BAB I
PENDAHULUAN

1.1 Latar Belakang

AI art yang dihasilkan menggunakan teknologi *diffusion models* (atau *diffusion pobabilistic models*) semakin umum dijumpai dan menjadi pelopor dalam *text-to-image generation* (Zhang et al., 2023). Model populer seperti *DALL-E* dan *Stable Diffusion* banyak digunakan, memungkinkan pembuatan *AI art* melalui input *prompt* yang dipersonalisasi. Teknologi ini juga memungkinkan penyesuaian yang lebih spesifik menggunakan teknik *fine-tuning* seperti *Low-Rank Adaptation* (*LoRA*). Cara kerja *LoRA* adalah dengan membekukan bobot model yang telah dilatih dan menyuntikkan *rank decomposition matrices* yang dapat dilatih ke dalam lapisan arsitektur *transformers*, secara signifikan mengurangi jumlah parameter yang dapat dilatih (Hu et al., 2021).

Sayangnya, teknologi ini disalahgunakan untuk *style mimicry* (peniruan gaya), di mana model *fine-tuning* menggunakan sampel kecil karya seniman tertentu untuk memproduksi *AI art* dengan *style* mereka. Model-model ini banyak tersedia di situs *Civitai*, *Tensor.art*, *PromptHero*, dan *HuggingFace*. (Passananti et al., 2024). Kemunculan *AI* generatif telah menimbulkan perdebatan etika dan hukum, dan secara dramatis telah mengganggu industri seni visual. Jiang et al. (2023) mengkaji kerugian yang dialami seniman profesional akibat *AI generator* skala besar, termasuk kerusakan reputasi, kerugian ekonomi, plagiarisme, dan pelanggaran hak cipta.

Dari uraian di atas, tampak jelas bahaya yang mengintai para seniman saat mengunggah karya mereka di internet. Oleh karena itu, seniman membutuhkan alat untuk melindungi hak intelektual mereka. *Glaze* hadir sebagai salah satu solusi potensial. Sistem ini melindungi seniman dari *style mimicry* dengan menambahkan perturbasi minimal pada karya seni digital mereka untuk menyesatkan model *AI* agar menghasilkan seni yang berbeda dari seniman yang ditargetkan (Shan, Cryan, et al., 2023).

Efektivitas *Glaze* telah diuji oleh Shan, Cryan, et al. (2023). Dalam percobaan mereka, filter *Glaze* diuji terhadap *style* unik empat seniman kontemporer dan juga beberapa seniman historis seperti Van Gogh dan Monet. Dengan menggunakan metrik *Perceptual Style Recognition* (*PSR*) yaitu survei berdasarkan skala *Likert* dan juga mendefinisikan sebuah metrik baru, *CLIP-based genre shift,* yang didasari oleh *CLIP* (*Contrastive Language-Image Pre-Training*). Penelitian ini menunjukkan tingkat keberhasilan yang baik. *Glaze* terbukti berhasil membuat serangan *style mimicry* kurang efektif pada karya seni yang telah dilindungi, mencapai *PSR* dengan peringkat artis > 93.3% dan pergeseran genre berbasis *CLIP* > 96.0%. Perlu dicatat metrik *CLIP* hanya digunakan sebagai suplemen dan memiliki keterbatasan. Akurasi *CLIP* sangat buruk, terutama saat mengevaluasi serangan terhadap *Glaze*, karena tidak dapat menangkap degradasi kualitas citra yang (Shan, Wu, et al., 2023).

Namun, efektivitas *Glaze* tidak selalu konsisten di semua *style*. Penelitian lain oleh Zulhan et al. (2024) menguji efektivitas *Glaze* (v1.1.1) menggunakan 3 *style*

yaitu; abstrak, realis, dan kartun. Penelitian ini menggunakan metode *Image Quality Assessment* (I*QA*) berupa *Peak-to-signal Radio* (*PSNR*), *Structural Similarity Index Measure* (*SSIM*), *Learned Perceptual Image Patch Similarity* (*LPIPS*), *supervised learning* yang dilakukan dengan prediksi arsitektur *Inception-v3*, serta *human feedback* melalui survei dengan teknik *Mean Opinion Score* (*MOS*). Hasil penelitian ini menunjukkan bahwa *Glaze* hanya menunjukkan perturbasi pada gaya abstrak, dengan mengurangi prediksi *supervised network* sebesar 47.7% dan *MOS* sebesar 1.285 poin.

Diperlukan metrik kuantitatif yang universal dan objektif untuk mengevaluasi kualitas citra. *Fréchet Inception Distance* (*FID*) dan *Inception Score* (*IS*) adalah dua metrik umum yang digunakan. Menurut Zhang et al. (2023), *FID* mengukur jarak *Fréchet* (atau jarak *Wasserstein-2*) antara distribusi citra hasil generasi dan citra asli. Di sisi lain, Dash et al. (2017) menyoroti bahwa *IS* sering menjadi tolak ukur untuk mengevaluasi kinerja *Generative Adversarial Networks* (GAN) karena menyediakan ukuran ter-standarisasi.

Selain itu, *CNN* telah muncul sebagai alat yang ampuh untuk ekstraksi fitur di berbagai aplikasi. Di antara banyak arsitektur yang tersedia, model-model tertentu menonjol karena kinerja, keserbagunaan, dan kemampuannya dalam mengekstraksi fitur dari gambar.

Inception-v3 adalah arsitektur yang dirancang untuk meningkatkan kecepatan dan akurasi dalam klasifikasi citra. Model ini menggunakan kombinasi berbagai jenis lapisan konvolusi (1x1, 3x3, 5x5) untuk mengekstrak fitur pada berbagai skala, yang krusial untuk mengidentifikasi fitur gaya yang halus (*fine-grained*) dalam aplikasi klasifikasi karya seni (membedakan karya seni asli dan hasil generasi *AI*). Duan (2023) menyimpulkan bahwa *Inception-v3* dapat mengklasifikasikan karya seni dengan akurasi 95.26%, jauh lebih tinggi dibandingkan dengan persepsi manusia (87.09%), yang membantu mengeliminasi bias pribadi.

Deteksi karya seni yang dihasilkan oleh *AI* juga diuji menggunakan model *ConvNeXt*. Arsitektur *CNN* ini menunjukkan kemampuan diskriminasi yang jauh melampaui persepsi manusia. Silva et al. (2024), menggunakan *ConvNeXt* sebagai *base model*, mampu mengidentifikasi citra yang dihasilkan *AI* dengan akurasi sekitar 99%. Sementara itu, subjek manusia hanya mencapai akurasi sekitar 58% melalui survey *Artistic Turing Test*. Disparitas besar ini menandakan bahwa *human feedback* sebagai metrik evaluasi tidak lagi relevan untuk menilai kualitas teknis maupun keaslian citra yang dihasilkan *AI*.

Dalam klasifikasi *fine-grained* lainnya, *Inception-Resnet-v2* juga menunjukkan hasil akurasi yang tinggi, bahkan unggul dibanding model lainnya, misalnya dalam penelitian klasifikasi bentuk wajah (Masruroh et al., 2023; Tasyanita, 2023). *Inception-Resnet-v2* menggabungkan dua metode, yaitu *Inception-v3* untuk mengurangi kompleksitas *CNN* dan *ResNet* untuk mengatasi degradasi kinerja melalui penggunaan blok residu, sehingga meningkatkan performa model.

Setiap model memiliki kelebihannya masing-masing. Oleh karena itu, arsitektur *Inception-v3*, *Inception-Resnet-v2*, dan *ConvNeXt-Tiny* perlu diuji dan dibandingkan secara komprehensif dalam konteks deteksi citra hasil generasi *AI* (*AI*

Art). Tujuan komparasi ini sangat penting demi kemajuan penelitian untuk mencegah potensi penyalahgunaan *AI Art*. Li et al., (2024) dalam penelitiannya menunjukkan bahwa beberapa alat pendeteksi gambar hasil *AI* yang tersedia memberikan hasil yang kurang memuaskan dengan akurasi yang rendah. Selain itu, alat pendeteksi yang ada umumnya tidak disertai dengan publikasi ilmiah untuk dipelajari atau divalidasi.

Berdasarkan (1) inkonsistensi efektivitas *Glaze* pada berbagai *style* (Zulhan et al., 2024) dan (2) keterbatasan metrik evaluasi yang ada, seperti ketidakcocokan *CLIP* serta superioritas metrik berbasis *CNN* dibanding persepsi manusia, dan (3) rendahnya akurasi alat pendeteksi *AI Art* yang tersedia, muncul kebutuhan mendesak untuk studi lebih lanjut mengenai efektivitas *Glaze* dengan pendekatan penilaian yang lebih komprehensif. Sehingga, penelitian ini dilakukan untuk mencapai dua tujuan utama. Pertama, menguji keefektifan *Glaze* dalam melindungi karya seni digital dari *style mimicry*. Kedua, membuat sebuah alat pendeteksi *AI Art* berdasarkan komparasi model *deep learning* terpilih (*Inception-v3*, *Inception-Resnet-v2*, dan *ConvNeXt-Tiny*). Pengujian ini akan dilakukan secara kuantitatif dengan menggunakan metrik berbasis fitur *FID*, *IS*, serta menggunakan *supervised learning* dengan menghubungkan metrik pada arsitektur *CNN* terpilih sebagai komparasi hasil keefektifan *Glaze*.

1.2 Rumusan Masalah

1. Bagaimana *Glaze* memengaruhi *FID* dan *IS*?
2. Bagaimana kinerja komparatif *Inception-v3*, *Inception-Resnet-v2*, dan *ConvNeXt-Tiny* dalam mengklasifikasikan *AI Art* dari data yang diproteksi *Glaze*?
3. Bagaimana model diimplementasikan dalam sebuah *website*?

1.3 Batasan Masalah

1. Menggunakan *dataset* yang terbatas
2. Menggunakan *setting* filter *Glaze* yang rendah (*default*)

1.4 Tujuan Penelitian

1. Mengevaluasi dan menganalisis *Glaze* terhadap *FID* dan *IS*.
2. Menganalisis dan mengevaluasi kinerja *Inception-v3*, *Inception-Resnet-v2*, dan *ConvNeXt-Tiny* dalam mengklasifikasikan *AI Art* dari data yang diproteksi *Glaze*.
3. Mengimplementasikan model yang sudah dievaluasi ke dalam *website*.

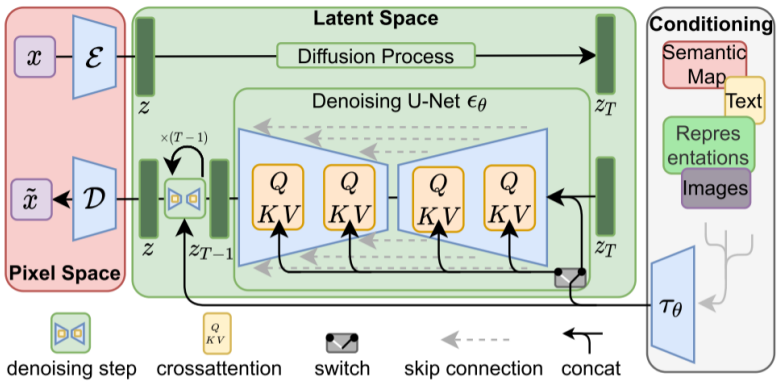
1.5 Manfaat Penelitian

1. Menyediakan dasar pengetahuan dan referensi empiris bagi penelitian-penelitian mendatang yang berfokus pada evaluasi efektivitas *Glaze*.
2. Menghadirkan alat bantu praktis berupa aplikasi web yang dapat digunakan publik untuk memverifikasi keaslian karya seni digital.

1.6 Landasan Teori

1.6.1 Stable Diffusion

Stable diffusion adalah implementasi dari arsitektur *Latent Diffusion Model* (LDM), yang diperkenalkan oleh Rombach et al. (2021). Model generatif *text-to-image* ini dirancang untuk menghasilkan citra berkualitas tinggi dari deskripsi tekstual (*prompt*) dengan efisiensi komputasi yang unggul. Kunci efisiensi *stable diffusion* adalah operasinya yang utama di ruang laten (*latent space*) yang terkompresi, daripada ruang piksel penuh, yang secara drastis mengurangi biaya pemrosesan dibandingkan model difusi sebelumnya. Pengembangan model lanjutan, seperti *Stable Diffusion XL* (SDXL) (Podell et al., 2023), menunjukkan peningkatan signifikan melalui peningkatan jumlah parameter dan penggunaan *text encoder* ganda, yang memungkinkan fidelitas, detail komposisi, dan kualitas visual keseluruhan yang jauh lebih tinggi. Arsitektur *LDM* dapat dilihat pada gambar berikut.



Gambar 1 Arsitektur *latent diffusion model* (Sumber: Rombach et al. (2021))

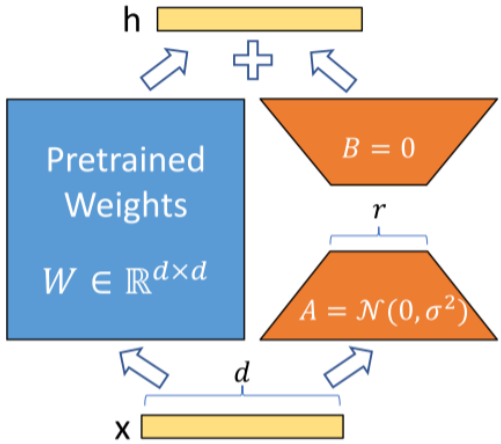
Menurut Rombach et al. (2021), *stable diffusion* bertumpu pada tiga komponen utama yang bekerja secara sinergis. Pertama, *Autoencoder Variational*

(VAE) yang berfungsi untuk kompresi citra piksel menjadi representasi laten yang ringkas melalui *encoder*, dan mengembalikannya ke citra piksel melalui *decoder*. Hal ini memastikan proses difusi terjadi pada dimensi yang lebih rendah, untuk menghemat sumber daya. Kedua, *U-Net Denoising Probabilistic* yang merupakan inti model, bertugas secara iteratif menghilangkan *noise* (kebisingan) dari representasi laten melalui serangkaian langkah *denoising* terbalik (*reverse diffusion*). Ketiga, mekanisme *Cross-Attention* yang krusial, yang memungkinkan *U-Net* dikondisikan oleh informasi semantik dari *prompt* teks yang sebelumnya telah di-*encode* oleh *Text Encoder*.

Proses konstruksi citra dimulai dengan representasi laten yang diisi *noise* acak. *Prompt* teks masukan diolah terlebih dahulu menjadi representasi vektor oleh *Text Encoder*. Model *U-Net* kemudian menjalankan proses *denoising* terbalik secara bertahap. Dalam setiap langkah, *U-Net* memprediksi dan mengurangi sejumlah kecil *noise* dari representasi laten, dengan panduan konstan dari vektor teks melalui mekanisme *cross-attention*. Proses ini berulang selama beberapa kali iterasi, secara progresif mengubah *noise* acak menjadi representasi laten yang terstruktur. Setelah iterasi *denoising* selesai, representasi laten diteruskan ke *Decoder VAE*, yang mengubahnya kembali menjadi citra piksel berkualitas tinggi yang menarik secara visual.

1.6.2 Low-Rank Adaptation (LoRA)

Low-Rank Adaptation (LoRA) adalah metode *fine-tuning* yang sangat efisien dan termasuk dalam kategori teknik *Parameter Efficient Fine-Tuning* (PEFT). *LoRA* diperkenalkan oleh Hu et al. (2021) sebagai solusi untuk mengatasi tantangan biaya komputasi dan memori yang sangat tinggi yang muncul ketika melakukan *fine-tuning* penuh pada model-model *deep learning* yang berukuran sangat besar (misalnya, dengan miliaran parameter). Meskipun *LoRA* awalnya dirancang untuk arsitektur *transformer* pada model bahasa, konsep dekomposisi *rank* rendah telah terbukti efektif dan diadopsi secara luas untuk melakukan adaptasi gaya (*style adaptation*) pada model generatif citra, seperti *stable diffusion*. Dalam konteks ini, *LoRA* memungkinkan pengguna untuk melakukan *fine-tuning* model generatif dengan *dataset* yang sangat kecil (misalnya, beberapa gambar dari gaya seni tertentu) untuk mencapai *style mimicry* tanpa perlu melatih ulang seluruh model yang berukuran gigabytes. Cara kerja *LoRA* dapat dilihat pada gambar berikut.



Gambar 2 Cara kerja *LoRA* (Sumber: Hu et al. (2021))

Konsep utama di balik *LoRA* adalah hipotesis bahwa perubahan bobot (ΔW) yang terjadi selama adaptasi model (saat *fine-tuning*) memiliki *rank* intrinsik yang rendah (Hu et al., 2021). Artinya, meskipun bobot model dasar (W) memiliki dimensi yang besar, informasi baru yang perlu dipelajari untuk tugas spesifik (*downstream tasks*) dapat direpresentasikan secara efektif oleh matriks yang jauh lebih kecil. *LoRA* mengimplementasikan hipotesis tersebut dengan beberapa cara. Pertama pembekuan bobot dasar, Bobot model *pre-trained* yang asli (W) dibekukan dan tidak diubah selama proses *fine-tuning*. Kemudian penyuntikan matriks dekomposisi *rank*, modifikasi pada bobot (W → W + ΔW) direpresentasikan secara tidak langsung melalui dua matriks dekomposisi *rank* rendah yang dapat dilatih, yaitu matriks A dan matriks B. Perubahan bobot (ΔW) didefinisikan sebagai perkalian dari matriks ini: ΔW = BA, di mana *rank r* (dimensi tengah matriks) jauh lebih kecil daripada dimensi asli matriks bobot (W). Dan terakhir efisiensi pelatihan, selama pelatihan, hanya matriks A dan B yang dioptimalkan (dilatih), sementara bobot dasar W tetap konstan. Hal ini secara drastis mengurangi jumlah parameter yang perlu dihitung gradiennya dan diperbarui.

Penerapan *LoRA* memberikan beberapa keuntungan kritis (Hu et al., 2021) seperti pengurangan jumlah parameter yang dapat dilatih hingga ribuan kali lipat dibandingkan *full fine-tuning*. Hal ini menurunkan kebutuhan memori dan *hardware*. Efisiensi Komputasi *throughput* yang lebih tinggi dan mengurangi hambatan *hardware*. Tanpa latensi inferensi tambahan, setelah pelatihan, matriks A dan B dapat digabungkan kembali ke bobot asli W (dikenal sebagai *merging*), sehingga tidak menimbulkan latensi atau keterlambatan tambahan saat inferensi. Kemudahan berbagi model dasar yang telah dilatih dapat digunakan bersama untuk berbagai tugas, dan *style* spesifik hanya perlu disimpan sebagai matriks *LoRA* yang sangat kecil.

1.6.3 Image Captioning: BLIP dan WD-Tagger

Ketika membandingkan efektivitas BLIP (*Bootstrapping Language-Image Pretraining*) dan WD-Tagger (*Web Data-Tagger*) untuk mmbuat model *LoRA* yang berfokus pada *style* seniman tertentu, penting untuk mempertimbangkan fungsionalitas dasar dan desain setiap model dalam kaitannya dengan konteks artistik.

BLIP dirancang untuk pemahaman kontekstual dan menghasilkan *caption* yang rumit berdasarkan input visual. Dibandingkan dengan *WD-Tagger*, yang utamanya berfokus pada penandaan citra dengan kategori yang telah ditentukan, *BLIP* unggul dalam menghasilkan teks deskriptif dan bernuansa yang dapat menangkap gaya dan konteks artistik sebuah citra. Dengan memanfaatkan pemahaman *multimodal*, kemampuan *BLIP* sangat menguntungkan saat mengadaptasi model untuk representasi artistik. Efisiensi dalam menghasilkan deskripsi yang relevan secara kontekstual ini memungkinkan seniman dan pengembang menyempurnakan model *LoRA* yang dapat dimodifikasi secara spesifik untuk mereplikasi gaya unik seorang seniman (Kaur et al., 2025).

Sebaliknya, *WD-Tagger* berfungsi terutama sebagai alat klasifikasi, berfokus pada *tagging* citra tanpa menghasilkan deskripsi tekstual. Kinerjanya bergantung pada kategorisasi cepat daripada pemahaman kontekstual yang lebih dalam, membuatnya kurang cocok untuk adaptasi bernuansa dalam domain artistik. Dengan demikian, saat membuat *LoRA* yang bertujuan mewujudkan gaya tertentu seorang seniman, keterbatasan WD-Tagger menjadi jelas, karena mungkin tidak secara efektif menangkap detail halus yang diperlukan untuk penyempurnaan ekspresi artistik.

1.6.4 Glaze

Glaze adalah alat inovatif yang dikembangkan oleh Shan, Cryan, et al. (2023) untuk melindungi seniman dari *style mimicry* di era model generatif, khususnya sistem *text-to-image*. *Glaze* menawarkan solusi terhadap kekhawatiran yang semakin meningkat dari para seniman yang gaya uniknya dapat dengan mudah di replikasi oleh algoritma *AI*, sehingga mengancam hak kekayaan intelektual dan ekspresi artistik asli mereka.

Arsitektur *Glaze* berfungsi dengan cara menyelubungi (*cloaking*) bagian tertentu dari karya seni seorang seniman sambil membiarkan bagian lain terbuka, secara efektif menciptakan perisai terhadap penyalinan tanpa izin. Penelitian menunjukkan bahwa *Glaze* mencapai tingkat perlindungan yang substansial, dengan tingkat keberhasilan lebih dari 85% bahkan dalam skenario di mana hanya sebagian kecil karya seniman yang diselubungi. Sebagai contoh, dalam kasus di mana seorang seniman berhasil menyelubungi hanya 25% dari karya *online* mereka, sistem tersebut mempertahankan penghalang pelindung yang masih diakui efektif oleh banyak pengguna (Shan, Cryan, et al., 2023). Tingkat perlindungan ini sangat penting, terutama jika mempertimbangkan proliferasi cepat konten buatan *AI* yang mungkin meniru gaya seniman secara tidak etis.

Implikasi Glaze meluas melampaui seniman individu. Kemunculan teknologi semacam ini memicu diskusi penting mengenai hak kekayaan intelektual secara lebih luas di era AI. Seniman semakin dihadapkan pada tantangan untuk menegaskan hak mereka di lingkungan di mana karya digital dapat dengan mudah direproduksi dan dimodifikasi tanpa atribusi yang layak. Kemampuan alat seperti *Glaze* untuk menyediakan perisai digital memperkuat pentingnya pengembangan kerangka hukum yang mendukung kreator dalam melindungi karya asli mereka terhadap peniruan dan plagiarisme yang berasal dari teknologi kecerdasan buatan yang berkembang pesat (Budiman & Hammar, 2024).

Oleh karena itu, *Glaze* tidak hanya berfungsi sebagai alat pelindung, tetapi juga menyoroti kebutuhan kritis akan percakapan berkelanjutan mengenai hak kekayaan intelektual dalam industri kreatif yang menghadapi kemajuan AI. Dengan menerapkan teknologi tersebut, seniman dapat lebih baik menavigasi kompleksitas lanskap artistik kontemporer, memastikan hasil kreatif mereka tetap unik di lingkungan digital yang semakin homogen (Pokrovskaya & Gronic, 2024; Kasmawati & Wiranata, 2023).

1.6.5 Inception Score (IS)

Inception Score (IS) Adalah metrik yang diakui secara luas dan digunakan untuk mengevaluasi kualitas citra yang dihasilkan oleh *Generative Adversarial Networks* (GANs). *IS* mengukur keragaman citra yang dihasilkan dan kualitasnya relatif terhadap model *Inception* yang telah dilatih sebelumnya menggunakan *dataset ImageNet*. Skor ini dihitung menggunakan distribusi probabilitas kondisional dari citra hasil generasi *AI*, memberikan wawasan mengenai seberapa baik *GAN* menangkap fitur-fitur dari citra asli. Salimans et al. (2016) memperkenalkan *IS* dan menunjukkan bahwa *IS* yang tinggi merefleksikan citra yang dihasilkan memiliki daya tarik visual dan termasuk dalam kategori yang dapat dibedakan, yang mengindikasikan representasi distribusi data yang baik.

Perhitungan *IS* melibatkan penggunaan model *Inception* yang telah dilatih sebelumnya untuk mengklasifikasikan citra hasil generasi *AI*. Secara spesifik, *IS* dihitung sebagai eksponen dari *Kullback-Leibler divergence* antara probabilitas kondisional dari citra yang dihasilkan ($p(y|x)$) dengan probabilitas kelas marginal di seluruh himpunan citra ($p(y)$). Secara matematis, *IS* didefinisikan sebagai:

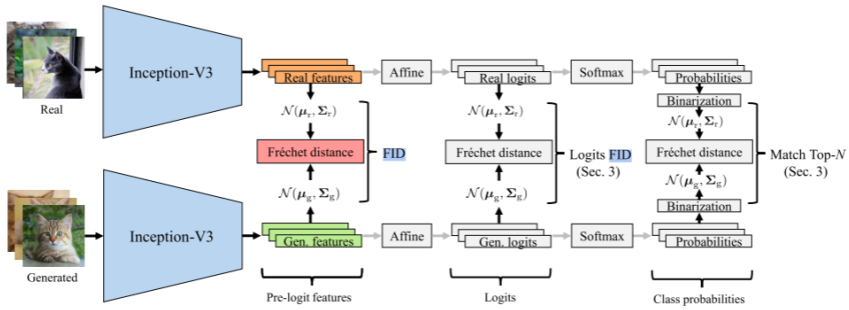
$$IS(\mathbb{P}_g) = e^{E_{x\sim\mathbb{P}_g}[KL(\mathcal{PM}(y|x)||\mathcal{PM}(y))]} \quad (1)$$

Proses ini memastikan bahwa skor *IS* yang tinggi mencerminkan dua karakteristik krusial yaitu citra individu memiliki kualitas visual dan kejelasan kategori yang baik (entropi $p(y|x)$ yang rendah), dan kumpulan citra secara kolektif menunjukkan variasi yang berarti antar kelas (distribusi $p(y)$ yang seragam atau beragam), menjadikannya tolak ukur yang berharga dalam skenario di mana realisme visual dan keragaman kolektif adalah hal yang paling krusial (Karras et al., 2017; Xu et al., 2018).

Berbagai penelitian (Lee & Seok, 2022;Gong et al., 2019) mencatat bahwa peningkatan kualitas citra hasil generasi berkorelasi dengan nilai *IS* yang lebih tinggi. Meskipun demikian, terdapat kritik terhadap metrik ini. Penelitian menunjukkan bahwa *IS* mungkin tidak secara universal menangkap kinerja berbagai *GAN* karena bias yang diperkenalkan oleh model klasifikasi yang digunakan dapat menghasilkan skor yang menyesatkan (Chong & Forsyth, 2019). Selain itu, Miyato et al. (2018) menyatakan bahwa *IS* yang tinggi menunjukkan diskriminasi citra yang baik, tetapi tidak selalu berkorelasi secara akurat dengan persepsi manusia. Oleh karena itu, LIASHCHYNSKYI & LIASHCHYNSKYI (2023) menyarankan pengguna *IS* bersamaan dengan kerangka evaluasi yang lain seperti *FID*.

1.6.6 Fréchet Inception Distance (FID)

Fréchet Inception Distance (FID) adalah metrik evaluasi objektif yang digunakan secara luas untuk menilai kualitas citra yang dihasilkan oleh model generatif, terutama *Generative Adversarial Networks* (GANs). Diperkenalkan oleh Heusel et al. (2017), *FID* mengukur jarak antara distribusi fitur dari citra asli dan citra hasil generasi *AI*, sehingga memberikan wawasan tentang seberapa mirip contoh yang dihasilkan menyerupai data asli (Borji, 2019; Kynkäänniemi et al., 2022). Metrik ini didefinisikan sebagai jarak *Fréchet* antara dua *multivariate gaussian distribution* yang dipasangkan pada representasi fitur citra yang diturunkan dari model *Inception-v3* yang telah dilatih sebelumnya. Cara kerja *FID* dapat dilihat pada gambar berikut.



Gambar 3 Cara Kerja *FID* (Sumber: Kynkäänniemi et al. (2022))

Perhitungan *FID* dimulai dengan melewatkan citra asli dan citra hasil generasi *AI* yang dihasilkan melalui jaringan *Inception-v3*, yang biasanya dihentikan pada salah satu lapisan awalnya untuk mengekstraksi representasi fitur yang bermakna. Fitur-fitur tersebut kemudian dimodelkan sebagai distribusi *Gaussian*, dengan parameter *mean*/rata-rata (μ) dan *covariance* (Σ), dihitung untuk dataset asli dan *dataset* hasil generasi *AI*. *Fréchet Distance*, yang mengukur kesamaan antara kedua distribusi ini, dihitung sebagai berikut:

$$FID = ||\mu_r - \mu_g||_2^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{\frac{1}{2}} \quad (2)$$

Di mana (μ_r) dan (μ_g) adalah rata-rata fitur citra asli dan citra hasil generasi *AI* secara berturut, dan (Σ_r) dan (Σ_g) adalah *covariance* (Andrade et al., 2020; Kynkäänniemi et al., 2022).

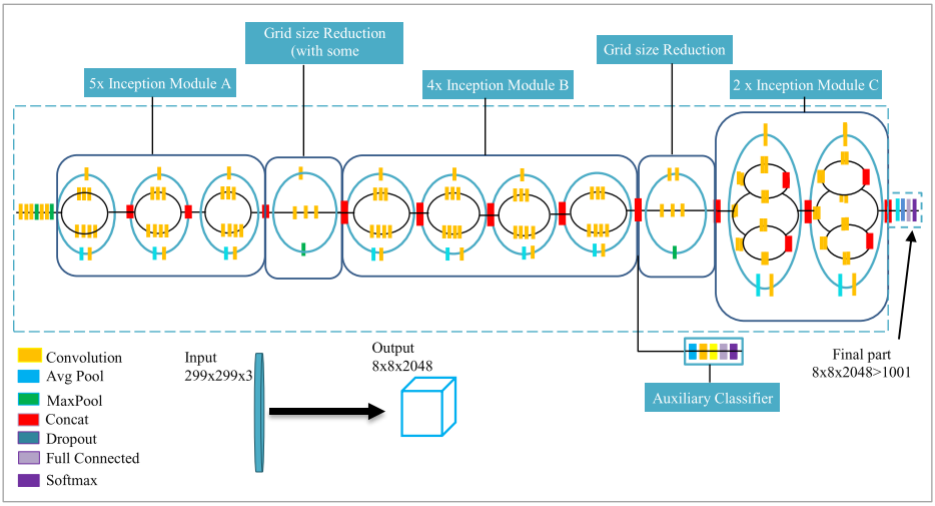
Jika dibandingkan dengan *IS, FID* memiliki keunggulan pada sensitivitasnya terhadap kualitas dan keragaman citra yang dihasilkan (Alam & Kehtarnavaz, 2024). Sebagi ganti evaluasi keyakinan model (seperti *IS*), *FID* berfokus pada keseluruhan distribusi dan kesamaan fitur citra sintesis terhadap citra asli. Hal ini menjadikan *FID* lebih andal untuk mendeteksi *moda collapse*, di mana *GAN* menghasilkan variasi keluaran yang terbatas meskipun skor *IS*-nya baik. Anschuetz & Zanoci (2019) menjelaskan bahwa nilai *FID* yang lebih rendah mengindikasikan kemiripan antara distribusi citra hasil generasi dan asli, menandakan kualitas dan keragaman citra yang lebih tinggi. Selain itu, *FID* telah banyak diterapkan untuk mengevaluasi arsitektur dan peningkatan *GAN* dalam generasi gambar (Ravuri et al., 2018; Azadi et al., 2018). Meskipun tangguh, *FID* masih dapat dipengaruhi oleh faktor-faktor seperti pemilihan lapisan pada jaringan *inception* yang digunakan untuk ekstraksi fitur (*feature extraction*), dan mungkin belum sepenuhnya merefleksikan persepsi manusia secara akurat dalam semua konteks (LIASHCHYNSKYI & LIASHCHYNSKYI, 2023; Alam & Kehtarnavaz, 2024).

1.6.7 Inception-v3

Model *Inception-v3* merepresentasikan kemajuan dalam arsitektur *deep learning*, mendemonstrasikan efikasi yang cukup besar di berbagai aplikasi, terutama dalam klasifikasi citra dan diagnostik medis. Diperkenalkan pertama kali oleh Szegedy et al. (2015), *Inception-v3* dicirikan oleh desain arsitektur yang menggabungkan konvolusi terfaktorisasi (*factorized convolutions*) dan jalur pemrosesan paralel, memungkinkannya untuk mengekstrak fitur kompleks secara efisien (Szegedy et al., 2015; Chollet, 2017).

Inti dari cara kerja *Inception-v3* terletak pada desain modular yang menggunakan modul *Inception*. Modul ini memungkinkan model untuk mengekstraksi fitur pada berbagai skala resolusi secara simultan dalam satu lapisan tunggal melalui pemrosesan paralel, menggabungkan hasil konvolusi 1x1, konvolusi terfaktorisasi 3x3, dan operasi *pooling*. Model ini juga mengimplementasikan inovasi konvolusi terfaktorisasi, yang memecah operasi konvolusi berukuran besar (misalnya, 5x5) menjadi urutan operasi konvolusi yang lebih kecil (misalnya, dua konvolusi 3x3). Mekanisme ini secara drastis mengurangi biaya komputasi dan jumlah parameter tanpa mengurangi kapasitas representasi model (Szegedy et al., 2015). Selain itu, *Inception-v3* mengadopsi teknik regularisasi canggih seperti *batch normalization* dan menggunakan *label smoothing* untuk meningkatkan kinerja dan mencegah *overfitting*. Efektivitas *Inception-v3* didukung oleh kapabilitas generalisasi yang kuat melalui *transfer learning*, memungkinkannya memanfaatkan *pre-trained weight* dari *dataset* besar seperti *ImageNet* untuk mencapai akurasi yang superior

pada domain khusus, termasuk dalam diagnostik medis dan aplikasi pencitraan berteknologi tinggi. Arsitektur *Inception-v3* dapat dilihat pada gambar berikut.



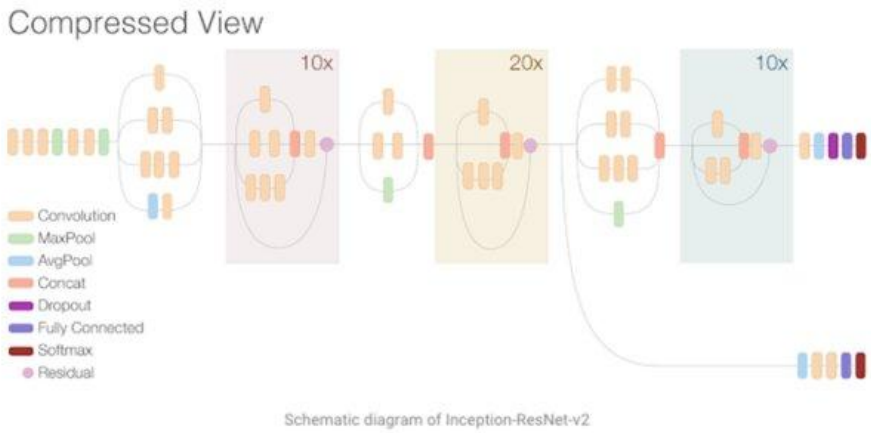
Gambar 4 Arsitektur *Inception-v3* (Sumber: Iparraguirre-Villanueva et al., 2022)

Sebagai contoh, sebuah studi yang membandingkan model untuk deteksi retinopati diabetik menemukan bahwa *Inception-v3* mengungguli *DenseNet201* dan *ResNet50*, mencapai skor *Cohen's Kappa* yang tinggi, mengindikasikan ketangguhannya (*robustness*) dalam menangani *dataset* medis yang kompleks (Charulatha et al., 2023). Demikian pula, ketika diterapkan untuk mendeteksi dan melokalisasi tumor otak dalam pemindaian *MRI*, modifikasi pada arsitektur *Inception-v3* menghasilkan akurasi sebesar 97,2% (Nazmul Arefin et al., 2023). Selain itu, klasifikasi *COVID-19* dari citra pindaian *CT* menyoroti efektivitas *Inception-v3* dalam situasi medis dunia nyata, mencapai metrik kinerja yang superior dibandingkan pengklasifikasi lainnya (Reis, 2021). Kemampuannya untuk menggeneralisasi dengan baik di berbagai aplikasi ditingkatkan oleh arsitekturnya, yang mendukung *transfer learning*, memungkinkannya memanfaatkan bobot yang telah dilatih sebelumnya dari *dataset* besar seperti *ImageNet* untuk akurasi yang lebih baik dalam domain khusus (H. Zhang et al., 2020; Xiao et al., 2018).

Dalam bidang klasifikasi citra yang lebih luas, *Inception-v3* secara konsisten menempati peringkat di antara model-model berkinerja terbaik. Analisis komparatif di berbagai *dataset*, termasuk klasifikasi morfologi galaksi dan deteksi penyakit kulit anjing, semakin menegaskan efikasinya, dengan akurasi yang tercatat sering kali melampaui model kontemporer seperti *VGG16* dan *ResNet* (Qian, 2023; Thoutan et al., 2023) Hasil ini mengonfirmasi fleksibilitas dan keandalan model di seluruh spektrum aplikasi, termasuk pertanian, di mana model ini digunakan untuk deteksi penyakit pada padi, serta teknologi pengenalan wajah tingkat lanjut (Maulana et al., 2023; Hidayat et al., 2023).

1.6.8 Inception-Resnet-v2

Inception-Resnet-v2 adalah arsitektur *deep learning* hibrida yang menggabungkan kekuatan modul *Inception* dengan koneksi residual, bertujuan untuk meningkatkan kinerja sambil mempertahankan efisiensi komputasi. Arsitektur ini diperkenalkan oleh Szegedy et al. (2016) dan telah mendapatkan pengakuan karena akurasinya yang meningkat dalam berbagai tugas klasifikasi citra dibandingkan dengan pendahulunya. Model ini terdiri dari 164 lapisan dan dirancang untuk memproses data berdimensi tinggi secara efektif, memanfaatkan pemrosesan jalur banyak dari blok *Inception* dan konsep pemetaan identitas dari jaringan residu (Szegedy et al., 2016). Arsitektur *Inception-Resnet-v2* dapat dilihat pada gambar berikut.



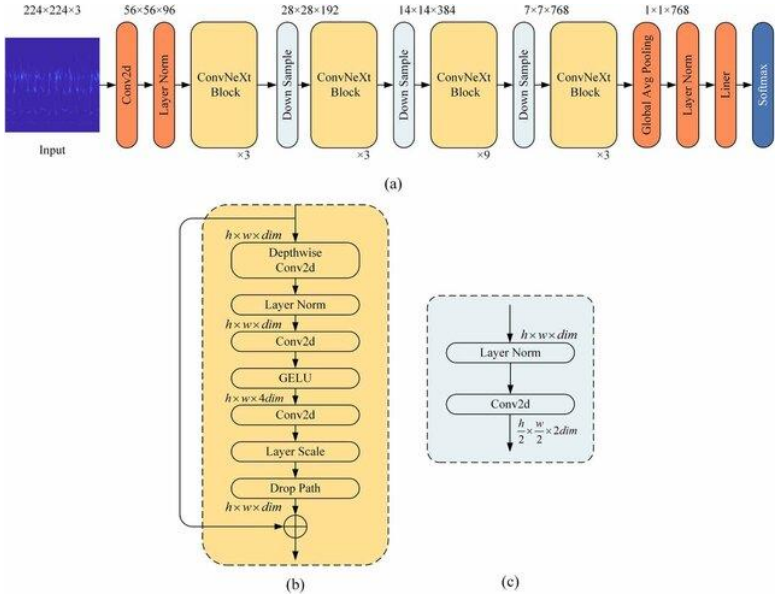
Gambar 5 Arsitektur *Inception-Resnet-v2* (Sumber: Mehta et al. (2018))

Cara kerja inti model ini terletak pada penggunaan *Residual Inception Blocks*, di mana setiap blok modular *Inception* yang berfungsi mengekstrak fitur pada berbagai skala secara paralel, dibungkus dengan koneksi residu atau jalur pintas. Koneksi residu ini memastikan informasi input langsung ditambahkan ke *output* modul, mengatasi masalah degradasi dan memungkinkan model dilatih dengan kedalaman yang ekstrem sambil mempertahankan konvergensi yang cepat dan stabil. Sebuah karakteristik penting dari *Inception-Resnet-v2* adalah penerapan faktor skala yang biasanya bernilai 0.1 hingga 0.3 pada *output* modul *Inception* sebelum penambahan residu. Skala ini krusial untuk mencegah akumulasi nilai aktivasi yang terlalu besar (*activation explosion*) dan menjaga stabilitas jaringan selama tahap awal pelatihan, yang jika diabaikan dapat menyebabkan jaringan mati (*network collapse*). Dengan menggabungkan efisiensi komputasi *Inception* dengan manfaat residu, *Inception-Resnet-v2* telah menjadi arsitektur yang sangat kuat dan efektif, sering digunakan sebagai dasar dalam tugas pengenalan citra tingkat lanjut dan *transfer learning* (Szegedy et al., 2016).

Analisis komparatif terkini telah memosisikan *Inception-Resnet-v2* secara menguntungkan terhadap arsitektur lain seperti *ResNet* dan *DenseNet*, terutama ketika mengevaluasi metrik akurasi dan ketangguhan (Ahn et al., 2021; Mahdianpari et al., 2018). Para peneliti telah mencatat bahwa sifat hibrida dari desainnya mengarah pada peningkatan kinerja sambil memastikan beban komputasi yang dapat dikelola, menjadikannya sesuai untuk aplikasi *desktop* maupun *mobile* (Al-Naji et al., 2024; PROCHUKHAN, 2024). Selain itu, desainnya menggabungkan teknik *dropout* untuk memitigasi *overfitting*, memperkuat penerapan praktisnya dalam skenario dunia nyata (Karabağ et al., 2020).

1.6.9 ConvNeXt-Tiny

ConvNeXt-Tiny adalah salah satu varian dari arsitektur *CNN* modern yang diperkenalkan oleh Liu et al. (2022) dengan tujuan memodernisasi desain *CNN* tradisional agar setara dengan kinerja *Vision Transformer* (*ViT*) dalam tugas pengenalan citra skala besar. Arsitektur *ConvNeXt* dibangun melalui serangkaian eksperimen yang secara bertahap mengadopsi elemen desain utama dari *transformer*, dimulai dari arsitektur *ResNet*. Inti dari cara kerja *ConvNeXt-Tiny* terletak pada blok modular yang dimodifikasi, yang menggabungkan tiga inovasi utama. Pertama, konvolusi *depthwise* berkernel besar, fungsi spasial dipisahkan dari fungsi kanal menggunakan konvolusi *depthwise* 7x7 yang diposisikan di awal blok (bukan di tengah), meniru perhatian lokal dari *transformer*. Kedua, struktur *inverted bottleneck*, blok *ConvNeXt* mengadopsi desain *inverted bottleneck* (mirip dengan blok *FFN* pada *transformer*) di mana dimensi fitur pertama-tama diperluas empat kali lipat menggunakan konvolusi 1x1 dan kemudian dikompresi kembali ke dimensi input, meningkatkan efisiensi komputasi dan ekspresi. Ketiga, normalisasi dan aktivasi *transformer*, arsitektur ini mengganti sebagian besar fungsi aktivasi *ReLU* dengan *GELU* dan mengganti *batch normalization* dengan *layer normalization*, yang terbukti meningkatkan stabilitas dan kesamaan konvergensi dengan arsitektur berbasis *transformer*. Secara keseluruhan, *ConvNeXt-Tiny* mempertahankan struktur bertahap dan koneksi residu dasar *ResNet*, tetapi dengan modifikasi mendalam pada struktur blok yang membuatnya menjadi model dasar yang tangguh dan efisien untuk berbagai tugas visi komputer. Detail arsitektur *ConvNeXt* dapat dilihat pada gambar berikut.



Gambar 6 (a) Arsitektur *ConvNeXt*; (b) *ConvNeXt block*; (c) *down-sampling*. (Sumber: Yu et al. (2024))

Selain itu, *ConvNeXt-Tiny* merupakan varian terkecil dalam keluarga *ConvNeXt*, yang menyediakan model yang mudah diakses untuk penerapan di skenario dengan keterbatasan sumber daya komputasi. Model ini mempertahankan fitur-fitur penting yang bermanfaat untuk mencapai kinerja kompetitif dalam tugas-tugas, termasuk deteksi penyakit dan pengenalan *malware*, sehingga menegaskan relevansinya dalam penelitian akademis maupun aplikasi praktis (Setyo & Kusuma, 2024; Kaur et al., 2025).

Peningkatan kinerja arsitektur dan kemudahan integrasinya dengan kerangka kerja yang sudah mapan semakin memperkuat posisinya dalam deretan solusi pembelajaran mendalam yang terus berkembang. Para peneliti terus mengeksplorasi kapabilitasnya dalam *fine-tuning* untuk berbagai aplikasi, memastikan bahwa *ConvNeXt-Tiny* tetap menjadi subjek yang menarik dalam studi masa depan mengenai arsitektur konvolusi (Alshomrani et al., 2025).

1.6.10 Confusion Matrix

Confusion Matrix adalah alat analisis krusial yang digunakan untuk mengevaluasi kinerja algoritma klasifikasi dalam *machine learning*. Alat ini menyajikan tata letak matriks yang merangkum klasifikasi yang diprediksi terhadap klasifikasi yang sesungguhnya, memungkinkan pemahaman yang lebih bernuansa mengenai seberapa baik kinerja model di berbagai kategori (Palma et al., 2022; Sathyanarayanan & Tantri, 2024). Matriks ini mengategorikan hasil evaluasi menjadi empat kelas fundamental yaitu, *True Positives* (TP), *True Negatives* (TN), *False Positives* (FP), dan *False Negatives* (FN), yang masing-masing memegang peranan

penting dalam penghitungan berbagai metrik kinerja. Berikut rincian *confusion matrix* dalam bentuk tabel.

Tabel 1 *Confusion matrix* yang digunakan pada penelitian

Confusion Matrix		**Actual Values**	
		Positive	Negative
Predicted Values	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Selain 4 kelas tersebut ada beberapa pengukuran performa seperti *accuracy, precision, recall,* dan *f1-score*. Tiap-tiap pengukuran ini memiliki persamaannya sendiri, dirumuskan sebagai berikut:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (3)$$

$$precision = \frac{TP}{TP + FP} \quad (4)$$

$$recall = \frac{TP}{TP + FN} \quad (5)$$

$$F1score = 2\frac{precision\ x\ recall}{precision + recall} \quad (6)$$

Accuracy adalah metrik fundamental yang mengukur proporsi prediksi benar dari total prediksi, ideal digunakan ketika distribusi kelas seimbang. Namun, pada data yang tidak seimbang, akurasi menjadi menyesatkan (*accuracy paradox*), sehingga metrik lain lebih diutamakan. *Precision* fokus pada minimalisir *FP* dengan mengukur fraksi hasil positif yang benar, sedangkan *recall* fokus pada minimalisir *FN*, mengukur proporsi kasus positif aktual yang berhasil dideteksi, yang sangat penting dalam konteks krusial seperti diagnosis medis. Karena *precision* mengabaikan *FN* dan *recall* mengabaikan *FP*, *f1-score* menjadi pilihan terbaik ketika kedua jenis kesalahan (*false*) tersebut sama-sama penting. *F1-score* adalah rata-rata harmonik dari *precision* dan *recall* yang menyeimbangkan kedua metrik tersebut, menjadikannya metrik yang lebih andal untuk mengevaluasi kinerja model klasifikasi pada himpunan data yang mengalami ketidakseimbangan kelas (Sathyanarayanan & Tantri, 2024).

Dalam klasifikasi biner, *confusion matrix* secara efektif membedakan antara kelas positif dan negatif, memungkinkan praktisi memperoleh wawasan yang dapat diterapkan untuk perbaikan model (Sathyanarayanan & Tantri, 2024). Metrik yang didefinisikan diturunkan dari *confusion matrix*, membantu menilai seberapa baik

algoritma mengklasifikasikan *instance*, mendukung pengambilan keputusan berdasarkan akurasi dan keandalannya. Jika sebuah model menghasilkan jumlah FN yang tinggi, hal tersebut mengindikasikan kegagalannya dalam mengidentifikasi *instance* positif secara memadai, pengklasifikasi dapat disesuaikan atau dilatih ulang untuk mengoptimalkan kinerja (Sathyanarayanan & Tantri, 2024).

1.6.11 Fine-Grained Visual Classification (FGVC)

Fine-Grained Visual Classification (FGVC) adalah bagian fundamental dalam *computer vision* yang bertujuan untuk mengklasifikasi sub-kategori yang berbeda di dalam suatu super-kategori, seperti membedakan spesies burung yang berbeda. Tugas ini menimbulkan tantangan unik karena adanya perbedaan antar-kelas yang sangat halus (*subtle inter-class discrepancies*) dan variasi intra-kelas yang besar (*large intra-class variations*) akibat faktor-faktor seperti skala, lokasi, dan deformasi (Sengodan, 2025).

Secara umum, tujuan fundamental pelatihan *CNN* adalah memaksimalkan variasi antar-kelas (*inter-class scatter*) sambil meminimalkan variasi instra-kelas (*intra-class variance*). Namun dalam skenario *FGVC*, model menghadapi dilema. Model harus mampu menggeneralisasi dalam satu kelas (invariansi) dan pada saat yang sama harus sensitif terhadap detail kecil yang menjadi kunci pembeda antar kelas (sensitivitas) (Pilarczyk & Skarbek, 2019). Jaringan konvolusi tradisional, terutama yang menggunakan operasi *pooling*, cenderung memprioritaskan invariansi, yang justru dapat menyebabkan hilangnya informasi diskriminatif penting, sehingga fitur yang dipelajari untuk kelas yang berbeda saling tumpang tindih (Wu et al., 2025).

Ketika *CNN* standar (tanpa modifikasi khusus) diterapkan pada tugas klasifikasi *FGVC*, akurasi puncaknya lebih rendah dibandingkan tugas klasifikasi umum. Sebagai contoh, pengujian dasar pada *dataset* burung *CUB-200-2011* menghasilkan akurasi klasifikasi hanya berkisar antara 73,5% hingga 76,8% (Y. Zhang et al., 2018). Kontras yang tajam ini, dibandingkan dengan akurasi 99% pada *dataset* dengan perbedaan kelas tinggi , menegaskan bahwa akurasi *CNN* secara fundamental berbanding terbalik dengan tingkat kemiripan visual antar-kelas (Isong, 2025).

1.6.12 Transfer Learning

Transfer learning dalam *Convolutional Neural Networks* (CNN) telah muncul sebagai teknik yang kuat, terutama dalam skenario di mana data berlabel langka atau sumber daya komputasi terbatas. Dengan memanfaatkan fitur-fitur yang dipelajari dari jaringan yang sudah dilatih sebelumnya (*pre-trained*), *transfer learning* memungkinkan penerapan model *deep learning* ke domain baru tanpa perlu pelatihan ulang yang ekstensif. Pendekatan ini secara menghemat waktu dan sumber daya komputasi, sekaligus meningkatkan kinerja model di berbagai aplikasi.

Studi fundamental oleh Yosinski et al. (2014) mengungkapkan bahwa pengetahuan yang dipelajari oleh *Deep Neural Networks* (DNN), khususnya *CNN*,

tidaklah seragam di setiap lapisan. Penelitian ini menunjukkan bahwa lapisan bawah (yang lebih dekat ke data *input*) bertindak sebagai ekstraktor fitur generik, mengidentifikasi pola visual universal seperti garis, tepi, dan gumpalan warna; fitur-fitur dasar ini memiliki kemampuan transfer yang tinggi dan dapat digunakan untuk tugas visual apa pun, seperti klasifikasi kucing maupun citra medis. Sebaliknya, lapisan atas (yang lebih dekat ke *output* klasifikasi) berfungsi sebagai ekstraktor fitur spesifik karena mereka menggabungkan pola dasar menjadi konsep kompleks yang hanya relevan dengan tugas pelatihan awal, sehingga memiliki kemampuan transfer yang rendah ke domain baru. Pengamatan ini sangat penting dalam *transfer learning*, karena menyiratkan bahwa strategi yang paling efisien adalah membekukan bobot pada lapisan-lapisan generik di bagian bawah, dan hanya melakukan *fine-tuning* atau *retraining* pada lapisan atas yang lebih spesifik untuk menyesuaikannya dengan masalah baru.

BAB II
METODE PENELITIAN

2.1 Sumber Data

Sumber data penelitian ini adalah karya orisinal seniman Roko (penulis), berupa kumpulan karya seni digital yang memiliki *style* khas. Penggunaan karya-karya ini telah mendapatkan otorisasi penuh dari pemilik hak cipta (Penulis sendiri) dan akan dimanfaatkan untuk menguji kemampuan model *AI* dalam melakukan *style mimicry* terhadap *style* milik Roko.

2.2 Waktu dan Tempat Penelitian

2.2.1 Waktu Penelitian

Penelitian berlangsung kurang lebih selama 5 bulan. Dimulai dengan studi literatur hingga *deploy website*. Adapun detail waktu penelitian dapat dilihat pada tabel berikut.

Tabel 2 Waktu penelitian

Kegiatan	Juni				Juli				Agustus				September				Oktober			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Studi Literatur																				
Persiapan Dataset																				
Evaluasi																				
Website																				

2.2.2 Tempat Penelitian

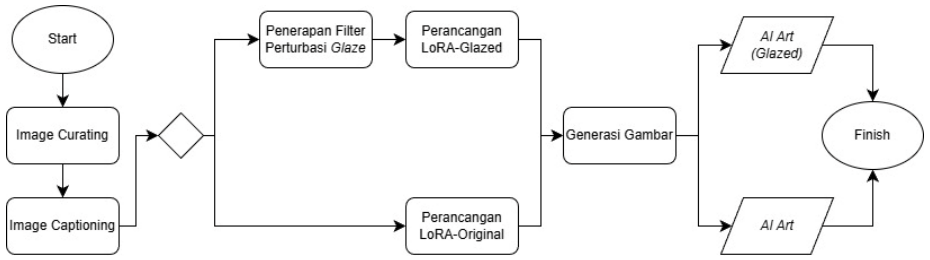
Penelitian dilaksanakan di Lab Rekayasa Perangkat Lunak (RPL), Program Studi Sistem Informasi, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Hasanuddin.

2.3 Tahapan Penelitian

Penelitian ini dibagi menjadi 2 tahapan utama. Tahap pertama adalah persiapan *dataset* dan pelatihan model, yang mencakup pembuatan *dataset AI art* menggunakan gambar-gambar yang dikumpulkan, diikuti oleh pelatihan model *LoRA* melalui *Kohya_ss* dan generasi gambar menggunakan *Automatic1111*. Hasilnya diklasifikasikan menjadi dua kategori *dataset*: *LoRA-ORI* (*AI art* dari *dataset original art*) dan *LoRA-GLZ* (*AI art* dari *dataset glazed art*). Tahap kedua berfokus pada evaluasi kinerja model menggunakan metrik seperti *FID* dan *IS*, serta *supervised learning* dengan arsitektur *CNN* melalui layer akhir model *pre-trained* seperti *Inception-v3*, *Inception-Resnet-v2*, dan *ConvNeXt-Tiny*.

2.3.1 Persiapan Dataset

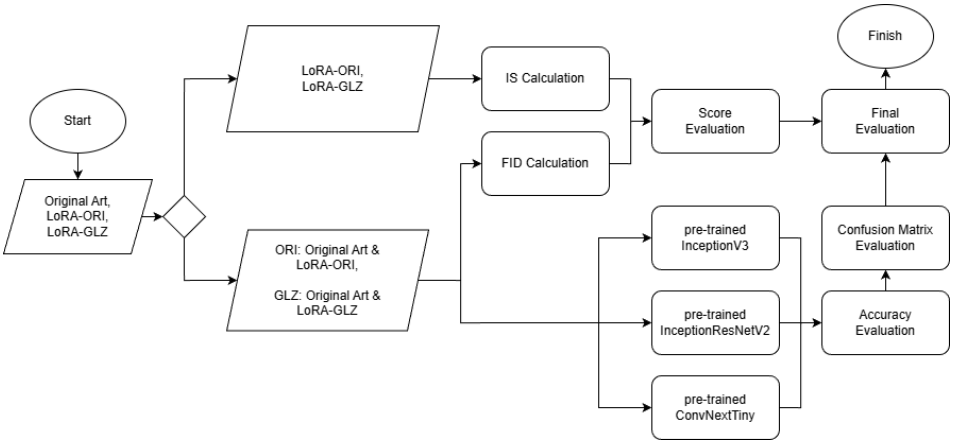
Tahap persiapan *dataset* merupakan fondasi penelitian, diawali dengan *image curating*, di mana kumpulan karya seni Roko dikumpulkan dan dikurasi ketat untuk memastikan kualitas dan keragaman representasi *style* Roko. Setelah itu, *dataset* diberi *caption*, di mana setiap gambar diberi *prompt* agar model *AI LoRA* dapat memahami korelasi teks-gambar selama pelatihan. Langkah krusial berikutnya adalah penerapan filter perturbasi menggunakan aplikasi *Glaze* yang membagi *dataset* menjadi *original art* dan *glazed art*. Kedua *dataset* ini kemudian digunakan dalam perancangan *LoRA*, di mana proses *fine-tuning* dilakukan melalui *Kohya_ss* untuk menghasilkan bobot *LoRA-ORI* dan *LoRA-GLZ*. Tahap ini diakhiri dengan generasi gambar, yaitu penggunaan bobot *LoRA* yang telah dihasilkan ke *AUTOMATIC1111 Stable Diffusion Web UI* (A1111) untuk menghasilkan gambar-gambar baru (*AI art*). Untuk lebih jelasnya dapat dilihat pada diagram berikut.



Gambar 7 Tahapan persiapan *dataset* penelitian

2.3.2 Evaluasi

Tahap evaluasi berfokus pada analisis kinerja model menggunakan *dataset* yang telah dihasilkan dengan metrik yang telah dipilih. Evaluasi dilakukan melalui dua pendekatan utama. Pendekatan pertama menggunakan metrik *FID* dan *IS*. *FID* mengukur realisme dan kesamaan distribusi citra hasil generasi terhadap citra asli, sementara *IS* menilai kejelasan dan keragaman citra hasil generasi saja. Pendekatan kedua adalah *supervised learning* yang memanfaatkan *transfer learning* dengan arsitektur *CNN pre-trained*, berupa *Inception-v3*, *Inception-Resnet-v2*, dan *ConvNeXt-Tiny*. Untuk metode *supervised learning*, model dilatih untuk tugas klasifikasi biner (antara *original art* dan *LoRA-ORI*, juga *original art* dan *LoRA-GLZ*) dengan dataset yang dimensinya telah disesuaikan (299x299 piksel untuk *Inception*, 224x224 piksel untuk *ConvNeXt*). Dalam *transfer learning*, lapisan dasar model dibekukan, dan ditambahkan *custom classification head* sebelum model dilatih menggunakan data latih dan diuji. Kinerja akhir dinilai menggunakan *confusion matrix* berupa *accuracy, precision, recall,* dan *f1-score*. Tahapan yang lebih detail pada tiap metrik evaluasi dapat dilihat pada diagram-diagram berikut.



Gambar 8 Tahapan evaluasi penelitian

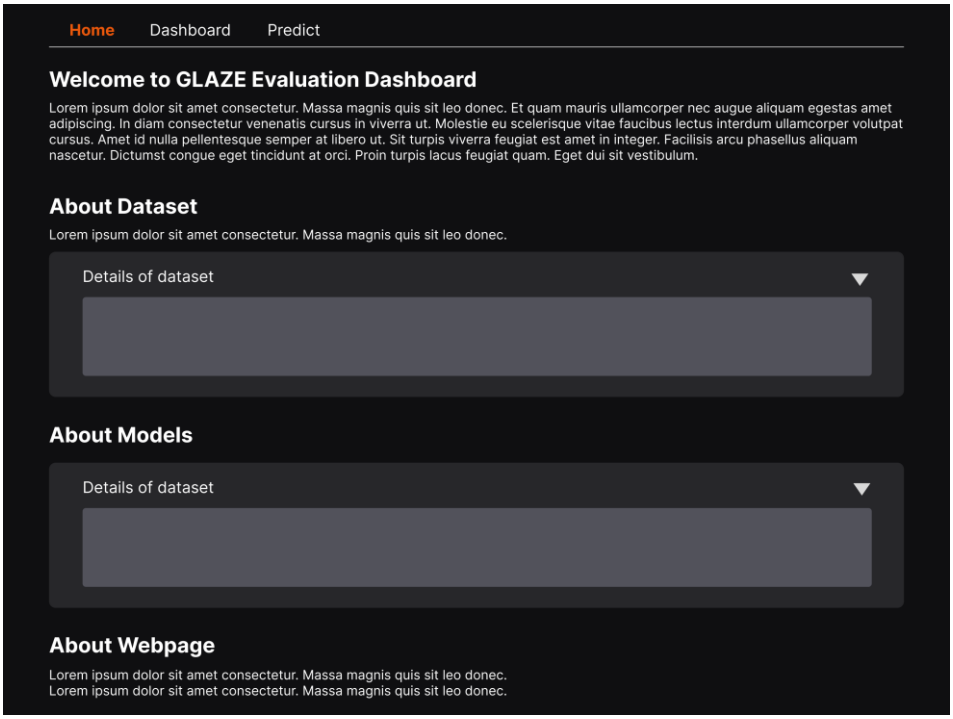
2.4 Rancangan Website dengan Gradio

Pengembangan aplikasi web pada penelitian ini diimplementasikan sebagai *dashboard* interaktif yang di-*deploy* melalui *Hugging Face Hub Spaces*. Pengembangan *dashboard* ini sepenuhnya memanfaatkan bahasa pemrograman *python* sebagai bahasa utama, didukung oleh *framework Gradio*. Penggunaan *Gradio* dipilih karena kemampuannya yang efisien dalam membangun antarmuka pengguna grafis (GUI) yang langsung terintegrasi dengan fungsi-fungsi model *machine learning* yang dikembangkan dan juga *Hugging Face Hub Space*.

Aplikasi web ini dirancang dengan tiga halaman utama. Halaman beranda berfungsi sebagai tampilan pertama yang diakses pengguna. Halaman ini menyajikan deskripsi singkat, latar belakang, dan tujuan aplikasi web. Halaman analisis model merupakan *dashboard* yang menampilkan visualisasi hasil penelitian. Terakhir, Halaman prediksi gambar berfungsi sebagai alat pendeteksi *AI art*.

2.4.1 Halaman Beranda

Halaman beranda merupakan halaman pertama yang dilihat ketika pengguna mengakses *website*. Untuk visualisasi perancangan, digunakan *wireframe*. Berikut detail halaman beranda.

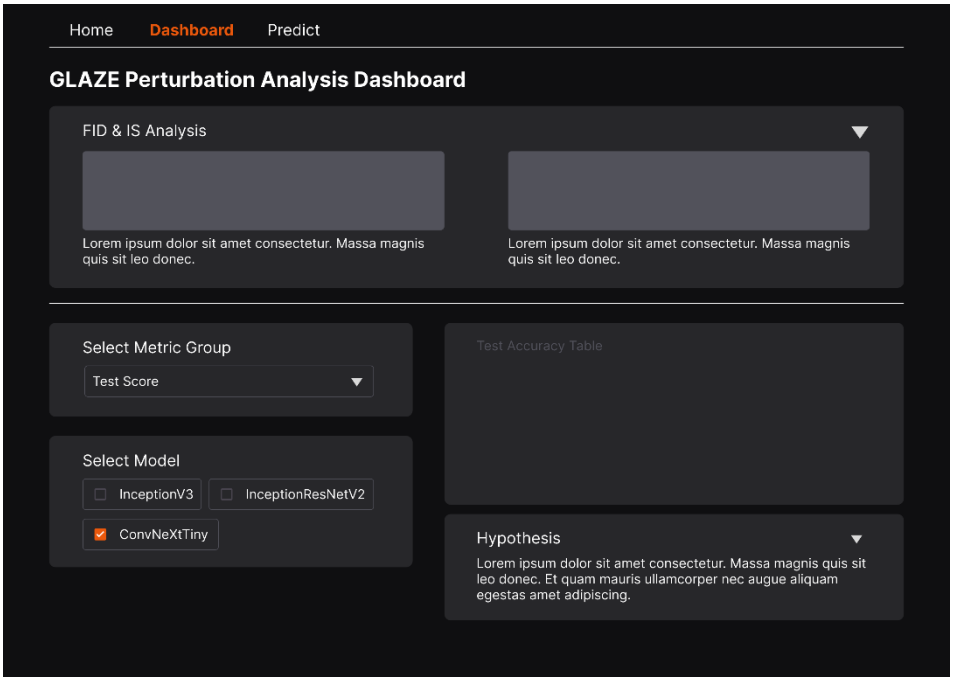


Gambar 9 Rancangan halaman beranda

Dapat dilihat pada gambar 9, terdapat tiga *tab*: *Home*, *Dashboard*, dan *Predict* dengan penanda berwarna oranye saat pengguna berada di halaman yang dipilih. *Tab* ini digunakan sebagai navigasi pengguna untuk berpindah halaman. Di bawah navigasi akan berisi informasi dari *tab* yang dipilih. Untuk halaman beranda, terdapat judul dan deskripsi ringkas mengenai situs web dan penelitian terkait. Kemudian, ada penjelasan *dataset* dan model yang digunakan untuk penelitian, disajikan menggunakan menu *accordion* yang dapat dibuka dan ditutup. Terakhir, terdapat informasi tentang *webpage* mengenai *tab Dashboard* dan *Predict*.

2.4.2 Halaman Analisis Model

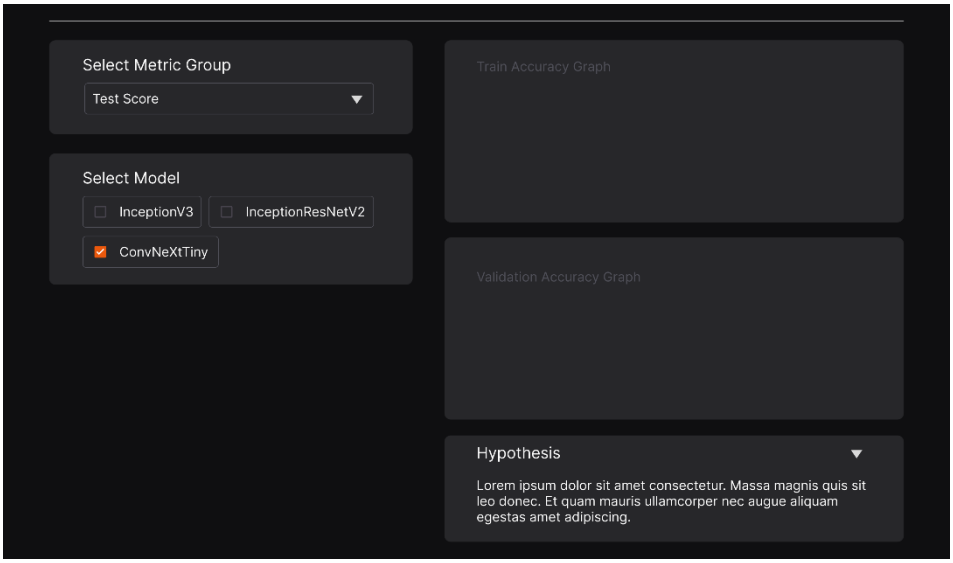
Halaman analisis model atau yang disebut juga *dashboard*, berisi hasil dari penelitian yang divisualisasikan. Perincian halaman analisis model adalah sebagai berikut.



Gambar 10 Rancangan halaman *dashboard* awal *load*

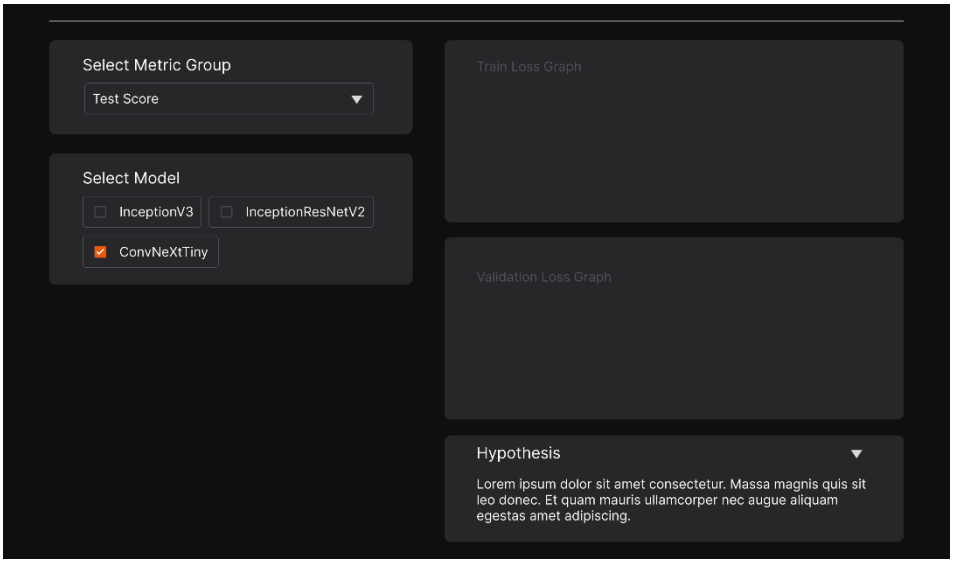
Sama seperti halaman beranda, halaman analisis juga diawali dengan *tab* navigasi di bagian paling atas,. menampilkan tampilan pertama saat memilih *tab Dashboard*. Dimulai dari atas, terdapat judul lalu ada menu *accordion* tentang analisis *FID* dan *IS*. Di dalam menu tersebut, terdapat tabel hasil skor *FID* dan *IS* beserta penjelasan skor. Kemudian, dipisahkan oleh garis horizontal, di bagian kiri terdapat beberapa elemen interaktif untuk pengguna. Yang pertama seleksi metrik dalam bentuk *dropdown*, tersedia empat pilihan metrik untuk dipilih pengguna. Pada awal *load*, pilihan metrik akan otomatis memilih *test score*, yang akan menampilkan hasil akhir semua *test accuracy* model dengan sorotan oranye untuk skor tertinggi dalam tabel. Di bawahnya terdapat seleksi model dalam bentuk *radio button*, di mana pengguna dapat memilih model yang ingin dilihat visualisasi grafiknya. Terdapat tiga model yang dapat dipilih, yaitu *Inception-v3*, *Inception-Resnet-v2*, dan *ConvNeXt-Tiny*. Kemudian, di bawah visualisasi *test accuracy* terdapat sebuah menu *accordion* yang berisi tentang hipotesis penelitian, yang akan menjelaskan korelasi nilai skor terhadap kesimpulan efektivitas perturbasi *Glaze*.

Ketika pengguna memilih *accuracy* pada menu metrik, di bagian kanan *test accuracy* berubah menjadi dua grafik *lineplot* yaitu, *train accuracy* dan *validation accuracy*. Lebih jelasnya dapat dilihat di gambar berikut.



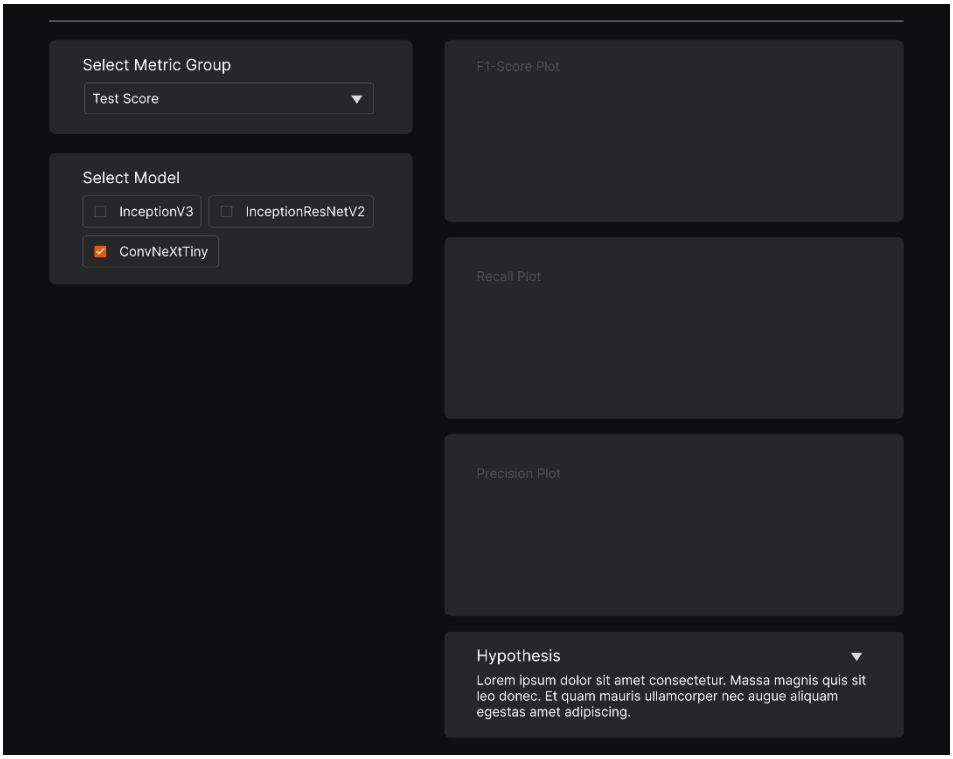
Gambar 11 Rancangan halaman *dashboard dropdown: Accuracy*

Mirip dengan pilihan *accuracy*, ketika pengguna memilih *loss* pada menu metrik, di bagian kanan akan muncul dua grafik *lineplot* tidak lain, *training loss* dan *validation loss*. Lebih jelasnya dapat dilihat pada gambar berikut.



Gambar 12 Rancangan halaman *dashboard dropdown: Loss*

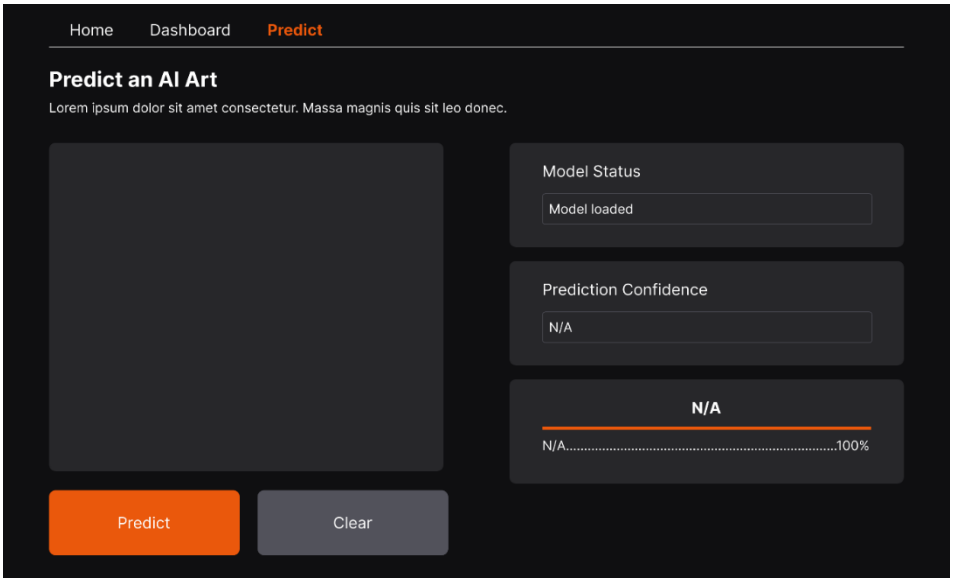
Pilihan metrik terakhir yaitu, *confusion matrix metrics* akan menampilkan tiga grafik *barplot* berupa *f1-score*, *recall*, dan *precision*. Lebih jelasnya dapat dilihat pada gambar berikut.



Gambar 13 Rancangan halaman *dashboard dropdown: Confusion Matrix Metrics*

2.4.3 Halaman Prediksi Gambar

Halaman terakhir adalah halaman prediksi gambar, yang merupakan implementasi model dari hasil penelitian. Halaman ini berfungsi untuk memprediksi apakah sebuah gambar merupakan hasil generasi *AI* atau merupakan karya orisinal manusia. Perinciannya dapat dilihat pada gambar berikut.

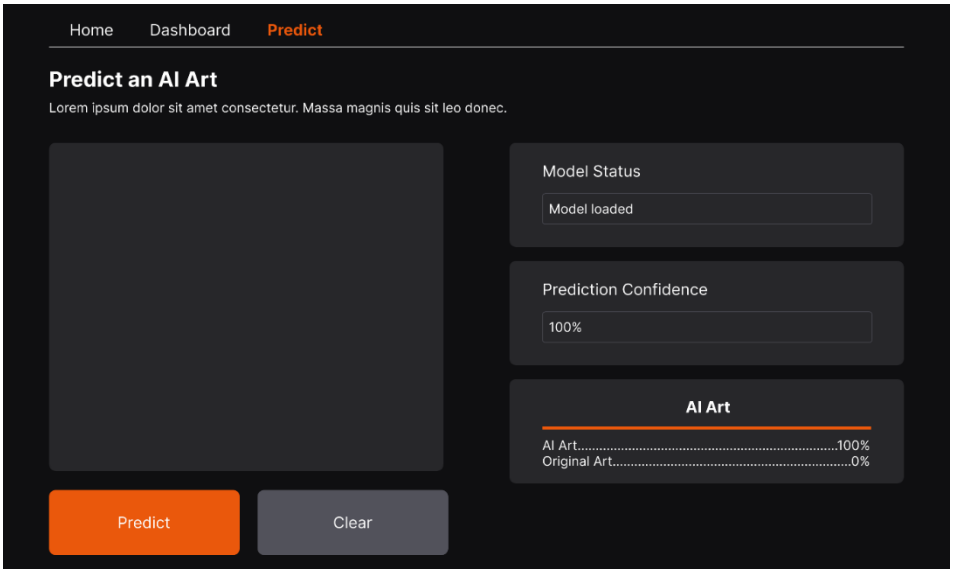


Gambar 14 Rancangan halaman prediksi awal *load*

Gambar di atas menampilkan tampilan awal dari halaman prediksi. Di bawah navigasi terdapat judul dan penjelasan ringkas mengenai model yang digunakan untuk prediksi. Kemudian, tampilan terpisah menjadi dua, di bagian kiri terdapat menu *input* gambar dengan dua tombol di bawahnya untuk interaksi pengguna. Tombol *predict* berwarna oranye untuk memulai prediksi dan tombol *clear* berwarna abu-abu untuk mengatur ulang semua *input*.

Di bagian kanan terdapat status model jika model berhasil di muat dan siap untuk melakukan prediksi. Di bawahnya terdapat *prediction confidence* yang akan menampilkan skor *confidence*. Dan di bawahnya lagi merupakan klasifikasi akhir dari prediksi: *AI art* dan *original art*.

Untuk tampilan setelah dilakukan prediksi, dapat dilihat pada gambar berikut.



Gambar 15 Rancangan halaman prediksi setelah melakukan prediksi

Gambar 15 menampilkan tampilan setelah pengguna melakukan prediksi melalui interaksi tombol *predict*. *Prediction score* akan menampilkan persentase skor *confidence* dan di bawahnya terdapat label klasifikasi akhir yang menampilkan label prediksi (*AI art/original art*) beserta persentase klasifikasinya.

2.5 Instrumen Penelitian

Adapun detail perangkat keras dan perangkat lunak yang digunakan selama penelitian untuk mempermudah penelitian adalah sebagai berikut.

2.5.1 Perangkat Keras (*hardware*)

Berikut detail spesifikasi perangkat keras yang digunakan dalam penelitian.

Tabel 3 Tabel spesifikasi perangkat keras

No	Perangkat Keras	Spesifikasi
1	Operating System	Windows 11 Home Single Language
2	Processor	AMD Ryzen 5 5600U with Radeon Graphics (2.30 Hz)
3	RAM	16 GB
4	GPU	Integrated Radeon Vega 7 graphic

2.5.2 Perangkat Lunak (*software*)

Berikut detail perangkat lunak dan fungsinya yang digunakan dalam penelitian.

Tabel 4 Tabel perangkat lunak

No	Perangkat Lunak	Spesifikasi
1	Glaze	Aplikasi *open-source* yang dapat memberikan perturbasi kepada gambar untuk melindungi karya seni digital dari *style mimicry*.
2	Runpod.io	Layanan *cloud computing* yang menyediakan akses ke *GPU* berkualitas tinggi secara *on-demand* untuk komputasi intensif seperti training *AI* dan *deep learning*.
3	Jupyter Notebook	Aplikasi web interaktif untuk membuat berbagai dokumen yang berisi kode (terutama *python*), visualisasi, dan teks narasi (digunakan untuk eksperimen dan analisis data).
4	Kohya_ss	*Toolkit GUI* yang populer digunakan untuk melakukan *fine-tuning* atau pelatihan model *stable diffusion* dengan metode seperti *LoRA.*
5	Automatic1111	Antarmuka web (WebUI) yang paling populer untuk menjalankan dan model *stable diffusion* untuk tugas seperti *text-to-image*, *img2img*, *inpainting*, dll.
6	Tensorboard	*Platform* visualisasi yang dikembangkan oleh *Google* untuk memantau metrik pelatihan *machine learning* dan membuat grafik data yang mengalir dalam *TensorFlow*.
7	Visual Studio Code	*Code editor* untuk pemrograman penelitian ini.
8	Numpy	*Library python* yang digunakan untuk operasi numerik dan manipulasi *array* multi-dimensi atau matriks
9	Torch	*Library python open-source* yang digunakan untuk membangun *neural network* dan melakukan komputansi *tensor*, inti dari *deep learning* (*pyTorch*)
10	Torchvision	Paket untuk *PyTorch* yang menyediakan model populer, kumpulan data, dan transformasi gambar untuk *computer vision*.

No	Perangkat Lunak	Spesifikasi
11	Scipy	*Library python* yang menyediakan algoritma dan alat untuk komputasi ilmiah dan teknis, seperti statistik, optimasi, dan integrasi.
12	tqdm	*Library python* yang berfungsi untuk menampilkan *progress bar* (indikator kemajuan) saat melakukan perulangan dalam program.
13	Pillow	*Library python* untuk pemrosesan gambar (image processing).
14	Pandas	*Library python* yang digunakan untuk pengelolaan dataset yang berupa *dataframe*.
15	Pickle	*Library python* untuk melakukan serialisasi (penyimpanan) dan deserialisasi (pemuatan ulang) objek *python* atau model *machine learning* agar efisien.
16	Matplotlib	*Library* yang digunakan untuk visualisasi data seperti grafik, plot, dsb.
17	Seaborn	Membantu *Matplotlib* dalam visualisasi data (khususnya membuat visualisasi statistik yang lebih menarik dan informatif).
18	Scikit-learn	*Library* yang digunakan untuk algoritma *machine learning* dan *data mining* (klasifikasi, regresi, *clustering*, *preprocessing*, dll.)
19	Tensorflow	*Library python* yang digunakan untuk *machine learning* (khususnya *deep learning*) dikembangkan oleh *Google*.
20	Python	Bahasa pemrograman yang digunakan dalam penelitian ini.
21	Hugging Face	*Platform* dan komunitas *open-source* yang menyediakan *model hub* (repositori ribuan model *ML/AI*, terutama *NLP*), *Dataset Hub*, dan *Spaces* (demo aplikasi *AI*).
22	Gradio	*Framework open-source* untuk membangun antarmuka web (UI) yang cepat dan interaktif bagi model *machine learning* atau fungsi *python*, memudahkan demo dan berbagi.