al., 2019)

Kedua arsitektur terakhir, StyleGAN dan StyleGAN2, menjadi dasar penting bagi pengembangan model *few-shot* berikutnya seperti FastGAN, yang difokuskan pada efisiensi pelatihan dan kemampuan menghasilkan citra berkualitas tinggi dengan jumlah data terbatas.

1.2.3 FastGAN

FastGAN adalah arsitektur GAN yang dikembangkan oleh Liu dkk. (2021) secara khusus untuk mengatasi masalah sintesis citra dengan data terbatas (*few-shot learning*). Berbeda dengan GAN konvensional yang memerlukan puluhan ribu sampel, FastGAN dirancang untuk menghasilkan citra berkualitas tinggi hanya dengan ratusan hingga ribuan sampel pelatihan (Liu et al., 2021). Motivasi pengembangan FastGAN berangkat dari observasi bahwa banyak domain aplikasi memiliki keterbatasan data, baik karena biaya akuisisi yang tinggi, kelangkaan subjek, atau keterbatasan praktis lainnya (Liu et al., 2021; Noguchi & Harada, 2019).

FastGAN memperkenalkan dua inovasi arsitektural utama:

1. Skip-Layer Excitation (SLE)

SLE adalah modul yang memungkinkan aliran informasi langsung dari lapisan resolusi rendah ke lapisan resolusi tinggi dalam Generator (Liu et al., 2021). Konsep ini terinspirasi dari squeeze-and-excitation networks yang menunjukkan efektivitas recalibration fitur melalui mekanisme attention (Hu et al., 2017). Berbeda dari *skip connections* konvensional seperti pada U-Net yang hanya menggabungkan fitur melalui *concatenation*, SLE menggunakan mekanisme *excitation* yang adaptif untuk memodulasi fitur dari lapisan sebelumnya sebelum digabungkan ke lapisan yang lebih tinggi (Liu et al., 2021). Hubungan ini secara formal dapat didefinisikan dengan persamaan:

$$\mathbf{y} = \mathcal{F}(x_{low}, \{\mathbf{W}_i\}) \cdot \mathbf{x}_{high} \quad (2)$$

Di mana $\mathbf{y}$ Adalah feature-map keluaran, x_{low} dan x_{high} Adalah feature-map masukan dari lapisan resolusi rendah dan tinggi, $\mathcal{F}$ Adalah fungsi yang berisi operasi pada x_{low} , dan $\{\mathbf{W}_i\}$ Adalah bobot yang dipelajari selama pelatihan (Liu et al., 2021).

Secara teknis, SLE bekerja melalui tahapan sebagai berikut (Liu et al., 2021):

- 1) Ekstraksi fitur: Mengambil fitur dari lapisan resolusi rendah sebagai representasi global.
- 2) Transformasi adaptif: Menerapkan transformasi menggunakan *fully connected layers* untuk menghasilkan bobot *excitation* yang spesifik terhadap konten fitur.
- 3) Modulasi fitur: Menggunakan hasil transformasi untuk memodulasi (*excite*) fitur pada lapisan resolusi tinggi melalui operasi perkalian *element-wise*, sehingga informasi dari lapisan rendah dapat dapat memengaruhi pembelajaran di lapisan tinggi.

Mekanisme ini memfasilitasi pembelajaran fitur multi-skala dengan lebih efektif, sehingga Generator dapat menangkap baik struktur global maupun detail lokal secara bersamaan (Liu et al., 2021). Pada dataset dengan jumlah terbatas, SLE membantu model memanfaatkan informasi secara lebih efisien dengan memfasilitasi gradient flow yang lebih baik dan mengurangi risiko overfitting melalui regularisasi implisit (Liu et al., 2021).

2. Self-Supervised Discriminator

Discriminator dalam FastGAN dilengkapi dengan decoder yang merekonstruksi bagian dari citra input (Liu et al., 2021). Pendekatan self-supervised learning telah terbukti efektif dalam berbagai tugas computer vision untuk mempelajari representasi yang kaya tanpa label eksplisit (Doersch et al., 2015; Pathak et al., 2016). Secara spesifik, decoder dalam FastGAN berusaha merekonstruksi patch acak dari citra asli menggunakan fitur yang diekstrak oleh Discriminator (Liu et al., 2021). Tugas rekonstruksi ini berfungsi sebagai pembelajaran tambahan (*auxiliary task*) yang membantu Discriminator mempelajari representasi fitur yang lebih kaya dan informatif, meningkatkan stabilitas pelatihan dan mengurangi risiko *mode collapse* (Liu et al., 2021).

Integrasi self-supervised learning ke dalam Discriminator memberikan beberapa keuntungan (Liu et al., 2021). Pertama, pendekatan ini mendorong Discriminator untuk mempelajari representasi semantik yang lebih kaya dan informatif, bukan hanya berfokus pada perbedaan antara citra asli dan sintesis. Kedua, tugas rekonstruksi memberikan sinyal pembelajaran tambahan yang membantu stabilisasi pelatihan dengan menyediakan gradient yang lebih informatif. Ketiga, pembelajaran tambahan ini mengurangi risiko Discriminator mengalami overfitting pada dataset kecil, karena model dipaksa untuk mempelajari struktur umum dari citra, bukan hanya menghafal sampel pelatihan.

Tugas rekonstruksi ini dioptimalkan menggunakan reconstruction loss sederhana yang hanya dilatih pada data asli, seperti yang dirumuskan di bawah ini:

$$L_{recons} = \mathbb{E}_{f \sim D_{encode}(x), x \sim I_{real}} [||g(f) - \mathcal{T}(x)||] \quad (3)$$

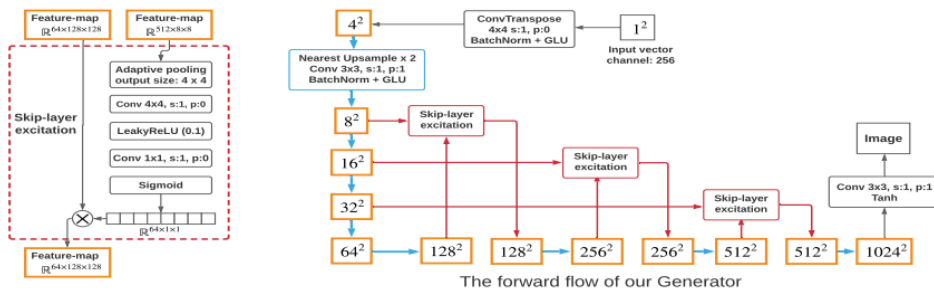
Di mana f Adalah *feature-map* internal dari Discriminator, G Adalah decoder yang mencoba merekonstruksi gambar dari fitur f , dan T Adalah fungsi transformasi (seperti cropping) pada gambar asli x (Liu et al., 2021).

Loss ini kemudian digabungkan dengan *adversarial loss* standar (*hinge loss*) untuk membentuk fungsi *loss* total dari Discriminator:

$$LD = -\mathbb{E}_{x \sim I_{real}} [\min(0, -1 + D(x))] - \mathbb{E}_{\hat{x} \sim G(z)} [\min(0, -1 - D(\hat{x}))] + L_{recons} \quad (4)$$

Di mana L_D Adalah *loss* total Discriminator yang berusaha diminimalkan. Bagian pertama dan kedua dari persamaan Adalah *hinge loss* untuk data asli (x) dan data sintesis ($\hat{x}$), sedangkan L_{recons} Adalah *loss* dari tugas rekonstruksi *self-supervised* (Liu et al., 2021).

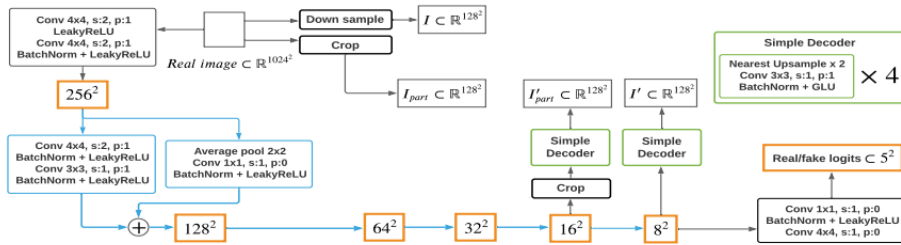
Selain kedua inovasi utama, FastGAN juga mengintegrasikan DiffAugment, teknik augmentasi data yang diterapkan pada citra sebelum masuk ke Discriminator untuk mencegah overfitting (Zhao et al., 2020). DiffAugment menerapkan transformasi stokastik seperti translation, cutout, dan color jittering yang memaksa Discriminator untuk belajar fitur yang invariant terhadap augmentasi, meningkatkan generalisasi pada dataset kecil (Zhao et al., 2020).



Gambar 1. Arsitektur FastGAN dengan Skip-Layer Excitation

Sumber: Liu dkk. (2021)

Keterangan Gambar: Diagram mengilustrasikan arsitektur FastGAN yang mengintegrasikan tiga komponen utama: Generator dengan koneksi Skip-Layer Excitation (SLE) antar lapisan untuk aliran informasi multi-skala, Discriminator yang dilengkapi decoder untuk tugas rekonstruksi self-supervised, dan modul DiffAugment yang diterapkan pada input Discriminator untuk augmentasi data stokastik.



Gambar 2. Arsitektur Discriminator FastGAN (Self-Supervised Decoder)

Sumber: Liu dkk. (2021)

Keterangan Gambar: Diagram menunjukkan arsitektur Discriminator FastGAN yang memanfaatkan rangkaian DownSample Blocks untuk ekstraksi fitur, diikuti classifier head untuk prediksi real/fake. Komponen self-supervised decoder digunakan untuk merekonstruksi patch citra sebagai tugas tambahan, yang berfungsi meningkatkan kualitas representasi fitur dan menjaga stabilitas pelatihan, terutama pada dataset kecil.

Kombinasi dari SLE, self-supervised discriminator, dan DiffAugment memungkinkan FastGAN untuk mencapai empat keunggulan utama (Liu et al., 2021; Zhao et al., 2020). Pertama, model dapat mencapai konvergensi lebih cepat dibandingkan GAN konvensional, dengan waktu pelatihan berkisar antara 5 hingga 20 jam pada single GPU (Liu et al., 2021). Kedua, FastGAN mampu menghasilkan citra berkualitas tinggi meskipun dilatih dengan dataset terbatas yang hanya berisi 60 hingga 1000 gambar (Liu et al., 2021). Ketiga, kombinasi ini mengurangi risiko overfitting pada dataset kecil melalui mekanisme regularisasi implisit yang berasal dari SLE dan self-supervised learning, serta regularisasi eksplisit dari augmentasi data DiffAugment (Liu et al., 2021; Zhao et al., 2020). Keempat, model dapat mempertahankan keragaman output meskipun jumlah sampel pelatihan terbatas, secara efektif mengatasi masalah mode collapse yang umum terjadi pada few-shot GAN (Liu et al., 2021).

Tabel 1. Perbandingan FastGAN dengan Metode Lain pada Few-Shot Setting

Metode	FFHQ(1k)	Art(1k)	Pokemon (800)	Skull (100)
DCGAN	72.32	98.45	125.34	201.56
StyleGAN2	45.21	65.43	87.92	165.43
FastGAN	24.45	45.08	57.19	130.05

Keterangan: Nilai yang ditampilkan adalah FID (semakin rendah semakin baik)

Sumber: Diadaptasi dari Liu dkk. (2021)

Evaluasi pada penelitian asli menunjukkan bahwa FastGAN mencapai performa kompetitif atau superior dibandingkan metode state-of-the-art pada berbagai domain dengan dataset 60-1000 gambar (Liu et al., 2021). Keberhasilan FastGAN pada domain dengan karakteristik berbeda—mulai dari wajah manusia yang memiliki struktur konsisten hingga kartun dengan variasi style tinggi—mendemonstrasikan fleksibilitas dan robustness pendekatan ini (Liu et al., 2021). Namun, hasil ini juga menunjukkan variasi performa yang signifikan antar domain, mengindikasikan bahwa kompleksitas dan

karakteristik visual domain memberikan pengaruh terhadap efektivitas model (Liu et al., 2021).

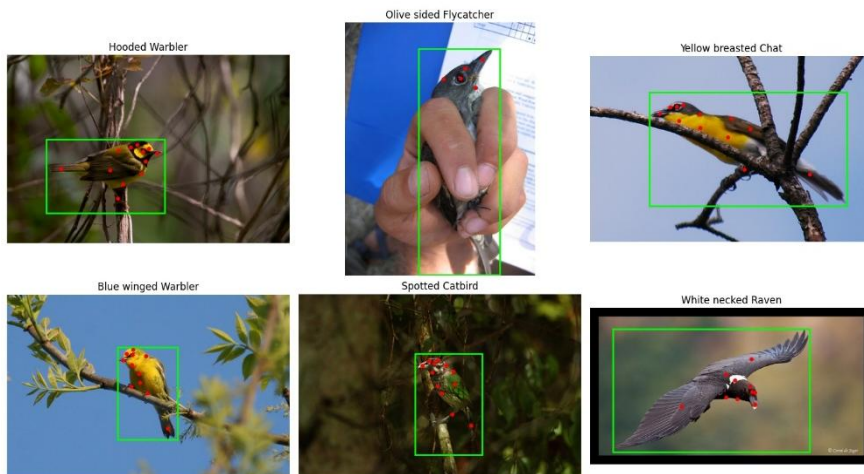
1.2.4 Ornitologi dan Dataset CUB-200-2011

Ornitologi merupakan cabang zoologi yang khusus mempelajari burung, mencakup aspek anatomi, fisiologi, perilaku, ekologi, evolusi, dan klasifikasi burung (Gill, 2007). Dalam konteks penglihatan komputer dan pembelajaran mesin, ornitologi menghadirkan tantangan yang kompleks karena variasi visual burung sangat tinggi dan sering kali bergantung pada detail morfologis yang halus. Beberapa tantangan utama dalam pengenalan citra burung antara lain:

1. Keragaman morfologis antar-spesies yang tinggi.
Dengan lebih dari 10.000 spesies burung di dunia, terdapat variasi ekstrem dalam ukuran, bentuk, warna, dan pola bulu antar spesies (Gill, 2007).
2. Ciri visual yang bersifat *fine-grained*.
Identifikasi spesies burung sering bergantung pada detail visual yang sangat halus seperti pola garis pada bulu, bentuk spesifik paruh, atau warna iris mata, yang membedakan spesies yang secara keseluruhan tampak serupa (Wah et al., 2011).
3. Variasi intra-spesies.
Dalam satu spesies dapat terdapat variasi signifikan berdasarkan usia (juvenil vs dewasa), jenis kelamin (dimorfisme seksual), musim (bulu kawin vs non-kawin), dan variasi geografis (subspesies) (Gill, 2007).
4. Kompleksitas pose dan konteks lingkungan.
Burung dapat difoto dalam berbagai pose (terbang, bertengger, berenang) dan konteks lingkungan (hutan, padang rumput, perairan) yang beragam (Wah et al., 2011).

Dataset CUB-200-2011 (*Caltech-UCSD Birds-200-2011*) merupakan salah satu dataset standar yang paling banyak digunakan dalam penelitian computer vision untuk domain ornitologi (Wah et al., 2011). Dataset ini terdiri dari 11.788 gambar yang mencakup 200 spesies burung dengan rata-rata 59 gambar per spesies (rentang: 41-60 gambar) (Wah et al., 2011). Setiap gambar dilengkapi dengan anotasi *bounding box* untuk lokalisasi burung, anotasi 15 bagian tubuh burung yang meliputi :

- Bounding box untuk lokalisasi objek burung.
- 15 bagian tubuh seperti paruh, mata, kepala, leher, punggung, dada, perut, sayap kiri dan kanan, kaki kiri dan kanan, serta ekor, dan
- 312 atribut visual biner yang mencakup aspek warna, pola, dan bentuk (Wah et al., 2011).



Gambar 3. Contoh Gambar dari Dataset CUB-200-2011 dengan Bounding Box

Sumber: Wah dkk. (2011)

Kelengkapan anotasi tersebut menjadikan CUB-200-2011 sangat sesuai untuk berbagai tugas dalam penglihatan komputer, seperti *fine-grained* (Branson et al., 2014), deteksi bagian tubuh (*part detection*) (N. Zhang et al., 2014), deskripsi gambar (*image captioning*) (Reed et al., 2016), dan pembelajaran metrik (*metric learning*) (Cui et al., 2017). Namun, meskipun dataset ini telah banyak digunakan dalam penelitian *recognition*, penggunaannya dalam sintesis citra menggunakan *Generative Adversarial Networks* (GAN) masih relatif terbatas, khususnya dalam konteks *few-shot learning*.

Hal ini menjadikan dataset CUB-200-2011 relevan sebagai landasan eksperimen dalam penelitian ini yang berfokus pada penerapan model FastGAN untuk sintesis citra burung berdasarkan jumlah data yang terbatas.

1.2.5 Metrik Evaluasi

Evaluasi model generatif memerlukan metrik kuantitatif yang dapat mengukur kualitas dan keragaman citra sintesis (Borji, 2018). Dua metrik yang paling umum digunakan untuk menilai kinerja model *Generative Adversarial Networks* (GAN) Adalah *Fréchet Inception Distance* (FID) dan *Inception Score* (IS).

1. *Fréchet Inception Distance* (FID)

FID merupakan metrik yang mengukur jarak antara distribusi citra asli dan distribusi citra sintesis dalam ruang fitur dari jaringan Inception-v3 yang telah dilatih pada dataset ImageNet (Heusel et al., 2017). FID dihitung dengan membandingkan statistik (mean dan covariance matrix) dari fitur yang diekstrak dari kedua distribusi, seperti ditunjukkan pada Persamaan (2):

$$FID = ||\mu_r - \mu_g||^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}) \quad (5)$$

Pada Persamaan (2), μ_r dan Σ_r adalah mean dan covariance dari distribusi citra asli, sedangkan μ_g dan Σ_g adalah mean dan covariance dari distribusi citra sintesis. Tr merupakan operasi trace pada matriks (Heusel et al., 2017).

FID yang lebih rendah mengindikasikan: - Citra sintesis lebih mirip dengan citra asli secara statistik - Distribusi citra sintesis lebih mendekati distribusi data asli - Kualitas dan keragaman citra sintesis yang lebih baik (Heusel et al., 2017).

Keunggulan FID: - Robust terhadap *noise* dan variasi kecil - Sensitif terhadap kualitas visual dan keragaman - Konsisten dengan penilaian manusia (Heusel et al., 2017)

-Lebih stabil dibandingkan Inception Score untuk mendeteksi *mode dropping* (Lucic et al., 2017).

2. Inception Score (IS)

IS mengukur kualitas dan keragaman citra sintesis berdasarkan prediksi model klasifikasi Inception-v3 (Salimans et al., 2016). IS didasarkan pada *Kullback–Leibler (KL) divergence* antara distribusi prediksi kondisional dan distribusi marginal dari label yang diprediksi, dengan rumus sebagai berikut:

$$IS = \exp (E_x[KL(p(y | x)p(y))]) \quad (6)$$

Di mana $p(y|x)$ adalah distribusi label yang diprediksi oleh Inception-v3 untuk citra x , dan $p(y)$ adalah distribusi label marginal dari seluruh citra sintesis (Salimans et al., 2016). *KL* menunjukkan Kullback-Leibler divergence (Salimans et al., 2016).

Nilai IS yang lebih tinggi menunjukkan: - Citra berkualitas tinggi (prediksi model *confident*, entropy rendah untuk $p(y|x)$) – Keragaman tinggi (distribusi marginal *uniform*, entropy tinggi untuk $p(y)$) (Salimans et al., 2016).

Namun, IS memiliki keterbatasan,di antaranya: - Tidak membandingkan dengan distribusi data asli – Bias terhadap dataset ImageNet (model Inception-v3 dilatih pada ImageNet) – Dapat memberikan skor tinggi untuk citra dengan artefak tertentu (Barratt & Sharma, 2018) – Sensitif terhadap jumlah sampel yang digunakan untuk evaluasi (Barratt & Sharma, 2018)

Dalam penelitian ini, kedua metrik digunakan secara komplementer: FID digunakan sebagai metrik utama untuk mengukur kedekatan distribusi citra hasil sintesis dengan data asli, sementara IS digunakan sebagai metrik pendukung untuk menilai kualitas dan keragaman citra yang dihasilkan (Liu et al., 2021; Lucic et al., 2017).

Tabel 2. Perbandingan Karakteristik Metrik Evaluasi GAN

Aspek	FID	IS
Menggunakan data asli	Ya	Tidak
Mengukur kualitas	Ya	Ya
Mengukur keragaman	Ya	Ya
Konsistensi dengan penilaian manusia	Tinggi	Sedang
Sensitivitas terhadap mode dropping	Tinggi	Rendah
Bias domain	Rendah	Tinggi (ImageNet)
Stabilitas evaluasi	Tinggi	Sedang

Sumber: Borji (2019)

1.3 Tujuan dan Manfaat

1.3.1 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Mengimplementasikan model FastGAN untuk menghasilkan citra sintesis dalam domain ornitologi menggunakan dataset CUB-200-2011.
2. Mengevaluasi kualitas citra sintesis yang dihasilkan oleh model FastGAN menggunakan metrik standar (FID dan IS) untuk menilai realisme dan keragaman gambar burung yang dihasilkan.

1.3.2 Manfaat Penelitian

Manfaat dari penelitian ini adalah:

1. Menjadi referensi bagi penelitian selanjutnya dalam pengembangan model sintesis citra untuk domain ornitologi dan domain biologis lainnya, dengan memberikan dokumentasi empiris tentang performa FastGAN pada dataset CUB-200-2011.
2. Mendukung penelitian di bidang konservasi burung dan ekologi dengan menyediakan citra sintesis yang dapat dimanfaatkan untuk keperluan edukasi, visualisasi, serta sebagai augmentasi data untuk pelatihan sistem klasifikasi burung berbasis kecerdasan buatan.
3. Memberikan wawasan tentang tantangan dan keterbatasan sintesis citra pada domain dengan keragaman spesies tinggi dan kebutuhan detail morfologis yang dapat menjadi panduan untuk penelitian sejenis.

BAB II

METODE PENELITIAN

2.1 Tempat dan Waktu

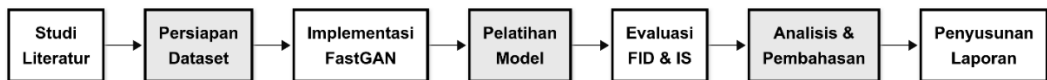
Penelitian dilakukan sejak bulan Mei 2025. Penelitian bertempat di Laboratorium *Cloud Computing and Information System (CCIS)*, Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin, Makassar, Sulawesi Selatan. Waktu pelaksanaan penelitian adalah selama 7 bulan dari bulan Mei hingga November tahun 2025.

2.2 Instrumen Penelitian

Berikut instrumen atau alat yang digunakan dalam penelitian ini:

1. Perangkat lunak:
 - a. Visual Studio Code
 - b. Microsoft Word
 - c. Microsoft Excel
 - d. Microsoft PowerPoint
 - e. Windows 11 Pro 64-bit
 - f. Zotero
2. Perangkat keras:
 - a. Intel(R) Core(TM) i9-14900K (32 CPUs), ~3.2GHz,
 - b. NVIDIA GeForce RTX 4090, RAM 64 GB
3. Bahasa pemrograman:
 - a. Python 3.8+
4. Model:
 - a. FastGAN
5. *Library*:
 - a. Pytorch (2.0.0): Kerangka kerja utama untuk membangun dan melatih model FastGAN.
 - b. Torchvision (0.15.1): Digunakan untuk transformasi data gambar dan model-model terkait visi komputer
 - c. Pandas (1.5.3): Digunakan untuk manipulasi dan analisis data.
 - d. numpy (1.23.5): Pustaka fundamental untuk komputasi numerik dengan Python.
 - e. opencv-python (4.5.4.60): Pustaka untuk tugas-tugas pemrosesan citra.
 - f. scikit-image (0.20.0): Pustaka lain untuk algoritma pemrosesan citra.
 - g. tqdm (4.64.1): Digunakan untuk menampilkan *progress bar* selama proses pelatihan.
 - h. scipy (1.10.1): Pustaka untuk komputasi ilmiah dan teknis.
 - i. Imdb (1.4.1): Digunakan untuk manajemen *database* memori yang efisien untuk memuat dataset.

2.3 Tahapan Penelitian



Gambar 4. Diagram Alur Tahapan Penelitian

Tahapan penelitian ini diawali dengan studi literatur, yaitu kegiatan pengumpulan referensi dan teori yang relevan dengan *Generative Adversarial Networks (GAN)*, *FastGAN*, serta metrik evaluasi citra yang digunakan dalam penelitian ini. Selanjutnya dilakukan pengumpulan dan persiapan data, yaitu proses menyiapkan dataset yang akan digunakan pada tahap pelatihan model agar sesuai dengan kebutuhan penelitian. Tahap berikutnya adalah implementasi dan pelatihan model *FastGAN*, yang dilakukan dengan menerapkan arsitektur model yang telah tersedia dan melatihnya untuk menghasilkan citra sintesis dengan kualitas yang baik. Setelah itu dilakukan evaluasi model menggunakan metrik yang sesuai untuk menilai performa dan kualitas citra yang dihasilkan oleh sistem. Tahap terakhir adalah analisis dan penyusunan laporan, yaitu menafsirkan hasil evaluasi serta menyusun keseluruhan proses penelitian ke dalam bentuk laporan akhir yang sistematis dan dapat dipertanggungjawabkan secara ilmiah.

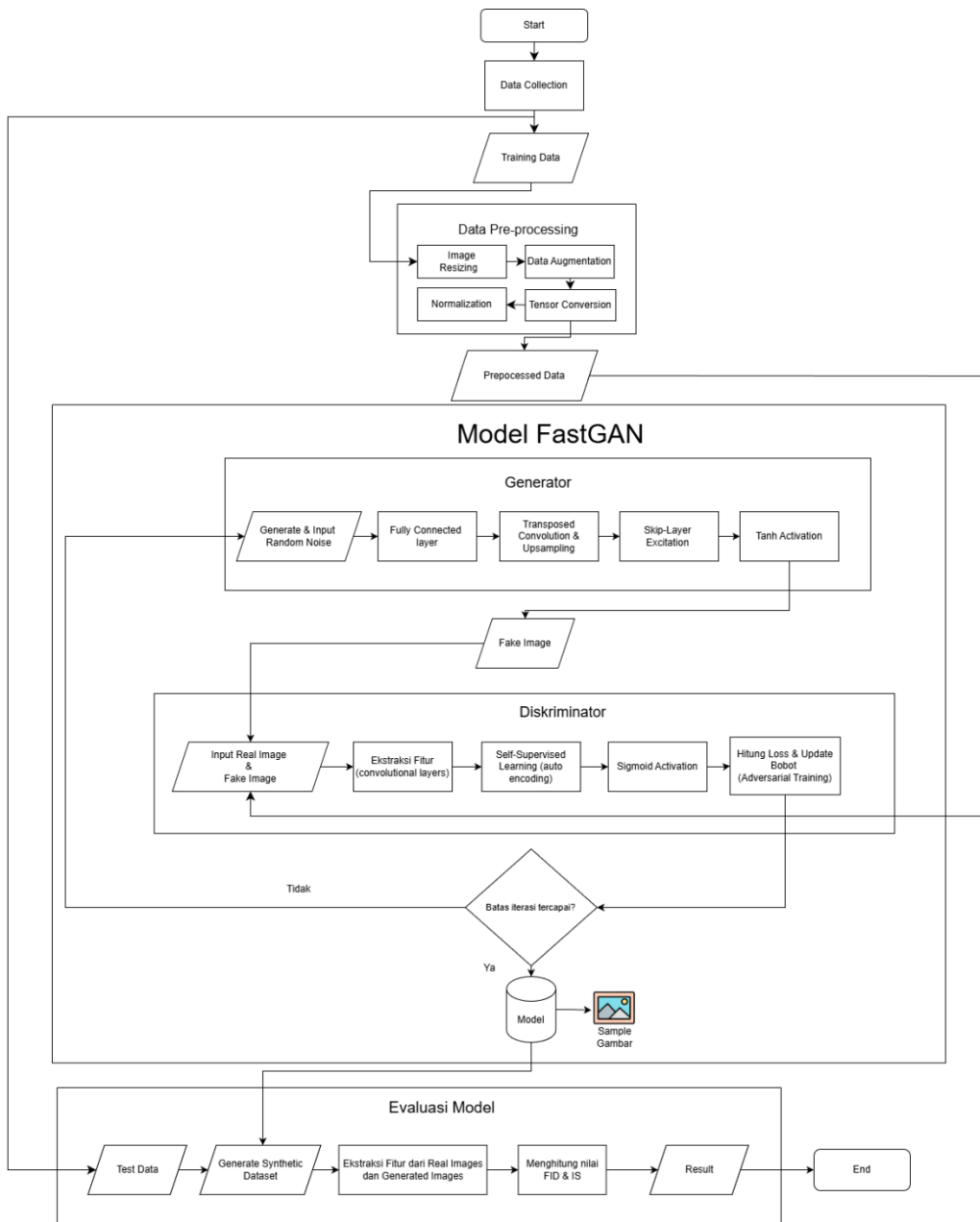
2.4 Teknik Pengambilan Data

2.4.1 Data Sekunder

Data pada penelitian ini diperoleh melalui metode pengambilan data sekunder, yaitu dengan memanfaatkan dataset yang telah tersedia secara publik. Dataset yang digunakan adalah Caltech-UCSD Birds-200-2011 (CUB-200-2011) yang dikembangkan oleh (Wah et al., 2011).

Metode ini dipilih karena dataset tersebut telah terverifikasi secara ilmiah dan banyak digunakan sebagai acuan dalam penelitian di bidang *computer vision* serta *image generation*, sehingga sesuai untuk dijadikan data dasar dalam pelatihan model *FastGAN*.

2.5 Perancangan dan Implementasi Sistem



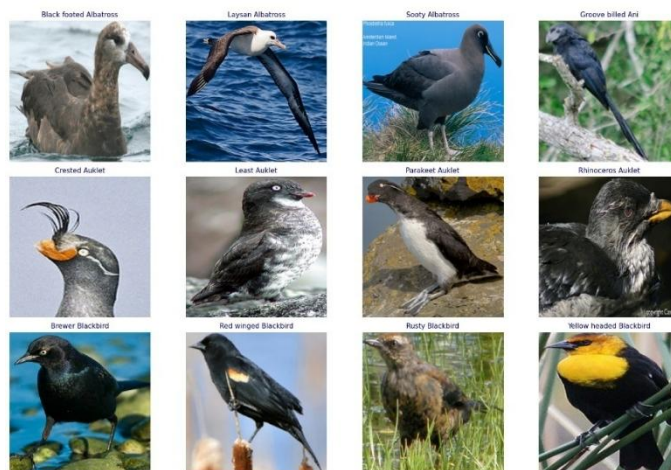
Gambar 5. Diagram Alur Perancangan dan Implementasi Sistem FastGAN

2.5.1 Data Collection

Tahap data collection merupakan proses awal dalam penelitian ini yang meliputi kegiatan pengunduhan, verifikasi, dan penyiapan dataset yang akan digunakan untuk pelatihan dan evaluasi model *FastGAN*. Data yang digunakan berasal dari dataset publik Caltech-UCSD Birds-200-2011 (CUB-200-2011) yang dikembangkan oleh Wah et al. (2011).

Dataset ini berisi 11.788 citra burung yang terdiri atas 200 kategori spesies dan telah disesuaikan dengan kebutuhan penelitian.

Dataset CUB-200-2011 diunduh dari repository resmi Caltech Vision Lab, kemudian dilakukan proses verifikasi integritas untuk memastikan seluruh citra berhasil diunduh secara lengkap dengan format berkas yang konsisten. Seluruh gambar yang diperoleh memiliki variasi latar belakang, pose, dan pencahayaan yang berbeda-beda, sehingga sesuai untuk digunakan sebagai data pelatihan model generatif. Contoh citra dari dataset CUB-200-2011 ditunjukkan pada Gambar 6.



Gambar 6. Contoh Gambar dari Dataset CUB-200-2011

Sumber: Wah et.al (2011)

Dalam penelitian ini, data dibagi menjadi dua bagian, yaitu:

- a. Data Pelatihan (Training Data), digunakan untuk melatih model FastGAN agar dapat mempelajari distribusi asli.
- b. Data Pengujian (Testing Data), digunakan untuk mengevaluasi performa model menggunakan metrik *Fréchet Inception Distance (FID)* dan *Inception Score (IS)*.

Proporsi pembagian data dilakukan dengan rasio 90% untuk data pelatihan dan 10% untuk data pengujian. Pembagian ini dilakukan secara acak agar distribusi data tetap seimbang pada setiap kategori.

2.5.2 Data Pre-processing

Data pre-processing merupakan suatu proses mengubah data mentah menjadi data dengan format yang siap digunakan untuk pelatihan model. Tahap ini dilakukan agar data yang digunakan memiliki format, ukuran, dan skala nilai yang sesuai dengan kebutuhan arsitektur FastGAN. Selain itu, pre-processing juga bertujuan untuk menghilangkan ketidakseragaman format dan ukuran data citra yang dapat mengganggu proses pelatihan model.

Pada penelitian ini, tahap data pre-processing dilakukan melalui beberapa langkah sebagai berikut:

- a. Perubahan Ukuran Citra (Resizing)
Seluruh citra pada dataset diseragamkan menjadi ukuran 1024×1024 piksel agar sesuai dengan resolusi yang digunakan dalam pelatihan model *FastGAN*. Proses perubahan ukuran dilakukan menggunakan metode *bilinear interpolation* untuk menjaga proporsi dan menghindari distorsi pada citra.
- b. Augmentasi Data (Data Augmentation)
Tahap ini bertujuan untuk meningkatkan variasi data tanpa menambah jumlah citra baru. Proses augmentasi dilakukan dengan menerapkan transformasi sederhana seperti *pembalikan horizontal (horizontal flip) secara acak, serta augmentasi berbasis DiffAugment dengan kebijakan color dan translation*. Augmentasi *horizontal flip* memberikan variasi pada arah tampilan objek burung, sedangkan *DiffAugment color* memberikan variasi terhadap warna, kecerahan, dan saturasi citra. Sementara itu, *DiffAugment translation* menggeser posisi objek secara acak di dalam citra.
Kombinasi dari ketiga augmentasi tersebut membantu model *FastGAN* mempelajari variasi orientasi, pencahayaan, dan posisi objek dengan lebih baik tanpa mengubah struktur utama citra.
- c. Normalisasi Nilai Piksel (Normalization)
Setelah proses augmentasi dilakukan, seluruh nilai piksel citra dinormalisasi dari rentang [0, 255] menjadi [-1, 1] menggunakan rumus:

$$x_{norm} = \frac{x}{127.5} - 1 \quad (7)$$

Proses normalisasi ini disesuaikan dengan fungsi aktivasi *Tanh* yang digunakan pada lapisan output *generator* dalam arsitektur *FastGAN*. Dengan normalisasi ini, distribusi nilai piksel menjadi lebih seimbang sehingga proses pelatihan dapat berjalan lebih stabil dan konvergen lebih cepat.

- d. Konversi ke Tensor (Tensor Conversion)
Setelah dinormalisasi, citra dikonversi menjadi format tensor tiga kanal (RGB) agar dapat diproses oleh *PyTorch*. Konversi ini memungkinkan setiap citra direpresentasikan dalam bentuk numerik yang dapat dihitung oleh GPU selama proses pelatihan. Selanjutnya, data dalam format tensor ini akan dimuat menggunakan modul *DataLoader* yang membagi data ke dalam batch serta melakukan *shuffle* secara acak di setiap epoch untuk menghindari bias urutan data.

2.5.3 Pembuatan Subset Few-Shot

Untuk eksperimen *few-shot learning*, dilakukan pembuatan beberapa subset data dengan jumlah gambar yang bervariasi untuk mengevaluasi pengaruh ukuran dataset terhadap kualitas hasil sintesis model *FastGAN*. Subset tersebut dibentuk menggunakan metode stratified random sampling, sehingga setiap spesies tetap terwakili secara proporsional pada setiap percobaan.

Proses pembuatan subset dilakukan dengan cara mengambil sejumlah citra secara acak dari masing-masing dari 200 kategori spesies pada dataset CUB-200-2011. Empat konfigurasi subset yang digunakan dalam penelitian ini terdiri atas 1.000, 2.000, 5.000, dan 10.000 gambar, dengan rincian pengambilan sampel sebagaimana ditunjukkan pada Tabel 3.

Tabel 3. Konfigurasi Dataset untuk Eksperimen

Eksperimen	Jumlah Gambar	Jumlah Spesies	Gambar per Spesies	Tujuan
Eksperimen 1	1.000	200	5	Evaluasi pada kondisi <i>few-shot</i> (sangat terbatas)
Eksperimen 2	2.000	200	10	Evaluasi peningkatan jumlah sampel
Eksperimen 3	5.000	200	25	Evaluasi performa pada dataset menengah
Eksperimen 4	10.000	200	50	Evaluasi performa mendekati <i>full dataset</i>

Pemilihan konfigurasi subset tersebut mengikuti rancangan eksperimen pada penelitian *FastGAN* oleh Liu dkk. (2021), yang menggunakan jumlah data 1.000, 2.000, 5.000, dan 10.000 gambar untuk menilai sensitivitas model terhadap ukuran dataset. Dengan mengadopsi konfigurasi yang sama, penelitian ini memastikan kesetaraan kondisi eksperimen dan memungkinkan perbandingan hasil yang valid terhadap *baseline* penelitian asli.

Seluruh subset disusun dalam struktur direktori yang sama, dengan 200 subfolder yang masing-masing mewakili satu spesies burung. Subset *few-shot* (1.000–10.000 gambar) ditempatkan dalam folder terpisah untuk menjaga keteraturan dan memudahkan proses pelatihan terpisah pada setiap eksperimen. Daftar nama berkas yang termasuk dalam setiap subset juga disimpan dalam format teks untuk memastikan reproduktibilitas eksperimen.

2.5.4 Implementasi Model FastGAN

Tahap implementasi meliputi *setup* lingkungan pengembangan dan implementasi arsitektur FastGAN sesuai dengan spesifikasi paper asli (Liu et al., 2021). Lingkungan pengembangan disiapkan dengan instalasi Python 3.10.12, CUDA 11.8, dan seluruh Pustaka *dependencies* dalam *virtual environment* untuk memastikan konsistensi versi dan reproduktibilitas.

Arsitektur Generator: Generator pada FastGAN menggunakan arsitektur berbasis konvolusi dengan mekanisme Skip-Layer Excitation (SLE) untuk memfasilitasi aliran informasi multi-skala. Model menerima vektor laten z berdimensi 256 dari distribusi Gaussian $N(0,1)$, kemudian ditransformasikan melalui sebuah fully connected layer menjadi feature map $4 \times 4 \times 1024$.

Selanjutnya, 8 UpSample Blocks meningkatkan resolusi citra secara bertahap hingga mencapai ukuran $1024 \times 1024 \times 3$. SLE memungkinkan feature map dari layer resolusi rendah memodulasi feature map resolusi tinggi melalui mekanisme *adaptive excitation*, sehingga fitur global tetap terhubung dengan detail lokal. Lapisan output menggunakan konvolusi 3×3 dengan aktivasi Tanh untuk menghasilkan citra RGB dengan rentang piksel $[-1, 1]$.

Arsitektur Discriminator pada FastGAN menggunakan arsitektur encoder-decoder dengan komponen *self-supervised* untuk memperkaya representasi fitur. Bagian encoder menurunkan resolusi citra dari $1024 \times 1024 \times 3$ menjadi $4 \times 4 \times 1024$ melalui blok konvolusi berstride 2 dengan spectral normalization dan Leaky ReLU. Hasil encoding kemudian diproses oleh *classifier head* melalui global average pooling dan fully connected layer untuk menghasilkan skor real atau fake. Selain itu, *self-supervised decoder* merekonstruksi *random patch* 64×64 dari citra masukan menggunakan L1 loss, yang berfungsi memperkuat pemahaman Discriminator terhadap struktur visual dan mengurangi risiko overfitting pada dataset kecil.

DiffAugment diterapkan pada citra masukan *Discriminator* untuk mencegah terjadinya *overfitting* dan menjaga stabilitas proses pelatihan model. Teknik ini diusulkan oleh Zhao dkk. (2020) sebagai metode *differentiable augmentation* yang memungkinkan augmentasi diterapkan langsung dalam proses pelatihan tanpa mengganggu aliran gradien. Pada penelitian ini, kebijakan augmentasi (*policy*) yang digunakan terdiri atas dua jenis transformasi, yaitu *color* dan *translation*. Augmentasi *color* memberikan variasi pada kecerahan, kontras, dan saturasi citra secara kontinu dan diferensial, sehingga model menjadi lebih adaptif terhadap perubahan pencahayaan dan warna pada data pelatihan. Sementara itu, augmentasi *translation* menggeser posisi objek secara acak hingga sekitar $\pm 12,5\%$ dari ukuran citra untuk mensimulasikan perbedaan posisi dan komposisi objek dalam bingkai gambar. Kedua augmentasi tersebut diterapkan secara langsung pada setiap batch data pelatihan untuk meningkatkan kemampuan generalisasi model tanpa mengubah distribusi asli dari dataset.

Konfigurasi hyperparameter pelatihan dalam penelitian ini sebagian besar mengadopsi spesifikasi paper asli FastGAN (Liu et al., 2021) sebagai *baseline* guna memastikan reproduktibilitas hasil. Parameter standar seperti *learning rate*, *optimizer*, dan ukuran *batch* dipertahankan sesuai nilai *default*. Namun, penelitian ini menerapkan penambahan khusus berupa mekanisme *Differentiable Augmentation* (DiffAugment) dengan kebijakan transformasi warna dan translasi. Langkah ini dilakukan untuk memitigasi risiko *overfitting* yang kerap terjadi akibat keterbatasan jumlah data. Selain itu, mekanisme penyimpanan model (*checkpointing*) ditetapkan setiap 10.000 iterasi. Interval ini dipilih sebagai titik keseimbangan yang optimal: cukup sering untuk memantau perkembangan kualitas citra dan memilih model terbaik (*model selection*) secara rinci, namun tidak terlalu sering hingga membebani efisiensi penyimpanan memori (*storage efficiency*).

Batas iterasi ditetapkan sebesar 100.000 iterasi mengikuti konfigurasi *ablation study* pada penelitian (Liu et al., 2021), di mana grafik pada Appendix A Figure 8 menunjukkan

bahwa nilai FID cenderung konvergen setelah 60.000-80.000 iterasi pada berbagai domain. Rincian lengkap hyperparameter disajikan pada Tabel 4.

Tabel 4. Konfigurasi Hyperparameter Pelatihan

Parameter	Nilai	Keterangan
Learning rate	0.0002	Kecepatan pembaruan bobot model
Optimizer	Adam ($\beta_1 = 0.5$, $\beta_2 = 0.999$)	Optimizer yang stabil untuk pelatihan GAN
Batch size	8	Ukuran batch disesuaikan dengan kapasitas GPU
Latent dimension (z)	256	Dimensi <i>input noise</i> untuk <i>Generator</i>
λ (reconstruction)	10.0	Bobot <i>reconstruction loss</i> Discriminator
DiffAugment Policy	Color,translation	Augmentasi untuk menjaga stabilitas pelatihan
Update ratio D:G	1:1	Frekuensi update per iterasi
Checkpoint Interval	10.000	Simpan model setiap 10.000 iterasi

Implementasi mengikuti konfigurasi paper asli dengan inisialisasi bobot menggunakan distribusi normal (mean=0, std =0.002) untuk memastikan stabilitas pelatihan awal.

2.5.5 Evaluasi Kuantitatif

Evaluasi kuantitatif dilakukan pada setiap checkpoint model yang disimpan selama proses pelatihan, yaitu setiap 10.000 iterasi. Setelah proses pelatihan selesai, sistem menghasilkan dua keluaran utama berupa sample image untuk memantau perkembangan kualitas visual, serta model checkpoint yang berisi bobot Generator dan Discriminator. Kedua keluaran ini digunakan sebagai dasar untuk menilai performa model dan menentukan checkpoint terbaik berdasarkan metrik kuantitatif.

Pada penelitian ini, penilaian dilakukan menggunakan dua metrik yang paling umum dalam penelitian Generative Adversarial Networks (GAN), yaitu Fréchet Inception Distance (FID) dan Inception Score (IS), karena keduanya mampu memberikan penilaian objektif terhadap kualitas visual dan keragaman citra hasil sintesis.

Pada setiap *checkpoint* yang dievaluasi, proses evaluasi dimulai dengan memuat model *Generator* dari berkas *checkpoint* berformat *.pth*. Selanjutnya, dilakukan proses generasi terhadap 5.000 citra sintesis dengan melakukan *sampling* terhadap 5.000 *noise vector* yang diambil dari distribusi Gaussian $N(0,1)$. Setiap vektor masukan kemudian dilewatkan melalui *Generator* untuk menghasilkan citra dengan resolusi 1024×1024 piksel. Setelah itu, hasil keluaran model didenormalisasi dari rentang nilai [-1, 1] ke [0,

255] dan disimpan dalam format JPEG untuk keperluan evaluasi. Jumlah sampel sebesar 5.000 digunakan untuk menjaga kestabilan statistik pada proses perhitungan metrik, sesuai dengan protokol yang diterapkan pada penelitian asli FastGAN untuk dataset dengan jumlah gambar terbatas.

Perhitungan *Fréchet Inception Distance (FID)* digunakan untuk mengukur jarak statistik antara distribusi fitur citra asli dan citra hasil sintesis. Proses ini memanfaatkan arsitektur Inception-v3 *pre-trained* yang berfungsi semata-mata sebagai ekstraktor fitur eksternal, tanpa melalui proses pelatihan ulang. Model ini mengekstraksi fitur dari lapisan *pool3* berdimensi 2048, yang kemudian digunakan untuk menghitung nilai rata-rata (μ_r, μ_g) dan kovariansi (Σ_r, Σ_g) untuk distribusi citra asli dan citra sintesis. Nilai FID dihitung menggunakan persamaan:

$$FID = \|\mu_r - \mu_g\|^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}) \quad (8)$$

di mana Tr merupakan operasi *trace* pada matriks. Nilai FID yang semakin rendah menunjukkan bahwa distribusi citra hasil sintesis semakin mendekati distribusi citra asli, yang berarti kualitas dan keragaman citra yang dihasilkan semakin baik. Pada penelitian ini, perhitungan FID dilakukan menggunakan pustaka *pytorch-fid* yang telah tervalidasi secara luas dalam komunitas penelitian GAN.

Selain FID, metrik Inception Score (IS) digunakan untuk menilai kualitas dan keragaman citra hasil sintesis. Evaluasi ini memanfaatkan arsitektur Inception-v3 pra-latih (*pre-trained*) yang berfungsi semata-mata sebagai alat ukur eksternal. Mengadopsi metode kalkulasi dari (Salimans et al., 2016), nilai IS yang tinggi menunjukkan bahwa citra hasil sintesis memiliki struktur visual yang jelas (mudah dikenali) serta memiliki variasi yang tinggi antar-sampel. Dalam perhitungannya, seluruh citra hasil sintesis dilewatkan ke dalam model *Inception-v3* untuk memperoleh distribusi probabilitas kelas $p(y | x)$. Selanjutnya, dihitung distribusi marginal $p(y) = \frac{1}{N} \sum_i p(y | x_i)$, lalu diukur *Kullback–Leibler (KL) divergence* antara $p(y | x)$ dan $p(y)$ yang kemudian dirata-ratakan. Nilai IS diperoleh dengan persamaan:

$$IS = \exp(\mathbb{E}_x[KL(p(y | x) \parallel p(y))]) \quad (9)$$

Untuk meningkatkan reliabilitas evaluasi, 5.000 citra dibagi menjadi 10 subset, kemudian IS dihitung pada setiap subset. Hasil akhirnya dilaporkan dalam bentuk mean $\pm$ standard deviation, sesuai dengan standar yang diperkenalkan oleh (Salimans et al., 2016) Perhitungan dilakukan menggunakan pustaka *inception-score* yang kompatibel dengan PyTorch.

Seluruh hasil evaluasi dicatat secara sistematis, termasuk nomor iterasi checkpoint, nilai FID, nilai IS, serta waktu komputasi. Checkpoint dengan nilai FID terendah dipilih sebagai model terbaik dan digunakan pada tahap evaluasi kualitatif di Bab III. Visualisasi tren FID dan IS yang muncul selama proses pelatihan juga dianalisis untuk menilai stabilitas, konvergensi, dan efektivitas arsitektur FastGAN dalam mempelajari distribusi data

pelatihan. Hasil lengkap seluruh evaluasi kuantitatif disajikan dan dibahas secara rinci pada Bab III.

2.5.6 Evaluasi Kualitatif

Evaluasi kualitatif dilakukan untuk melengkapi evaluasi kuantitatif dengan melakukan inspeksi visual dan analisis morfologis terhadap citra sintesis yang dihasilkan. Evaluasi ini penting karena metrik kuantitatif seperti *Fréchet Inception Distance (FID)* dan *Inception Score (IS)* tidak selalu mampu merepresentasikan persepsi kualitas visual dan ketepatan detail anatomis yang dapat diamati secara manusiawi, terutama pada domain biologis yang menuntut reproduksi karakteristik morfologis secara akurat.

Inspeksi visual dilakukan melalui pembuatan grid perbandingan antara citra asli dan citra hasil sintesis untuk menilai Tingkat realisme secara menyeluruh. Proses ini mencakup penyusunan grid berukuran 4×8 yang menampilkan citra asli dan citra sintesis secara bergantian untuk memudahkan perbandingan langsung. Selain itu, dilakukan visualisasi perkembangan kualitas citra dengan menampilkan hasil sintesis dari beberapa *checkpoint pelatihan* (iterasi ke-10.000, 30.000, 50.000, 70.000, dan 100.000) guna memantau peningkatan realisme seiring proses pelatihan. Model juga dievaluasi terhadap keragaman hasil sintesis dengan menghasilkan beberapa batch citra untuk memastikan tidak terjadi *mode collapse* atau keseragaman berlebih pada keluaran model.

Analisis morfologis dilakukan untuk menilai sejauh mana model mampu mereproduksi karakteristik anatomi burung secara akurat. Aspek-aspek yang diamati meliputi proporsi tubuh (kepala, badan, sayap, dan ekor), bentuk dan posisi paruh, pola serta tekstur bulu, bentuk lipatan sayap, proporsi kaki, detail mata, serta kesesuaian dan konsistensi warna antara spesies. Setiap aspek dinilai secara subjektif dalam tiga kategori, yaitu baik (reproduksi akurat dan detail), cukup (reproduksi dikenali tetapi kurang detail), dan kurang (mengandung distorsi atau tidak akurat). Penilaian dilakukan terhadap sampel citra sintesis dari *checkpoint* terbaik berdasarkan hasil evaluasi FID terendah.

Selain itu, dilakukan identifikasi terhadap *failure modes* atau pola kegagalan yang muncul selama proses sintesis untuk memahami keterbatasan model. Kategori kegagalan yang diamati meliputi distorsi anatomis (bagian tubuh tidak proporsional atau posisi tidak alami), *texture artifacts* berupa blur, noise, atau inkonsistensi tekstur pada bulu, *color bleeding* atau penyebaran warna yang tidak wajar, *mode collapse* yang ditandai dengan rendahnya keragaman keluaran, serta *background fusion* di mana burung tampak menyatu dengan latar belakang. Setiap pola kegagalan yang teridentifikasi didokumentasikan beserta contoh citra sebagai bahan analisis lebih lanjut. Hasil lengkap dari evaluasi kualitatif, termasuk grid perbandingan visual dan dokumentasi *failure modes*.

2.6 Analisis dan Pembahasan

Tahap analisis dan pembahasan bertujuan untuk menginterpretasikan hasil evaluasi kuantitatif dan kualitatif yang diperoleh dari pelatihan model FastGAN, serta

membandingkannya dengan penelitian asli oleh Liu et al. (2021). Analisis ini dilakukan untuk menjawab tujuan penelitian dan memberikan pemahaman yang lebih mendalam mengenai performa *FastGAN* dalam mensintesis citra burung pada domain ornitologi, sekaligus mengidentifikasi faktor-faktor yang memengaruhi hasil pelatihan, baik dari sisi arsitektur model maupun karakteristik dataset.

Analisis dilakukan melalui dua pendekatan utama, yaitu perbandingan kuantitatif dan interpretasi kualitatif hasil sintesis citra. Pada perbandingan kuantitatif, nilai *Fréchet Inception Distance (FID)* dan *Inception Score (IS)* yang diperoleh dibandingkan dengan hasil penelitian asli *FastGAN* pada berbagai domain seperti *FFHQ*, *Art Paintings*, *Pokémon*, dan *Skull*. Tujuannya adalah untuk menilai posisi performa *FastGAN* pada domain ornitologi dan sejauh mana kompleksitas visual burung memengaruhi kualitas hasil sintesis. Selain itu, analisis juga dilakukan terhadap pengaruh ukuran dataset terhadap performa model dengan membandingkan hasil eksperimen *few-shot* (1.000, 2.000, dan 5.000 gambar) dengan eksperimen *full dataset* (10.000 gambar), guna melihat sensitivitas model terhadap jumlah data pelatihan.

Interpretasi hasil mencakup analisis tren konvergensi pelatihan, stabilitas nilai metrik sepanjang iterasi, serta efektivitas komponen arsitektur *FastGAN* seperti *Skip-Layer Excitation (SLE)*, *self-supervised discriminator*, dan *DiffAugment*. Aspek-aspek seperti keragaman antarspesies, kompleksitas morfologis, serta variasi pose dan latar belakang juga dipertimbangkan sebagai faktor yang memengaruhi performa model.

Seluruh hasil analisis disajikan dalam bentuk tabel, grafik, dan visualisasi citra sintesis untuk memperlihatkan perbandingan nilai metrik, tren pelatihan, serta kategori *failure modes*. Ringkasan mengenai kekuatan dan keterbatasan *FastGAN* pada domain ornitologi turut disertakan untuk memberikan pemahaman menyeluruh mengenai kemampuan model dalam merepresentasikan citra. Hasil lengkap dari analisis ini dijabarkan secara detail pada Bab III Hasil dan Pembahasan.