IMPLEMENTASI DIFFUSION MODELS DAN DREAMBOOTH UNTUK GENERATOR GAMBAR KARAKTER JEPANG BERGAYA RETRO

LAPORAN TUGAS AKHIR

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

DI SUSUN OLEH :
MUHAMMAD SYIHAB HABIBI
32602100088

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

2025

FINAL PROJECT

IMPLEMENTATION OF DIFFUSION MODELS AND DREAMBOOTH FOR GENERATING RETRO-STYLE JAPANESE CHARACTER IMAGES

Proposed to complete the requirement to obtain a bachelor's degree (S-1) at Informatics Engineering Departement of Industrial Technology Faculty Sultan Agung Islamic University

ARRANGED BY :
MUHAMMAD SYIHAB HABIBI
32602100088

MAJORING OF INFORMATICS ENGINEERING
INDUSTRIAL TECHNOLOGY FACULTY
SULTAN AGUNG ISLAMIC UNIVERSITY
SEMARANG
2025

LEMBAR PENGESAHAN
TUGAS AKHIR

IMPLEMENTASI DIFFUSION MODELS DAN DREAMBOOTH UNTUK GENERATOR GAMBAR KARAKTER JEPANG BERGAYA RETRO

MUHAMMAD SYIHAB HABIBI
32602100088

Telah dipertahankan di depan tim penguji ujian sarjana tugas akhir
Program Studi Teknik Informatika
Universitas Islam Sultan Agung
Pada tanggal : 17 Februari 2025

TIM PENGUJI UJIAN SARJANA :

Ghufron, ST, M.Kom NIDN. 0609108802 (Penguji 1)	.....	17 Februari 2025 .....
Sam Farisa Chaerul Haviana, ST, M.Kom. NIDN. 0602079005 (Penguji 2)	.....	17 Februari 2025 .....
Ir. Sri Mulyono, M.Eng NIDN. 0626066601 (Pembimbing)	.....	17 Februari 2025 .....

Semarang, 17 Februari 2025
Mengetahui,
Kaprodi Teknik Informatika
Universitas Islam Sultan Agung

Moch. Taufik, ST., MIT
NIDN. 0622037502

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Muhammad Syihab Habibi

NIM : 32602100088

Judul Tugas Akhir : IMPLEMENTASI DIFFUSION MODELS DAN DREAMBOOTH UNTUK GENERATOR GAMBAR KARAKTER JEPANG BERGAYA RETRO

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

Semarang, 5 - Maret - 2025

Yang Menyatakan,

Muhammad Syihab Habibi

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Muhammad Syihab Habibi

NIM : 32602100088

Program Studi : Teknik Informatika

Fakultas : Teknologi industri

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : IMPLEMENTASI DIFFUSION MODELS DAN DREAMBOOTH UNTUK GENERATOR GAMBAR KARAKTER JEPANG BERGAYA RETRO

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

Semarang, 5 - Maret - 2025

Muhammad Syihab Habibi

KATA PENGANTAR

Dengan mengucap syukur alhamdulillah atas kehadirat Allah SWT yang telah memberikan rahmat dan karunianya kepada penulis, sehingga dapat menyelesaikan Tugas Akhir dengan judul "Generator Gambar Karakter Jepang Bergaya Retro Menggunakan Diffusion Models Dan Dreambooth" ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana (S-1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.

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

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

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

.

Semarang,

Muhammad Syihab Habibi

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

ABSTRAK

Permintaan gambar karakter Jepang bergaya retro meningkat seiring dengan kebutuhan industri kreatif, seperti game, desain, dan seni digital. Stable Diffusion mendukung pengembangan generator gambar namun memiliki tantangan utama berupa inkonsistensi output visual dan risiko overfitting selama proses pelatihan model generatif. Penelitian ini bertujuan untuk mengatasi permasalahan tersebut dengan teknik fine-tuning DreamBooth pada model Stable Diffusion v2.1, disertai penerapan regularisasi L2 untuk meningkatkan generalisasi model. Hasil uji coba menggunakan CLIP menunjukkan peningkatan kesesuaian gambar terhadap deskripsi teks, dengan skor yang meningkat dari sekitar 0.30-an menjadi 0.70-an setelah proses fine-tuning, sementara regularisasi L2 turut meningkatkan skor sebesar 0.02 dan membantu mengurangi risiko overfitting. Namun, ditemukan kendala dalam penerjemahan prompt, di mana beberapa gambar belum sepenuhnya merepresentasikan instruksi teks, sehingga penelitian lebih lanjut diperlukan untuk meningkatkan pemahaman model terhadap variasi bahasa.

Kata Kunci: *Diffusion models*, Stable Diffusion, DreamBooth Fine-Tuning, Regularisasi L2, Inkonsistensi *Output, Overfitting*.

ABSTRACT

The demand for retro-style Japanese character images has been increasing in line with the needs of the creative industries, such as gaming, design, and digital art. Stable Diffusion supports the development of image generators but faces major challenges, including visual output inconsistencies and the risk of overfitting during the training process of generative models. This study aims to address these issues by employing the fine-tuning technique of DreamBooth on the Stable Diffusion v2.1 model, along with the application of L2 regularization to enhance model generalization. CLIP-based testing results indicate an improvement in image-text alignment, with scores increasing from around 0.30 to 0.70 after fine-tuning, while L2 regularization further boosts the score by 0.02 and helps mitigate overfitting risks. However, challenges remain in prompt interpretation, as some images do not fully represent the given textual instructions, highlighting the need for further research to improve the model's understanding of language variations.

Keywords: Diffusion models, Stable Diffusion, DreamBooth Fine-Tuning, L2 Regularization, Output Inconsistency, Overfitting.

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Di era digital saat ini, perkembangan te knologi kecerdasan buatan (AI) telah memberikan dampak di berbagai bidang, termasuk seni dan desain. Salah satu inovasi adalah penggunaan model generatif untuk menciptakan gambar dan karya seni salah satu pedekatannya adalah *Diffusion models*. *Diffusion models* bekerja dengan memanfaatkan proses *noise* dan *de-noising* untuk membangun gambar dari *noise* acak menjadi representasi visual yang lebih jelas (Anderson dan Akram, 2024). Model ini unggul dalam menghasilkan detail halus serta mempertahankan konsistensi dalam gaya visual, menjadikannya sangat cocok untuk menciptakan karakter Jepang yang memiliki desain khas, seperti ekspresi yang dinamis, pakaian berwarna cerah, dan detail yang rumit.

Karakter Jepang, yang mencakup berbagai gaya seperti anime, manga, hingga desain video game, telah menjadi salah satu bentuk seni yang paling populer di dunia. Desain karakter ini seringkali mencerminkan estetika yang unik, seperti proporsi tubuh yang khas dan warna-warna cerah, yang menjadi ciri khas dalam berbagai genre dan tema. Seiring dengan pesatnya perkembangan industri hiburan, kebutuhan untuk menghasilkan karakter Jepang secara otomatis semakin meningkat, baik untuk aplikasi komersial, pengembangan game, maupun proyek kreatif pribadi (Firdaus, 2023). Hal ini membuka peluang untuk mengembangkan alat yang dapat membantu seniman, desainer, dan pengembang dalam menciptakan karakter dengan efisiensi yang lebih tinggi.

Untuk menghasilkan gambar karakter Jepang yang sesuai dengan preferensi pengguna, penggunaan teknik *fine-tuning* pada model generatif seperti *Diffusion models* menjadi salah satu solusi yang ditawarkan. *Fine-tuning* memungkinkan model untuk dilatih kembali dengan data yang lebih terarah dan khusus, sehingga bisa menghasilkan karakter yang sesuai dengan gaya atau tema tertentu. Pendekatan *Fine-tuning* seperti Dreambooth, sendiri

memungkinkan model untuk menggabungkan atribut tertentu dari karakter yang telah ada ke dalam gambar yang dihasilkan, desainer dapat menyesuaikan karakteristik seperti ekspresi wajah, warna rambut, dan gaya berpakaian sehingga sesuai dengan identitas karakter yang diinginkan. Teknik ini memperkuat potensi personalisasi dalam karya seni digital, yang sangat berguna dalam bidang seperti pembuatan konten game, animasi, dan aplikasi kreatif lainnya (Hidalgo *dkk.*, 2023).

Selain *fine-tuning*, penggunaan teknik regularisasi juga diperlukan untuk mencegah *overfitting* dalam proses generalisasi model. Salah satu teknik regularisasi adalah regularisasi L2, yang bekerja dengan menambahkan penalti terhadap besarnya bobot model pada fungsi *loss* (Hutagalung, 2024). Dalam konteks *Diffusion models*, regularisasi L2 dapat membantu model untuk menghindari terlalu banyak menyesuaikan parameter dengan data pelatihan, sehingga menghasilkan gambar karakter Jepang yang lebih konsisten dan realistis.

Secara keseluruhan, penggunaan *Diffusion models* yang didukung oleh teknik *fine-tuning* seperti Dreambooth serta penerapan regularisasi L2 digunakan dalam menghasilkan gambar karakter Jepang dengan gaya visual tertentu. Salah satu gaya yang menarik perhatian dalam desain karakter Jepang adalah gaya retro, yang terinspirasi oleh estetika visual dan elemen desain dari era 80-an hingga 90-an. Karakter-karakter Jepang bergaya retro sering kali menonjolkan warna-warna pastel, garis-garis tegas, serta pengaruh dari desain grafis dan animasi masa lalu yang memiliki nuansa nostalgia namun tetap relevan dengan tren kontemporer (Lailiyah, 2024). Gaya retro ini tidak hanya memberikan sentuhan artistik yang khas tetapi juga menciptakan rasa kenangan bagi banyak orang yang tumbuh dengan budaya pop Jepang pada masa itu. Studi ini bertujuan untuk menjelajahi lebih lanjut potensi *Diffusion models* dan Dreambooth dalam mengembangkan alat generatif yang mendukung kreativitas para seniman dan desainer di era modern.

1.2 Perumusan Masalah

Rumusan masalah dalam penelitian ini adalah sebagai berikut :

a. Bagaimana membangun generator gambar karakter Jepang dengan gaya Retro menggunakan *diffusion models*?
b. Bagaimana pemrosesan data dan regularisasi berpengaruh terhadap kualitas gambar karakter Jepang yang dihasilkan?

1.3 Pembatasan Masalah

Pembatasan masalah di bawah ini bertujuan untuk menghindari adanya kegiatan di luar sasaran, sehingga dalam pembuatan laporan ini perlu ditentukan suatu batasan masalah sebagai berikut

a. Penelitian ini menggunakan Stable Diffusion v2.1 sebagai model utama, dengan proses *fine-tuning* yang dilakukan menggunakan metode DreamBooth.
b. Penelitian ini berfokus pada pembuatan gambar tokoh Jepang dua dimensi (2D) tanpa mengkaji aspek animasi. Gambar yang dihasilkan mencakup pose statis maupun dinamis, atribut visual dasar (seperti aksesori sederhana), latar belakang, serta tindakan yang dilakukan oleh karakter. Model ini hanya difokuskan pada visualisasi karakter bergaya Retro.
c. Sistem hanya mendukung generasi satu tokoh per input, tanpa kemampuan menghasilkan beberapa karakter dalam satu proses. Selain itu, sistem tidak menyediakan fitur editing gambar secara langsung, sehingga pengguna hanya menerima hasil akhir sesuai dengan *prompt* yang dimasukkan.
d. Pengguna hanya dapat memasukkan deskripsi teks (*prompt*) dalam bahasa Inggris melalui antarmuka yang disediakan. Input lain seperti sketsa, gambar referensi, atau audio tidak termasuk dalam cakupan penelitian. Waktu pemrosesan bergantung pada kompleksitas *prompt* dan kapasitas server yang digunakan.

1.4 Tujuan

Adapun tujuan dari penelitian ini adalah :

a. Membangun generator gambar karakter Jepang bergaya Retro yang menggunakan *diffusion models*.
b. Mengevaluasi performa *diffusion models* dalam menghasilkan gambar karakter Jepang dari segi detail visual dan kesesuaian dengan gaya yang diinginkan

1.5 Manfaat

a. Kemudahan Pembuatan Karakter Visual. Membantu penciptaan karakter visual bergaya unik tanpa memerlukan keterampilan menggambar manual.
b. Memberikan referensi gambar karakter Jepang bergaya retro untuk mendukung berbagai kebutuhan di bidang seni dan hiburan.

1.6 Sistematika Penulisan

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

BAB 1 : PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang pemilihan judul tugas akhir "Generator Gambar Karakter Jepang Bergaya Retro Menggunakan *Diffusion models* Dan Dreambooth". Rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, dan sistematika penulisan.

BAB 2 : TINJAUAN PUSTAKA DAN DASAR TEORI

Bab ini memuat dasar teori yang berfungsi sebagai sumber dalam memahami permasalahan yang dipilih.

BAB 3 : METODE PENELITIAN

Bab ini menjelaskan proses tahapan- tahapan penelitian dimulai dari analisa kebutuhan sistem, kemudian perancangan sistem hingga selesai dibuat.

BAB 4: HASIL PENELITIAN DAN IMPLEMENTASI SISTEM

Bab ini menjelaskan mengungkapkan hasil penelitian yang berupa pembuatan generator gambar Karakter Jepang menggunakan *diffusion models*.

BAB 5: KESIMPULAN DAN SARAN

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

BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Model generatif dalam bidang computer vision telah menjadi area penelitian yang berkembang pesat, di mana beberapa pendekatan utama meliputi *Variational Autoencoders* (VAEs), *Generative Adversarial Networks* (GANs), *Energy-Based Models*, *Autoregressive Models*, dan *Normalizing Flows*. Setiap model memiliki karakteristik dan pendekatan tersendiri dalam membangkitkan data baru yang realistis dari distribusi data yang ada. Dalam penelitian yang berjudul "*Diffusion models in Vision: A Survey*" menyatakan bahwa *diffusion models* dinilai unggul dalam hal kualitas gambar yang dihasilkan, namun masih memerlukan banyak langkah dalam proses inferensi, sehingga waktu komputasinya cukup lama. Proses *sampling* yang melibatkan banyak tahapan membutuhkan waktu komputasi yang lama, menjadikan DDM kurang efisien dibandingkan dengan GANs atau VAEs dalam hal kecepatan (Croitoru *dkk.*, 2023).

Penelitian lain yang berjudul "LaDiffGAN: *Training GANs with Diffusion Supervision in Latent Spaces*" menjelaskan tentang pendekatan inovatif dengan menggabungkan kekuatan *diffusion models* dan GAN dalam tugas penerjemahan gambar yang tidak terawasi. Pada LaDiffGAN, proses difusi diterapkan dalam ruang laten untuk memperkuat representasi fitur GAN. Dengan menggabungkan GAN dengan supervisi difusi, model ini mengatasi beberapa keterbatasan *diffusion models*, seperti *data leakage*, biaya inferensi yang tinggi, dan ketergantungan pada data besar. Meskipun demikian, kompleksitas pelatihan model ini masih menjadi salah satu tantangan yang dihadapi (Liu *dkk.*, 2024) .

Penggunaan *diffusion models* dalam konteks generasi gambar menggambarkan kekuatan mereka dalam memahami dan merepresentasikan kompleksitas data visual dan hubungan antara teks dan gambar. Dalam "*Understanding Diffusion models: A Unified Perspective*" tentang kemampuan *diffusion models* dalam menghasilkan gambar berbasis teks,

seperti pada proyek-proyek terkenal seperti Imagen dan DALL-E 2. meskipun *diffusion models* unggul dalam hasil visual, kelemahan utamanya adalah proses *sampling* yang membutuhkan banyak langkah denoising, sehingga menjadi mahal secara komputasi. Selain itu, kesulitan dalam mempelajari *latens* yang bermakna juga menjadi tantangan yang perlu diatasi dalam penelitian lebih lanjut (Luo, 2022) .

Pewarnaan (colorization) adalah langkah penting dalam proses produksi animasi yang melibatkan transformasi gambar garis (*line drawing*) menjadi gambar penuh warna. Pewarnaan manual memakan waktu lama dan membutuhkan ketelitian tinggi, yang mengakibatkan adanya kebutuhan untuk mengembangkan metode otomatis guna mempercepat proses ini. Dalam penelitiannya yang berjudul "*AnimeDiffusion: Anime Face Line Drawing Colorization via Diffusion models*" terkait pengembangan sebuah model pertama yang secara khusus dirancang untuk pewarnaan gambar garis wajah anime menggunakan *diffusion models*. Meskipun begitu, model ini memiliki keterbatasan, seperti ketergantungan pada data pelatihan berpasangan dan gaya tertentu, yang membuatnya kurang fleksibel dalam berbagai gaya anime (Cao *dkk.*, 2023).

Stable Diffusion memiliki keterbatasan signifikan karena dilatih pada *dataset* tertentu, yang membatasi kemampuannya untuk menghasilkan gambar di luar data yang telah diberikan. Dalam penelitian berjudul "*Personalizing Text-to-Image Diffusion models by Fine-tuning Classification for AI Applications*" penggunaan *Hypernetworks* dan *DreamBooth* menunjukkan peningkatan fleksibilitas pada *Stable Diffusion*, model dapat diadaptasi untuk memperkenalkan gambar baru secara efisien tanpa memerlukan pelatihan ulang ekstensif (Hidalgo *dkk.*, 2023). Dengan memperluas kemampuan *Stable Diffusion* menggunakan metode yang efisien, penelitian ini membuka peluang bagi inovasi di berbagai bidang dan memperkuat peran teknologi AI dalam memberikan solusi yang adaptif.

Seiring dengan semakin berkembangnya ukuran model, tantangan muncul dalam hal kemampuan komputasi yang diperlukan untuk menjalankan model.

Untuk mengatasi tantangan ini, penelitian dengan judul "*Parameter-Efficient Fine-tuning for Large Models: A Comprehensive Survey*" melakukan pendekatan *Parameter Efficient Fine-tuning* (PEFT) yang memungkinkan penyesuaian model besar untuk tugas tertentu dengan mengurangi jumlah parameter tambahan yang diperkenalkan. Sehingga menghemat sumber daya komputasi yang diperlukan (Han *dkk.*, 2024).

Fine-tuning adalah teknik yang umum digunakan untuk mengadaptasi model yang telah dilatih sebelumnya agar lebih sesuai dengan tugas spesifik, mengurangi risiko *overfitting* yang sering terjadi ketika melatih model dari awal dengan data terbatas. Dalam penelitian berjudul "*Text to Image Latent Diffusion Model with Dreambooth Fine Tuning for Automobile Image Generation*" pendekatan *fine-tuning* dilakukan dengan menggunakan model *pre-trained* seperti DreamBooth diharapkan dapat meningkatkan performa *Latent Diffusion Model* (LDM) dalam menghasilkan gambar mobil dari teks, meskipun dengan *dataset* yang terbatas. Pendekatan ini tidak hanya meningkatkan akurasi dalam interpretasi *input* teks, tetapi juga memungkinkan penghematan sumber daya komputasi dan menghasilkan gambar dengan kualitas tinggi yang lebih relevan dengan *input* pengguna (Sutedy dan Qomariyah, 2022).

Dalam penelitian berjudul "*Enhancing Control in Stable Diffusion Through Example-based Fine-tuning and Prompt Engineering*" dilakuakn pendekatan dengan memanfaatkan *DreamBooth* untuk menanamkan pengenal unik pada subjek. Dengan menggunakan *prompt* teks, proses generasi gambar dapat diarahkan ke detail spesifik seperti pose, lingkungan, dan pencahayaan (Rao dan Patel, 2024). Metode ini memungkinkan pembuatan gambar subjek yang sangat disesuaikan dalam berbagai konteks, meskipun elemen-elemen tersebut tidak ada dalam gambar referensi awal yang digunakan untuk pelatihan *DreamBooth*. Keefektifan metode ini ditunjukkan dalam berbagai tugas, termasuk re-kontekstualisasi subjek, sintesis tampilan yang dipandu oleh teks, dan rendering artistik.

Dalam penelitian lain berjudul "*AttnDreamBooth: Towards Text-Aligned Personalized Text-to-Image Generation*" diperkenalkan *AttnDreamBooth*, pendekatan baru yang mengatasi masalah *DreamBooth* yang sering kali dianggap gagal mempertahankan konsep, dengan memisahkan pembelajaran penyelarasan *embedding*, peta perhatian (*attention map*), dan identitas subjek dalam tahapan pelatihan yang berbeda (Pang *dkk.*, 2024). Selain itu, ditambahkan istilah regularisasi peta cross-attention untuk memperkuat proses pembelajaran peta perhatian. Evaluasi menunjukkan bahwa metode ini memberikan peningkatan yang signifikan dalam mempertahankan identitas subjek dan keselarasan dengan teks dibandingkan dengan metode baseline.

Sedangkan untuk mengatasi tantangan efisiensi model difusi, salah satu pendekatan yang dikembangkan adalah *Distribution Matching Distillation* (DMD), yang menawarkan kerangka kerja fleksibel untuk pelatihan generator satu Langkah (Rakitin, Shchekotov dan Vetrov, 2024). Meskipun efektif untuk skenario umum, DMD sebelumnya tidak sepenuhnya disesuaikan untuk masalah penerjemahan gambar-ke-gambar *(image-to-image*, I2I) tanpa pasangan data.

2.2 Dasar Teori

2.2.1 Gambar Karakter Jepang dengan Gaya Retro

Gambar karakter Jepang bergaya retro merujuk pada estetika visual yang populer pada era 1970-an hingga 1990-an dalam industri animasi dan ilustrasi Jepang (Budinugroho dan Islam, 2023). Gaya ini memiliki ciri khas yang mencerminkan perkembangan teknologi gambar dan desain pada masanya serta pengaruh budaya pop yang berkembang saat itu. Gaya retro dalam gambar karakter Jepang memiliki beberapa karakteristik utama yang membedakannya dari gaya modern, antara lain:

- Palet Warna yang Khas – Ilustrasi retro sering menggunakan palet warna yang lebih lembut, dengan dominasi warna pastel atau warna-warna kontras yang lebih sederhana.

- Garis Tebal dan Sederhana – Kontur karakter biasanya digambar dengan garis tebal dan lebih sedikit detail dibandingkan dengan gaya ilustrasi modern yang lebih kompleks.
- Efek Grain dan Noise – Beberapa ilustrasi bergaya retro mengadopsi efek visual seperti grain atau noise untuk meniru tampilan cetakan atau layar CRT (Cathode Ray Tube) dari televisi dan arcade game zaman dulu.
- Proporsi dan Bentuk Mata – Mata karakter dalam gaya retro sering kali lebih kecil dibandingkan dengan gaya modern yang cenderung memiliki mata lebih besar dan detail. Namun, dalam beberapa variasi, mata besar dengan highlight minimal juga ditemukan dalam desain karakter klasik.
- Pose dan Ekspresi Simpel – Karakter dalam ilustrasi retro biasanya memiliki pose yang lebih statis atau sederhana, dengan ekspresi wajah yang lebih minimalis dibandingkan dengan gaya modern yang lebih dinamis.
- Tekstur dan Shading Manual – Banyak ilustrasi retro menggunakan teknik shading dengan gradasi sederhana atau dither untuk menciptakan kesan kedalaman tanpa menggunakan efek digital yang kompleks.

Gambar 2. 1 Contoh Gambar Karakter Jepang Bergaya Retro

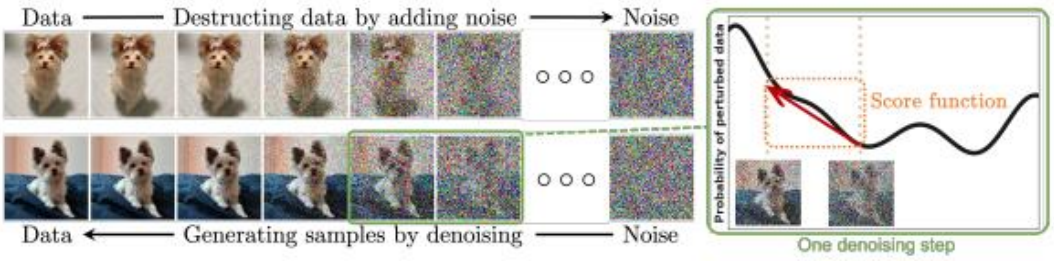
(Source : https://images.app.goo.gl/sT5GFUL2jHjJ1u1k8)

Meskipun gaya ilustrasi terus berkembang, estetika retro tetap memiliki penggemar dan terus digunakan dalam berbagai proyek kreatif. Banyak

proyek anime dan game modern mengadopsi kembali elemen visual retro, seperti UFO Robo Grendizer: The Feast of the Wolves (2023) atau Octopath Traveler II (2023) yang menghadirkan pixel art ala 16-bit. Selain itu, seniman dan desainer indie juga sering mengadaptasi gaya retro untuk menciptakan nuansa nostalgia dalam karya mereka, baik dalam ilustrasi maupun animasi pendek. Kemajuan teknologi kecerdasan buatan juga semakin mendukung tren ini, dengan model seperti *diffusion models* yang kini dapat dilatih untuk menghasilkan gambar karakter berestetika retro secara otomatis, sehingga mempercepat proses produksi dan memperluas eksplorasi kreativitas.

2.2.2 *Diffusion models*

Diffusion models merupakan metode generatif yang memanfaatkan proses stokastik untuk menghasilkan gambar dengan mengubah distribusi noise secara bertahap hingga membentuk pola yang sesuai dengan data latih. Proses ini terdiri dari dua tahap utama yang saling berlawanan, seperti yang ditunjukkan pada Gambar 2.1. Model dilatih untuk mengenali pola dalam data latih dan merekonstruksi gambar dari distribusi noise awal. Dengan pendekatan ini, *diffusion models* mampu menghasilkan gambar yang selaras dengan karakteristik yang diinginkan.



Gambar 2. 2 *Forward Process* dan Backward Process (Yang *dkk.*, 2023)

- *Forward Process* (*Destructing data*): Pada tahap ini, *noise* ditambahkan secara bertahap ke gambar hingga gambar tersebut menjadi tidak dapat dikenali. Proses ini dilakukan melalui serangkaian langkah, di mana setiap langkah menambahkan *noise* Gaussian ke gambar asli, menghasilkan distribusi yang semakin mendekati distribusi *noise*.

- Backward Process (Generating samples): Di tahap ini, model dilatih untuk memulihkan gambar asli dari *noise*. Dengan menggunakan teknik pembelajaran mendalam, model belajar untuk mengurangi *noise* secara bertahap, akhirnya menghasilkan gambar yang bersih. Proses ini mengandalkan representasi ruang laten untuk menghasilkan gambar baru berdasarkan distribusi yang telah dipelajari.

  (Croitoru *dkk.*, 2023)

Model dilatih untuk menghilangkan noise secara bertahap hingga membentuk kembali gambar yang mendekati data latih. Pendekatan ini memungkinkan model merekonstruksi struktur gambar berdasarkan distribusi yang dipelajari selama pelatihan.

2.2.3 Stable Diffusion

Stable Diffusion menggabungkan prinsip dari Denoising Diffusion Probabilistic Models (DDPM) dan NCSN untuk meningkatkan efisiensi dan stabilitas dalam menghasilkan gambar. Seperti DDPM, Stable Diffusion menggunakan proses denoising bertahap, tetapi bekerja di ruang latent, sehingga lebih efisien (Anderson dan Akram, 2024). Selain itu, dengan menggunakan prinsip score-based modeling dari NCSN, Stable Diffusion dapat memandu denoising berdasarkan distribusi data, sehingga lebih cepat dan fleksibel dalam menghasilkan gambar yang sesuai dengan prompt.

DDPM adalah model generatif berbasis proses difusi bekerja dengan menambahkan noise secara bertahap ke dalam data asli hingga menjadi distribusi Gaussian murni, kemudian melatih model untuk membalik proses tersebut guna merekonstruksi kembali data asli dari noise tersebut (Everaert *dkk.*, 2023). Pada tahap Forward Process, DDPM secara bertahap menambahkan noise ke gambar asli dengan rumus:

$$x_t = \sqrt{1-\beta_t}\,.\,x_{t-1} + \sqrt{\beta_t}\,.\,z_t\,, \quad z_t \sim N\,(0,1) \tag{1}$$

β_t adalah parameter yang mengontrol seberapa besar noise yang ditambahkan di tiap langkah t,

x_t adalah hasil dari kombinasi linier antara x_{t-1} dan noise Gaussian z_t

x_{t-1} adalah data pada waktu sebelumnya sebelum noise ditambahkan.

z_t adalah Noise Gaussian acak dengan distribusi normal N (0,I) yang bertanggung jawab untuk membuat gambar semakin tidak dapat dikenali.

Pada tahap Reverse Process, model melatih jaringan saraf untuk memperkirakan distribusi posterior dari data asli menggunakan pendekatan probabilistik, yang didefinisikan dengan:

$$x_{t-1} = \mu_0(x_t, t) + \sqrt{\beta_t}\,.z_t,\ z_t \sim N\,(0,1) \quad (2)$$

$\mu_0(x_t, t)$ adalah prediksi distribusi data sebelumnya berdasarkan parameter,

$\sqrt{\beta_t}$ adalah faktor noise yang dikurangi dalam langkah denoising.

NCSN adalah model generatif berbasis score-based generative modeling, yang bertujuan untuk mempelajari gradien logaritma dari distribusi data (dikenal sebagai score function), alih-alih langsung memodelkan distribusinya. Model ini menggunakan Annealed Langevin Dynamics untuk menghasilkan sampel gambar dengan cara memperkirakan gradien dari distribusi data dan melakukan iterasi rekonstruksi secara bertahap (Jung *dkk.*, 2024). Pada tahap Forward Process, NCSN menambahkan noise ke data asli dengan cara yang mirip DDPM:

$$x_t = x_{t-1} + \sqrt{\sigma_t^2 - \sigma_{t-1}^2}\,.z_t,\ z_t \sim N\,(0,1) \quad (3)$$

σ_t^2 adalah skala noise pada waktu, yang meningkat seiring waktu

σ_{t-1}^2 adalah skala noise pada waktu sebelumnya.

Namun, berbeda dari DDPM, pada tahap Reverse Process, NCSN menggunakan metode berbasis score function, dengan prinsip:

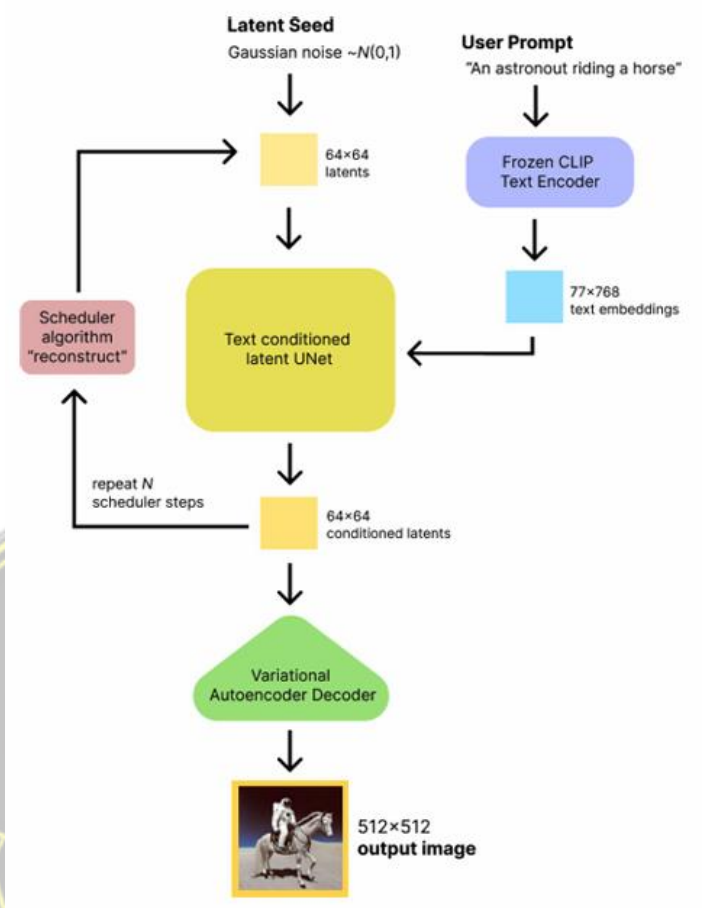
$$\partial x = \left|\dot{f}(x,t) - \sigma(t)^2\,\nabla_x log p_t(x)\right| dt + \sigma(t)\,d\bar{\omega} \quad (4)$$

$\nabla_x log p_t(x)$ adalah *score function*, yang merupakan gradien dari log-*likelihood* data pada waktu

$d\bar{\omega}$ adalah *Brownian motion* pada proses *denoising*, yang dapat mengoreksi noise agar kembali ke bentuk aslinya.

Karena menggunakan pendekatan berbasis *score function*, NCSN dapat memperbaiki gambar dengan cara yang lebih fleksibel dibandingkan DDPM, namun sering kali kurang stabil dibandingkan DDPM. Gabungan dari dua

pendekatan ini menjadikan Stable Diffusion lebih efisien, cepat, dan stabil dibandingkan model difusi lainnya.



Gambar 2. 3 Ilustrasi Cara Kerja *Stable Diffusion*

Berikut adalah cara kerja dari Stable Diffusion sebagaimana yang diilustrasikan pada gambar 2.3 :

- Input Prompt dari Pengguna. Pengguna memasukkan teks prompt, seperti "An astronaut riding a horse", sebagai panduan untuk menghasilkan gambar.
- Pemrosesan dengan CLIP Text Encoder. Prompt diproses menggunakan Frozen CLIP Text Encoder, mengubahnya menjadi vektor embedding berukuran 77×768 sebagai referensi model.
- Inisialisasi Gaussian Noise dalam Ruang Laten. Model menginisialisasi latent seed berupa Gaussian noise berdistribusi $N(0,1)$ dalam ruang laten berukuran 64×64.

- Proses Denoising dengan Text-Conditioned Latent U-Net. Latent noise dan embedding teks diproses oleh Text-Conditioned Latent U-Net, yang melakukan denoising secara bertahap melalui beberapa iterasi menggunakan scheduler algorithm.
- Konversi Laten ke Gambar dengan VAE Decoder. 64×64 conditioned latents diterjemahkan ke gambar beresolusi lebih tinggi menggunakan Variational Autoencoder (VAE) Decoder.
- Output Gambar. Model menghasilkan gambar 512×512 yang sesuai dengan teks prompt pengguna.

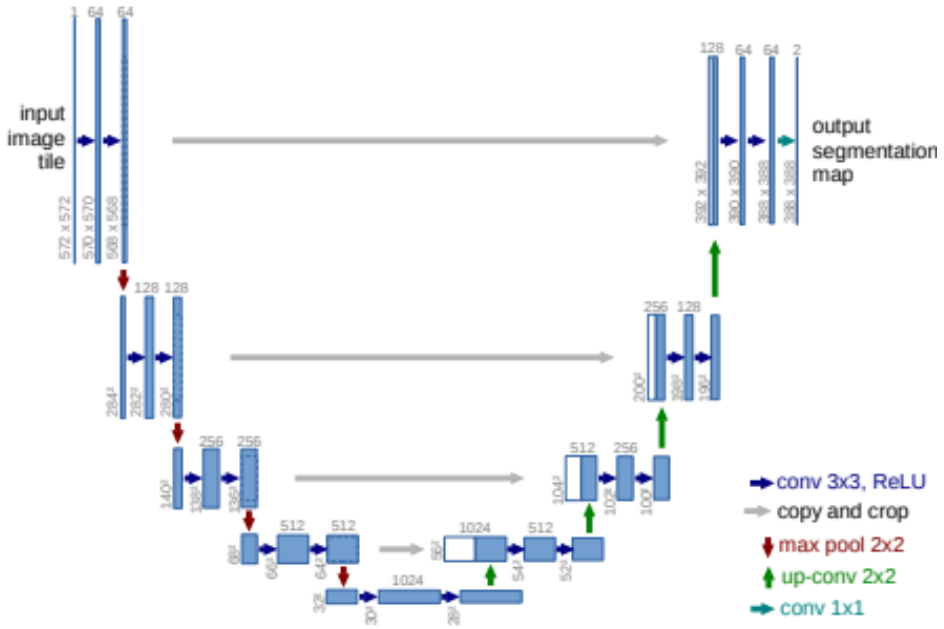
Arsitektur yang ditampilkan dalam gambar adalah Stable Diffusion atau *Latent Diffusion Model (LDM)*, yang digunakan untuk menghasilkan gambar dari teks. Berikut adalah penjelasan komponennya:

- *Latent Seed*: Model memulai proses dengan menghasilkan latar latar acak menggunakan distribusi Gaussian $N(0,1)$yang kemudian digunakan sebagai dasar untuk membentuk gambar.
- *Frozen CLIP Text Encoder*: Teks dari pengguna dikodekan menjadi vektor embedding berukuran 77 × 768 menggunakan model CLIP, yang berfungsi untuk memahami dan menerjemahkan makna teks ke dalam fitur numerik.
- *Text Conditioned Latent UNet*: *UNet* yang dikondisikan dengan teks mengambil latar awal dan melakukan serangkaian langkah denoising untuk mengubahnya menjadi latar terstruktur yang lebih mendekati bentuk gambar yang diinginkan.
- *Scheduler Algorithm* ("*Reconstruct*"): Algoritma ini bertugas mengontrol proses rekonstruksi gambar dengan menjalankan beberapa langkah iteratif untuk menyempurnakan hasil.
- *Variational Autoencoder (VAE) Decoder*: Setelah latar akhir diperoleh, *variational autoencoder* mendekode latar 64 × 64 menjadi gambar resolusi tinggi 512 × 512.

  (Prasad *dkk.*, 2024)

2.2.4 U-Net

U-Net terdiri dari dua bagian utama: bagian kontraksi (*encoder*) dan bagian ekspansi (*decoder*). Bagian kontraksi bertugas untuk mengekstraksi fitur dengan melakukan serangkaian operasi konvolusi yang diikuti oleh pooling, yang mengurangi dimensi data sambil mempertahankan informasi penting. Sebaliknya, bagian ekspansi bertujuan untuk menghasilkan *output* melalui teknik up*sampling* dan konvolusi transposisi, yang menggabungkan fitur dari bagian kontraksi. Keunggulan U-Net terletak pada penggunaan skip *connections* yang memungkinkan transfer informasi dari layer yang lebih dalam ke layer yang lebih dangkal, sehingga meningkatkan akurasi dan detail pada *output* akhir (Siddique *dkk.*, 2021).



Gambar 2. 4 Arsitektur U-Net (Weng dan Zhu, 2021)

Keterangan di bawah gambar 2.4 memberikan informasi tentang berbagai operasi yang digunakan dalam arsitektur jaringan saraf konvolusional (CNN) yang ditampilkan pada diagram, yang menyerupai U-Net**,** model yang umum digunakan untuk segmentasi citra.

1. conv 3x3, ReLU. Konvolusi 3x3 dengan aktivasi ReLU untuk ekstraksi fitur.

2. copy and crop. Penggabungan fitur dari encoder ke decoder untuk mempertahankan detail.
3. max pool 2x2. Pooling 2x2 untuk mengurangi dimensi dan menangkap fitur global.
4. up-conv 2x2. Upsampling 2x2 untuk meningkatkan resolusi gambar.
5. conv 1x1. Konvolusi 1x1 untuk menyesuaikan jumlah channel output.

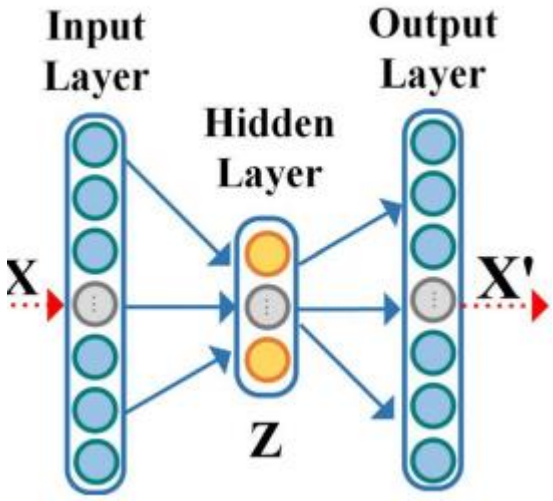
(Weng dan Zhu, 2021)

Arstitektur U-Net awalnya dirancang untuk segmentasi citra, memiliki struktur yang dapat menangkap fitur spasial dari gambar, yang sangat berguna dalam konteks difusi. Dalam penggunaan U-Net untuk *Stable Diffusion*, model ini beroperasi berdasarkan prinsip, di mana dilatih untuk mempelajari representasi dari data gambar mentah. Fungsi *loss* yang digunakan, seperti mean squared error, bertujuan untuk meminimalkan perbedaan antara gambar yang dihasilkan dan gambar target.

Penggunaan U-Net dalam *Stable Diffusion* mengintegrasikan arsitektur yang kuat untuk menghasilkan gambar berkualitas tinggi melalui proses difusi. Dengan kemampuan untuk menangkap dan memanfaatkan fitur spasial, U-Net memberikan kontribusi signifikan dalam mencapai stabilitas dan kualitas dalam generasi gambar, menjadikannya alat yang berharga dalam domain pemrosesan gambar dan pembelajaran mesin.

2.2.5 Autoencoders (AEs)

Autoencoder adalah salah satu model utama yang membentuk dasar dari berbagai alat generatif dalam Artificial Intelligence (AI), termasuk aplikasi seperti Stable Diffusion dan variational autoencoders (VAEs). Autoencoder dirancang sebagai model jaringan saraf yang bertujuan untuk merekonstruksi inputnya melalui proses enkoding dan dekoding, sehingga dapat merepresentasikan data dalam bentuk laten yang lebih terkompresi (Bengesi *dkk.*, 2024). Autoencoder terdiri dari dua komponen utama, seperti yang ditunjukkan pada gamabr 2.4.



Gambar 2. 5Arsitektur Auto-encoder (Berahmand *dkk.*, 2024)

- Encoder (Input): Bagian ini menerima input dan mengubahnya menjadi representasi laten dengan dimensi yang lebih kecil. Encoder memproses data menjadi bentuk yang lebih padat dan kompresi, dengan tujuan untuk menangkap informasi yang paling relevan dari data tersebut.
- Decoder (*Output*): Setelah data dimampatkan, decoder bertugas untuk mengubah representasi laten kembali ke ruang asli. Tujuannya adalah untuk menghasilkan *output* yang mirip dengan input asli.

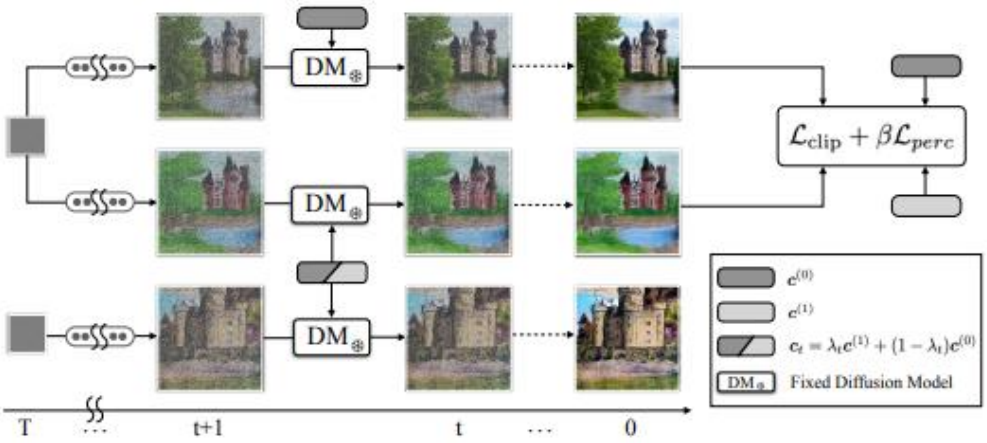
Autoencoder, khususnya Variational Autoencoder (VAE), memainkan peran penting dalam banyak model generatif seperti generasi gambar dan pembuatan music (Xiang *dkk.*, 2023). Misalnya, dalam model seperti Stable Diffusion dan DALL-E**,** autoencoder digunakan untuk memproses data dalam ruang laten yang memungkinkan transformasi data masukan menjadi *output* yang lebih kompleks dan realistis.

2.2.6 Text-to-Image Generation

Dalam konteks *diffusion models* dan *Stable Diffusion*, text-to-image generation i berfungsi sebagai penghubung antara input tekstual dan *output* visual. Proses ini melibatkan beberapa langkah:

- *Embed Text* : Deskripsi teks yang diberikan diubah menjadi representasi vektor menggunakan model bahasa (seperti CLIP atau BERT), yang mampu menangkap makna dan konteks dari teks.
- *Conditioning* : Representasi teks digunakan sebagai kondisi dalam proses pembangkitan gambar. Model diffusion yang terlatih dapat menghasilkan gambar karakter Jepang yang sesuai dengan deskripsi teks yang diberikan, menangkap elemen-elemen kunci dari karakter yang diinginkan.

  (Parmar *dkk.*, 2024)



Gambar 2. 6 Penggunakan Embedding Text untuk Mengubah Atribut Gambar (Wu *dkk.*, 2023)

Rumus interpolasi kondisi :

$$c_t = \lambda_t c^{(1)} + (1 - \lambda_t) c^{(0)} \quad (5)$$

c_t adalah kondisi pada awal langkah *t.*

λ_t adalah koefisien interpolasi antara dua kondisi.

$c^{(0)}$ adalah kondisi awal.

$c^{(1)}$ adalah kondisi target.

Rumus ini berguna dalam transfer gaya gambar, pengeditan berbasis teks, serta kontrol atribut dalam generasi gambar, di mana model dapat menyesuaikan seberapa besar pengaruh kondisi awal dan kondisi target dalam setiap langkah difusi.

Fungsi loss :

$$\mathcal{L} = \mathcal{L}_{clip} + \beta\mathcal{L}_{perc} \quad (6)$$

$\mathcal{L}$ adalah total loss yang digunakan untuk melatih model difusi.

$\mathcal{L}_{clip}$ adalah loss berbasis CLIP yang mengontrol keselarasan antara gambar dan kondisi target.

$\mathcal{L}_{perc}$ adalah loss perseptual yang mempertahankan detail visual berdasarkan fitur gambar.

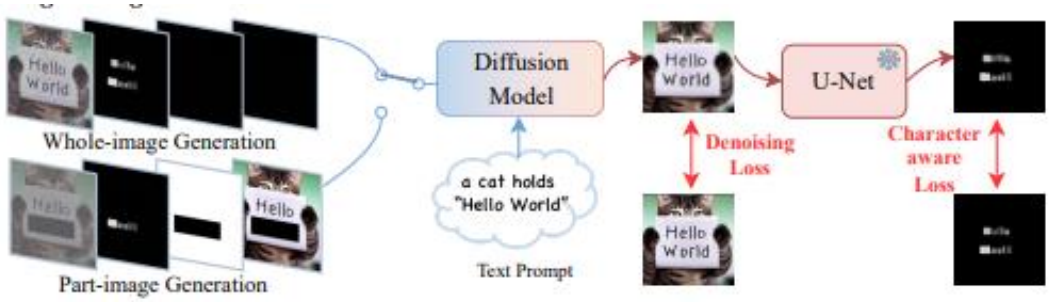
β adalah faktor skalar yang mengontrol kontribusi $\mathcal{L}_{perc}$

Rumus ini digunakan untuk mengoptimalkan hasil keluaran model difusi dengan mempertahankan keselarasan terhadap kondisi yang diinginkan.

Setelah embedding teks, model diffusion akan menggunakan representasi vektor tersebut untuk memandu proses pembuatan gambar. Proses conditioning memungkinkan model untuk memahami hubungan antara elemen-elemen teks dan visual, sehingga gambar yang dihasilkan bisa mencerminkan atribut-atribut yang diinginkan, seperti gaya, bentuk, dan detail lainnya. Dalam konteks karakter Jepang, misalnya, model akan menghasilkan gambar dengan ciri khas visual yang mencakup elemen budaya atau estetika tertentu yang ada dalam teks, seperti warna pakaian, bentuk rambut, atau latar belakang yang sesuai dengan deskripsi. Dengan demikian, text-to-image generation memungkinkan penciptaan gambar yang lebih dinamis dan tepat sesuai dengan input teks yang diberikan.

2.2.7 Tranformers

Library transformers bertanggung jawab untuk memproses deskripsi teks dan mentransformasikannya menjadi representasi vektor. Representasi ini kemudian digunakan untuk membimbing proses generasi gambar dengan model difusi. Perpustakaan ini menggabungkan kemudahan penggunaan dengan fleksibilitas, sehingga mendukung berbagai tugas NLP, seperti klasifikasi teks, analisis sentimen, penerjemahan, dan generasi teks (Prasad *dkk.*, 2024).



Gambar 2. 7 Transformer pada *Diffusion models* (Chen *dkk.*, 2024)

Seperti yang ditunjukkan pada gambar 2.6, dalam stable diffusion transformer digunakan untuk menghasilkan *layout* kata kunci yang diambil dari deskripsi teks (*text prompt*). Transformer menganalisis *text prompt* dan mengekstrak kata kunci utama, yang kemudian digunakan untuk membangun *layout* awal. Dengan mekanisme *self-attention*, transformer mampu memahami konteks kata dalam kalimat, memastikan bahwa kata kunci yang dipilih relevan dan sesuai dengan tujuan deskripsi. *Output* dari transformer adalah representasi tata letak teks yang memuat posisi, ukuran, dan orientasi teks dalam gambar (Chen *dkk.*, 2024). Representasi ini menjadi landasan bagi model difusi untuk menghasilkan gambar yang mengintegrasikan teks dengan latar belakang secara harmonis.

2.2.8 Fine-tuning

Gambar 2. 8 Gambar yang Dihasilkan oleh *Stable Diffusion* Setelah Fine Tuning (Yang *dkk.*, 2023)

Fine-tuning adalah proses penyesuaian model yang telah dilatih sebelumnya untuk meningkatkan performanya dalam konteks spesifik seperti

yang dicontohkan pada gambar 2.7. Dalam pembangunan generator gambar karakter Jepang, *fine-tuning* dilakukan dengan cara berikut:

- *Dataset* Khusus : Model yang telah dilatih pada *dataset* umum dapat di-fine-tune menggunakan *dataset* yang lebih spesifik mengenai karakter Jepang. Hal ini memungkinkan model untuk lebih memahami gaya, warna, dan elemen visual yang khas dari karakter Jepang.
- *Transfer Learning* : Dengan memanfaatkan model yang sudah ada, *fine-tuning* mengurangi waktu dan sumber daya yang dibutuhkan untuk melatih model baru dari awal. Teknik ini memungkinkan penyesuaian yang lebih cepat dan efektif, menghasilkan gambar yang lebih relevan dengan tema yang diinginkan.

  (Shi, 2024)

Untuk mengarahkan *fine-tuning* agar menghasilkan kesan retro, dataset yang digunakan bisa berfokus pada gambar yang mengusung estetika retro—misalnya, gaya desain dari dekade tertentu seperti tahun 1980-an, dengan perhatian pada elemen-elemen visual seperti warna, garis-garis geometris, tekstur, dan elemen desain vintage. *Fine-tuning* ini memungkinkan model untuk menangkap karakteristik retro. Dengan *fine-tuning* yang terfokus pada gaya retro, model akan lebih mudah menghasilkan gambar karakter Jepang yang bukan hanya sesuai dengan budaya visual Jepang, tetapi juga mengandung sentuhan estetik retro yang kuat, memberikan kesan nostalgia.

2.2.9 Dreambooth Fine Tuning

Model difusi adalah jenis model generatif yang dilatih untuk mempelajari distribusi data dengan menambahkan *noise* pada data asli dan kemudian belajar untuk merekonstruksi data tersebut. Pada DreamBooth, model ini digunakan sebagai basis, dan dilatih ulang secara selektif menggunakan gambar tertentu untuk mengenali objek atau gaya baru (Sutedy dan Qomariyah, 2022). DreamBooth menggunakan proses *fine-tuning*, yaitu melatih ulang model yang sudah ada menggunakan data baru dengan jumlah terbatas. *Fine-tuning* ini menjaga kemampuan asli model sambil menambah kemampuan baru,.

Gambar 2. 9 Perbandingan Dreambooth dengan *Textual Inversion* (Ruiz *dkk.*, 2023)

Gambar 2.9 menunjukkan perbandingan penggunaan Dreambooth dengan *Textual Inversion. DreamBooth* lebih cocok untuk personalisasi objek atau karakter baru dengan pelatihan mendalam yang menyesuaikan bobot model, sementara *Textual Inversion* lebih ringan dan fleksibel karena hanya menyimpan konsep sebagai token teks dalam ruang laten. *DreamBooth* memberikan hasil yang lebih presisi untuk satu konsep tertentu, tetapi jika ingin menambahkan konsep tanpa mengubah bobot model, *Textual Inversion* adalah pilihan yang lebih efisien.

Selain itu, DreamBooth memungkinkan model untuk lebih efektif mengadaptasi dan menghasilkan gambar berdasarkan objek atau konsep baru dengan kualitas yang tetap terjaga, meskipun hanya menggunakan data

terbatas. Proses *fine-tuning* pada DreamBooth tidak hanya meningkatkan kemampuan model dalam mengenali dan merekonstruksi gambar tertentu, tetapi juga memastikan bahwa model tetap dapat menghasilkan gambar yang lebih fleksibel dalam berbagai gaya atau konteks yang ditentukan. Dengan penggunaan Regularization *loss*, model dapat menghindari penurunan kualitas yang signifikan yang biasanya terjadi saat melakukan *fine-tuning* pada dataset kecil. Hal ini membuat DreamBooth menjadi alat yang efektif dalam menciptakan gambar yang lebih spesifik dan sesuai dengan kebutuhan visual tertentu, seperti gambar karakter dalam gaya atau tema tertentu.

2.2.10 Regularisasi L2

Stable Diffusion memiliki resiko untuk mengalami *overfitting* selama proses pelatihan, terutama jika model menjadi terlalu bergantung pada pola spesifik dalam data pelatihan. Hal ini dapat menyebabkan generalisasi yang buruk, di mana model menghasilkan gambar yang tidak cukup bervariasi atau tidak sesuai dengan input teks yang diberikan. Sedangkan Regularisasi L2, juga dikenal sebagai *penalti ridge*, adalah teknik regulasi yang menambahkan penalti berbasis kuadrat bobot ke fungsi *loss* model (Hutagalung, 2024).

$$Loss_{L2} = Loss_{original} + \lambda \sum_{i=1}^{n} w_i^2 \quad (7)$$

$Loss_{original}$: Nilai *loss* awal (misalnya, MSE atau cross-entropy) tanpa regularisasi.

λ : Koefisien regularisasi (hyperparameter) yang menentukan seberapa besar kontribusi regularisasi.

w_i^2 : Parameter model (bobot) yang sedang dioptimasi.

n : Jumlah total parameter (bobot) dalam model.

Dalam kasus *Stable Diffusion*, regulasi L2 dapat diterapkan pada berbagai tahap pelatihan model untuk mengurangi risiko *overfitting* dengan cara :

- Pada *Encoder Text-to-Image*: Mencegah model terlalu menghafal pola dalam representasi teks, sehingga menghasilkan gambar yang lebih sesuai dengan konteks prompt.

- Pada UNet dalam Proses Difusi: UNet menangani manipulasi noise dalam proses difusi. Regularisasi L2 dapat membantu menjaga bobot UNet tetap terkendali, memastikan proses rekonstruksi noise tetap stabil dan tidak bias terhadap pola tertentu.
- Pada *Decode*r: Pada tahap *decoding*, L2 dapat membantu menghasilkan visual yang lebih bervariasi dan tidak terlalu mirip dengan dataset pelatihan.

  (Rakitin, Shchekotov dan Vetrov, 2024)

Dalam penelitian ini, regularisasi L2 diterapkan pada CLIP encoder untuk mencegah *overfitting*, karena CLIP encoder berperan penting dalam menghubungkan teks dengan representasi visual (Li *dkk.*, 2023). Secara umum Clip Score dihitung dengan menggunakan prinsip *cosine similiarity* antara embedding teks dengan embedding gambar dari model yang telah di *finetune* dengan rumus :

$$S = \frac{E_t \cdot E_i}{\|E_t\| \cdot \|E_i\|} \quad (8)$$

S adalah Clip Score, nilai kesesuaian antara vteks dan gambar,

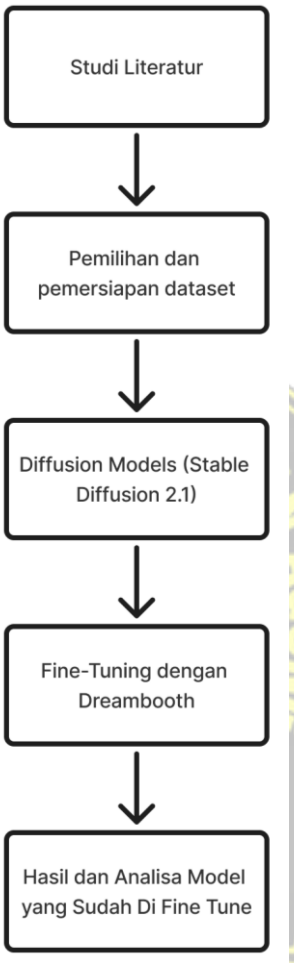
E_t adalah vector embedding dari teks (prompt input)

E_i adalah vector embedding dari gambar hasil dari model yang sudah di finetune

Dengan menambahkan penalti pada bobot encoder, model diharapkan dapat menghasilkan representasi teks yang lebih umum dan tidak terlalu terikat pada pola spesifik yang ada dalam dataset pelatihan. Hal ini memastikan bahwa model dapat menggenerasikan gambar yang lebih sesuai dengan variasi teks yang lebih luas, serta meningkatkan kemampuan generalisasi model terhadap input yang belum terlihat sebelumnya.

BAB III
METODE PENELITIAN

3.1 Metode Penelitian



Gambar 3. 1 Langkah-langkah Penelitian

Penelitian ini mengandalkan model diffusion, yaitu Stable Diffusion v2.1, sebagai inti dari proses generasi gambar. Stable Diffusion v2.1 dipilih karena kemampuannya untuk menghasilkan gambar berkualitas tinggi dengan berbagai gaya visual, termasuk karakter Jepang. Model ini memanfaatkan proses difusi untuk membangun gambar secara bertahap dari noise hingga menghasilkan representasi yang sangat mendetail dan realistis. Dengan memanfaatkan representasi ruang laten, model ini dapat menghasilkan

gambar yang mencakup berbagai elemen penting, seperti proporsi tubuh, detail wajah, dan gaya visual karakter Jepang.

Untuk meningkatkan kemampuan model dalam menghasilkan gambar yang lebih spesifik dan sesuai dengan tema karakter Jepang, penelitian ini menerapkan teknik *fine-tuning* menggunakan Dreambooth. Dreambooth memungkinkan penyesuaian model untuk mengenali elemen-elemen khusus, seperti atribut fisik, pakaian, atau latar belakang yang menjadi ciri khas dalam karakter Jepang. Proses *fine-tuning* ini dilakukan dengan menggunakan dataset yang lebih fokus pada karakter Jepang, sehingga model dapat menghasilkan gambar dengan kualitas yang lebih baik dan kesesuaian yang lebih tinggi terhadap gaya visual yang diinginkan. Kualitas gambar dievaluasi berdasarkan kriteria seperti ketajaman detail visual, kesesuaian dengan gaya karakter Jepang, serta konsistensi dalam pose dan atribut visual lainnya.

3.1.1 Studi Literatur

Meninjau dan menganalisis topik seputar *Stable Diffusion*, *Fine-tuning*, Dreambooth, dan *Regularization* L2 dengan sumber dari penelitian terdahulu.

3.1.2 Pengumpulan dan Pengolahan *Dataset*

Penelitian ini dimulai dengan pemilihan sumber data yang dapat diandalkan dari sumber *daring*, seperti platform-platform gambar terbuka dan repositori online, untuk memperoleh gambar karakter Jepang yang relevan dengan tujuan penelitian. Gambar-gambar yang diperoleh akan diseleksi dengan memperhatikan kesesuaian ekspresi, gaya, dan atribut visual khas pada karakter Jepang, sehingga memastikan data yang digunakan mendukung tujuan pengembangan model generatif yang lebih akurat dan sesuai. Setelah proses pengumpulan, data akan diproses dengan langkah-langkah berikut:

- Pengelompokan Gambar: Gambar-gambar yang sudah dipilih akan dikelompokkan berdasarkan atribut seperti pose, gaya, dan elemen visual karakter untuk memudahkan penyesuaian saat proses pelatihan.
- Validasi Keberagaman dan Penyesuaian: Validasi dilakukan untuk memastikan dataset mencakup variasi yang cukup, seperti perbedaan pose,

atribut, dan gaya visual, sehingga model dapat mempelajari variasi karakter dengan baik.

- Penyimpanan Dataset: Dataset yang telah diproses akan dibagi menjadi dua kategori utama: Data Pelatihan dan Data Uji. Data pelatihan harus mencakup berbagai contoh karakter dengan pose dan atribut visual yang beragam agar model dapat belajar secara menyeluruh.

3.1.3 Stable Diffusion 2.1

Stable Diffusion 2.1 dipilih dalam penelitian ini karena kemampuan generatifnya yang lebih baik dibandingkan versi sebelumnya, khususnya dalam menghasilkan gambar berkualitas tinggi dengan detail yang lebih jelas dan pengendalian *prompt* yang lebih baik. Namun, meskipun memiliki banyak keunggulan, model ini tetap menghadapi tantangan dalam dua aspek utama:

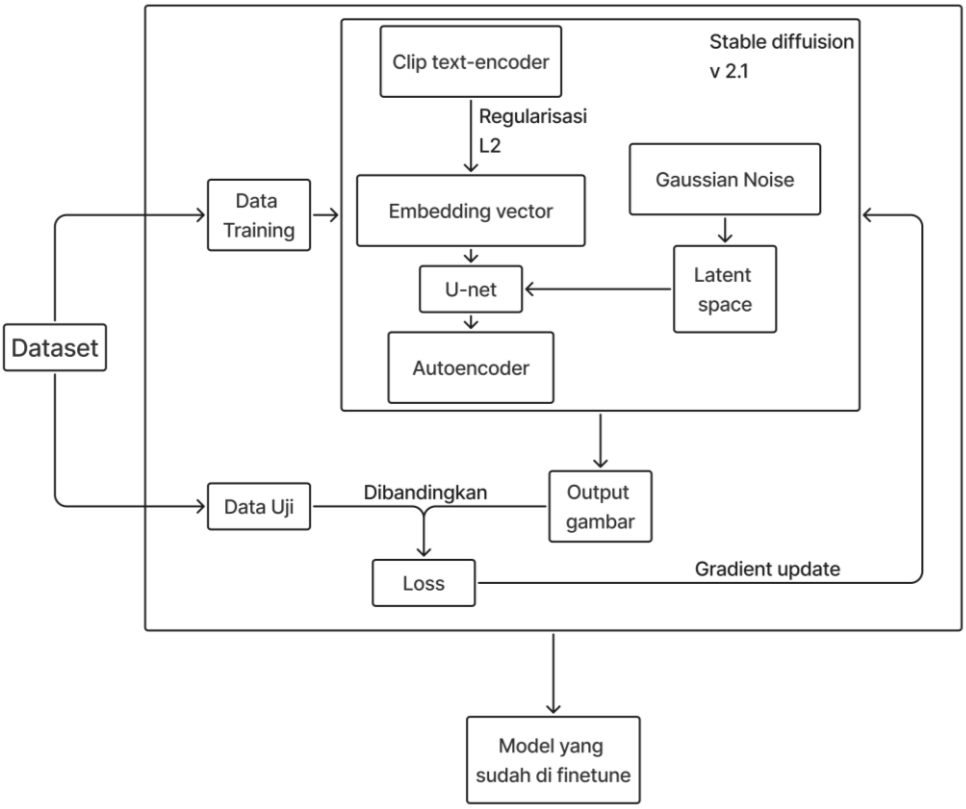
- Inkonsistensi *Output*

  Model ini terkadang menghasilkan variasi yang tidak konsisten saat diminta untuk menggambarkan karakter dengan atribut visual yang spesifik. Misalnya, dalam menggambarkan karakter Jepang bergaya retro dengan elemen seperti pola pakaian khas, ekspresi wajah tertentu, atau detail dekoratif, hasil yang dihasilkan bisa berbeda-beda meskipun menggunakan *prompt* yang sama. Hal ini disebabkan oleh pelatihan model pada dataset umum tanpa fokus khusus pada gaya visual tertentu, sehingga atribut yang lebih kompleks sering kali diabaikan.

- *Overfitting*

  Saat dilakukan adaptasi dengan dataset kecil menggunakan metode seperti DreamBooth, model ini dapat mengalami *overfitting*. Hal ini menyebabkan kemampuan generalisasi model menurun, sehingga hanya mampu menghasilkan gambar yang sangat mirip dengan data pelatihan dan kehilangan variasi yang diharapkan. Tantangan ini menjadi perhatian khusus dalam menghasilkan karakter dengan gaya unik seperti Jepang retro, yang membutuhkan keseimbangan antara kesesuaian gaya dan fleksibilitas visual.

3.1.4 Training Model dengan Dreambooth



Gambar 3. 2 *Workflow Training* Sistem

Alur kerja yang digambarkan dalam diagram tersebut dapat dijelaskan dalam bentuk poin per poin sebagai berikut:

1. Proses dimulai dengan menerima Keyword (kata kunci) atau deskripsi teks yang digunakan sebagai input. Kata kunci yang diberikan akan menjadi *prompt* teks, yang menggambarkan apa yang akan dihasilkan oleh model.
2. CLIP Text Encoder

   Kata kunci atau *prompt* diproses melalui CLIP Text Encoder, yang mengonversi teks menjadi embedding vector, yaitu representasi vektor yang menggambarkan makna dan konteks dari teks tersebut. Representasi ini kemudian digunakan sebagai acuan visual dalam proses generasi gambar, memungkinkan model untuk menghasilkan gambar yang sesuai dengan deskripsi tekstual yang diberikan.

3. Selanjutnya, representasi vektor ini digunakan dalam arsitektur Stable Diffusion v2.1, yang berisi beberapa komponen:
   - Gaussian Noise: Ditambahkan secara bertahap pada gambar dalam ruang laten untuk mengacak struktur gambar asli. Proses ini menghasilkan gambar yang semakin acak seiring bertambahnya iterasi noise, yang kemudian disiapkan untuk tahap denoising.
   - U-Net: Arsitektur U-Net digunakan untuk melakukan denoising, yaitu mengurangi noise yang ditambahkan dalam proses sebelumnya. Dengan arsitektur berbentuk encoder-decoder, U-Net dapat menangkap informasi kontekstual dan menghasilkan gambar yang lebih bersih.
   - Regularisasi L2: Diterapkan pada model untuk mencegah *overfitting*. Dengan menambahkan penalti pada bobot yang terlalu besar, teknik ini memastikan model tetap dapat menggeneralisasi dengan baik terhadap data yang belum terlihat.
   - Latent Space: Dalam proses diffusion, gambar terlebih dahulu diubah menjadi representasi laten. Ruang laten ini digunakan untuk memanipulasi gambar tanpa mengorbankan kualitas detail, sebelum diproses lebih lanjut untuk menghasilkan *output* visual.
   - Autoencoder: Setelah noise dihilangkan, autoencoder digunakan untuk mengubah representasi laten menjadi gambar yang dapat dipahami. Autoencoder ini memungkinkan model untuk mengembalikan gambar yang lebih realistis dari ruang laten yang lebih terkompresi.
4. *Dataset*

   Dataset digunakan untuk melatih model dengan data yang relevan, seperti gambar dan teks yang sesuai dengan karakter yang ingin digambarkan. Data ini memberikan variasi yang cukup untuk model memahami berbagai elemen visual dan kontekstual yang diperlukan.
   - Data *Training.* Data pelatihan diproses melalui model untuk mengoptimalkan bobot dan meningkatkan kemampuan model dalam menghasilkan gambar sesuai dengan deskripsi teks. Proses ini

memungkinkan model belajar mengenali hubungan antara teks dan visual dalam konteks karakter yang diinginkan.

- Data Uji. Setelah tahap pelatihan selesai, data uji digunakan untuk menguji hasil model dengan menggunakan *prompt* yang berbeda. Data uji berfungsi untuk mengevaluasi akurasi dan efektivitas model dalam menghasilkan gambar yang sesuai dengan permintaan teks yang baru dan bervariasi.

Dalam penelitian ini, gambar yang digunakan untuk pelatihan juga digunakan dalam tahap pengujian, meskipun tanpa adanya deskripsi teks yang menyertainya. Pada tahap pelatihan, gambar-gambar ini diproses untuk mengoptimalkan model dan membiasakan model dalam menghasilkan gambar berdasarkan representasi laten. Ketika memasuki tahap pengujian, gambar yang sama digunakan untuk mengevaluasi kualitas *output* model dalam menghasilkan gambar yang sesuai dengan ekspektasi, meskipun tanpa tambahan teks sebagai acuan.

5. *Loss Function*

Hasil gambar yang dihasilkan oleh model kemudian dibandingkan dengan gambar referensi untuk memverifikasi kesesuaian dan kualitas *output* sebagai bagian dari proses evaluasi menggunakan *loss* function. *Loss* function ini dihitung untuk mengukur kesalahan atau perbedaan antara gambar yang dihasilkan dengan gambar referensi, memberikan indikasi sejauh mana model berhasil menghasilkan gambar yang sesuai dengan tujuan yang diinginkan. Proses ini digunakan untuk mengevaluasi kinerja model secara objektif dan untuk memandu pembaruan bobot, memastikan bahwa model terus meningkatkan akurasi dan menghasilkan gambar yang lebih relevan dan berkualitas.

6. *Gradient Update*

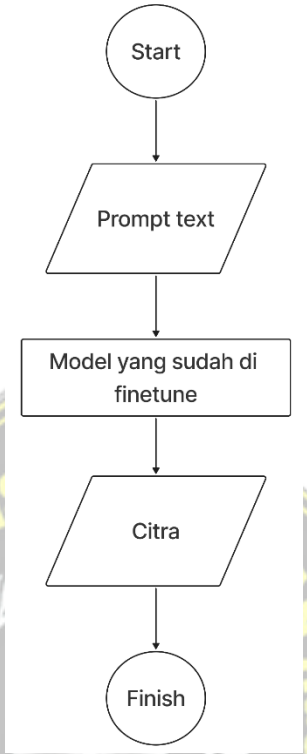
Berdasarkan perhitungan *loss*, dilakukan gradient update untuk memperbarui bobot model, yang bertujuan meningkatkan akurasi dalam menghasilkan gambar yang sesuai dengan deskripsi atau referensi yang diinginkan. Proses ini melibatkan perhitungan gradien dari *loss* function

terhadap bobot model, kemudian bobot diperbarui melalui algoritma optimasi seperti *stochastic gradient descent* (SGD). Pembaruan bobot ini memungkinkan model untuk belajar dari kesalahan yang ada dan mengoptimalkan kemampuannya dalam menghasilkan gambar yang lebih baik, memperkecil perbedaan antara *output* dan gambar referensi seiring berjalannya waktu.

7. Model yang sudah *di fine tune*

   Setelah pembaruan gradien, model yang telah diperbarui akan menghasilkan representasi yang lebih akurat dan lebih sesuai dengan *prompt* yang diberikan. Proses ini memungkinkan model untuk meminimalkan kesalahan pada iterasi sebelumnya, menghasilkan gambar yang lebih baik dalam hal kesesuaian visual dan detail sesuai dengan yang diinginkan. Dengan model yang telah disesuaikan melalui update gradien, model siap digunakan untuk menghasilkan gambar baru yang lebih relevan dan berkualitas tinggi berdasarkan input atau deskripsi teks yang diberikan.

3.2 Perancangan Alur Sistem



Gambar 3. 3 Alur Kerja Sistem

1. *Input* Deskripsi Teks. Pengguna memberikan deskripsi teks yang menggambarkan karakter yang ingin dihasilkan, termasuk atribut visual. Deskripsi ini menjadi panduan awal bagi sistem untuk memahami dan membentuk karakter sesuai dengan keinginan pengguna.

   Input deskripsi teks yang mencakup elemen-elemen seperti jenis kelamin (cowok/cewek), usia (tua/muda), aksesoris yang dipakai (misalnya kaca mata), kegiatan yang dilakukan (seperti makan atau naik sepeda), serta latar belakang, akan diproses oleh model untuk menghasilkan gambar yang sesuai dengan setiap detail tersebut.
2. *Encoding* Teks. Deskripsi teks yang diberikan pengguna diproses menggunakan CLIP *Text Encoder* untuk mengubahnya menjadi representasi vektor (*text embedding*). Vektor ini menyimpan informasi esensial mengenai atribut karakter yang diminta dan digunakan untuk

membimbing model generatif dalam menghasilkan gambar yang akurat dan sesuai dengan deskripsi.

3. Generasi Gambar. Setelah model di-fine-tune, proses generasi gambar dimulai dengan *Diffusion models*. Proses ini dimulai dengan noise acak, yang kemudian secara bertahap diubah menjadi gambar sesuai dengan deskripsi teks. Proses denoising ini menggunakan embedding teks untuk membimbing model, sehingga gambar yang dihasilkan semakin mendekati representasi visual yang diminta pengguna.
4. Evaluasi dan Validasi. Setelah gambar dihasilkan, evaluasi dilakukan untuk memeriksa keselarasan visual antara gambar yang dihasilkan dengan deskripsi yang diberikan. Proses ini memastikan bahwa atribut seperti ekspresi wajah, gaya pakaian, dan elemen desain retro telah tercermin dengan tepat. Selain itu, konsistensi gaya visual juga diperiksa untuk memastikan gambar tidak keluar dari batasan gaya yang diminta.
5. *Output* Gambar. Setelah evaluasi selesai, gambar akhir yang dihasilkan disediakan untuk diunduh oleh pengguna. Gambar tersebut adalah hasil dari proses generasi yang telah dievaluasi untuk kesesuaian dan konsistensi, sehingga siap digunakan oleh pengguna sesuai dengan kebutuhan atau preferensinya.

3.3 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem bertujuan untuk memastikan bahwa sistem yang dibangun dapat memenuhi tujuan penelitian, yaitu menghasilkan gambar tokoh Jepang bergaya retro berdasarkan deskripsi teks. Kebutuhan sistem dibagi menjadi dua kategori utama: perangkat keras, perangkat lunak, serta kebutuhan fungsional dan non-fungsional.

- Perangkat keras
  a. Komputer atau Laptop:Spesifikasi minimum: Spesifikasi minimum yang dibutuhkan adalah RAM 8 GB dengan prosesor dual-core. Sistem operasi yang digunakan bisa berbasis Windows, macOS, atau Linux, asalkan mendukung penggunaan *browser web modern* untuk mengakses platform seperti Google Colab.

b. Koneksi internet yang stabil diperlukan untuk mengakses platform berbasis cloud seperti Google Colab dan untuk mengunggah atau mengunduh data serta model.

- Perangkat lunak

a. Google Colab :

Platform ini memberikan akses GPU atau TPU (misalnya Tesla T4 atau K80) yang dibutuhkan untuk mendukung pelatihan dan inferensi model dengan lebih cepat dan efisien.

b. *Library*

Tabel 3. 1 Tabel *Library*

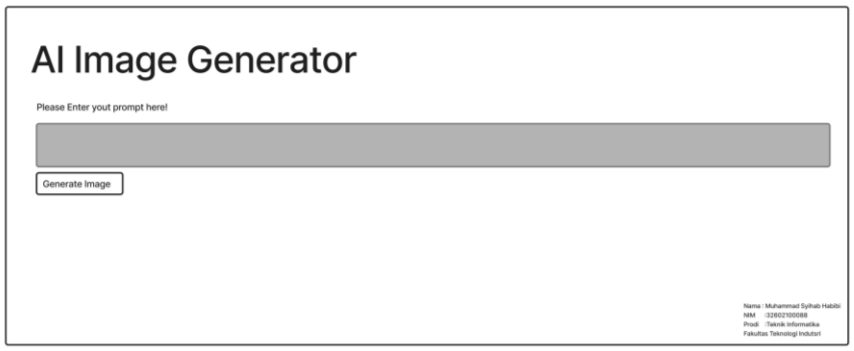
Library	**Deskripsi**	**Fungsi**
Pytorch	*Library* deep learning fleksibel untuk membangun, melatih, dan menguji model pembelajaran mendalam.	• Implementasi model Stable Diffusion 1.5. • Mendukung backpropagation untuk pelatihan model. • Memanfaatkan GPU untuk efisiensi.
Transformers	*Library* untuk pemrosesan teks dan encoding representasi vektor.	• Encoding deskripsi teks menggunakan CLIP Text Encoder. • Integrasi dengan Diffusers untuk proses teks ke gambar.
Diffusers	*Library* khusus untuk *diffusion models*, termasuk Stable Diffusion.	• Pipeline untuk proses denoising. • Mendukung *fine-tuning* DreamBooth.

		• Akses ke pre-trained model Stable Diffusion.
Matplotlib	*Library* visualisasi data.	• Menampilkan grafik evaluasi fidelity dan personalisasi. • Visualisasi metrik pelatihan, seperti *loss function.*
NumPy	*Library* untuk komputasi numerik.	• Normalisasi data numerik. • Operasi pada array dan matriks untuk pemrosesan gambar dan analisis data.
Pandas	*Library* manipulasi data berbasis tabel.	• Pengorganisasian metadata *dataset* • Menyajikan hasil evaluasi fidelitas dalam tabel.

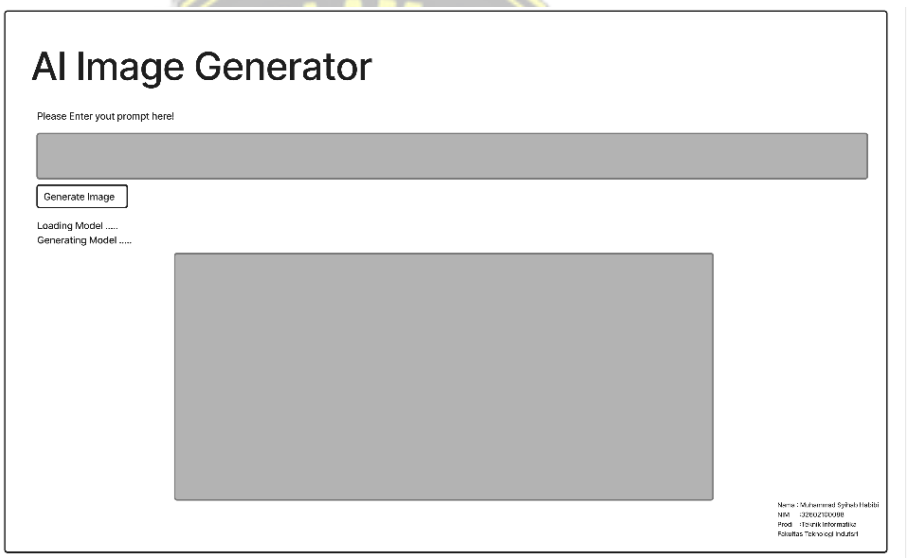
3.4 Perancangan User Interface

Antarmuka pengguna untuk AI Image Generator dirancang dengan pendekatan sederhana yang menekankan fungsi utama dan kemudahan navigasi. Pada tampilan awal, halaman menampilkan judul "AI Image Generator" di bagian atas sebagai elemen utama yang menarik perhatian. Tepat di bawahnya, terdapat kolom input berbentuk persegi panjang dengan latar abu-abu muda, yang menjadi tempat bagi pengguna untuk memasukkan deskripsi atau *prompt* terkait gambar yang ingin dihasilkan. Untuk memudahkan pengguna, kolom ini dilengkapi dengan teks panduan berupa placeholder bertuliskan "Please Enter your *prompt* here!". Di bawah kolom input, terdapat tombol "Generate Image" dengan desain bersih—

menggunakan latar putih dan teks hitam—yang dirancang agar mudah dikenali dan diakses.



Gambar 3. 4 Tampilan Awal



Gambar 3. 5 Tampilan Saat Generasi Gambar

Setelah tombol "Generate Image" diaktifkan, antarmuka memberikan umpan balik kepada pengguna dengan menampilkan status proses seperti "Loading Model..." dan "Generating Model..." di bawah tombol tersebut. Pada bagian bawah halaman, terdapat area besar berbentuk persegi panjang dengan latar abu-abu yang disiapkan untuk menampilkan hasil gambar setelah proses selesai. Desain elemen ini ditempatkan secara strategis agar pengguna dapat langsung memusatkan perhatian pada hasil yang dihasilkan oleh sistem.

BAB IV
HASIL DAN ANALISIS PENELITIAN

4.1 Persiapan Model dan Dataset

4.1.1 Pengumpulan dan Pengolahan Dataset

Dataset yang digunakan dalam penelitian ini berisi gambar karakter Jepang bergaya retro, yang mencakup variasi antara karakter laki-laki, perempuan, serta beberapa hewan. Setiap gambar memiliki latar belakang yang beragam, menambah kompleksitas dalam data. Jumlah dataset 50 gambar diambil dari internet, yang memberikan cukup variasi untuk analisis lebih lanjut.

Gambar 4. 1 Contoh Dataset yang Digunakan

Dataset dikumpulkan dari folder sumber dengan berbagai format gambar seperti .png, .jpg, dan .jpeg. Semua gambar diberi nama ulang menjadi format art1.jpg, art2.jpg, dan seterusnya. Langkah ini dilakukan untuk memastikan memastikan format nama file konsisten saat digunakan dalam model. Proses ini menggunakan pustaka shutil dan os dalam Python.

```python
import shutil

source_folder = "/content/file/MyDrive/Anime"
destination_folder = "/content/file/MyDrive/Anime/finale"
file_list = os.listdir(source_folder)
file_list.sort()
for index, file_name in enumerate(file_list):
    if file_name.lower().endswith(('.png', '.jpg', '.jpeg')):
        new_name = f"art{index + 1}.jpg"  # Format nama baru
        source_path = os.path.join(source_folder, file_name)
        destination_path = os.path.join(destination_folder, new_name)
        shutil.copy(source_path, destination_path)
        print(f"{file_name} -> {new_name}")
print("Pengaturan dataset selesai!")
```

```
0e6ca1f3-da28-4b81-bc66-0ac6174ec67b.jpeg -> art1.jpg
18a9a7ba-2655-4543-88d8-379e4cf0e2b5.jpeg -> art2.jpg
387ab8b9-6437-4b43-b8cc-a71ce5abd4d4.jpeg -> art3.jpg
66943777-fb00-4fab-83c7-4ee0b47c3c1d.jpeg -> art4.jpg
8846bc53-4858-4f41-ab1a-0bfee1b328b4.jpeg -> art5.jpg
9f11eda3-1694-4548-b843-8aa03586e16d.jpeg -> art6.jpg
Anime11.jpeg -> art7.jpg
Anime12.jpeg -> art8.jpg
Anime13.jpeg -> art9.jpg
Anime14.jpeg -> art10.jpg
```

Gambar 4. 2 Kode Pengurutan Nama File

. Gambar 4.1 menunjukkan pengurutan nama file, di mana *output* dari proses ini mencantumkan nama file lama beserta nama baru yang telah diubah. Hal ini bertujuan untuk memastikan bahwa hasil pengaturan nama file telah sesuai dan memudahkan pengecekan serta verifikasi terhadap perubahan yang telah dilakukan pada file tersebut. Pengurutan nama file yang sistematis memungkinkan model untuk dengan cepat mengidentifikasi urutan atau hubungan antar gambar, yang dapat meningkatkan efisiensi dalam pengolahan data. Misalnya, dengan menggunakan penamaan yang mencakup urutan atau kategori tertentu (seperti menggunakan angka atau label yang jelas), AI dapat memahami dan memproses gambar berdasarkan konteks yang diberikan oleh urutan tersebut, menghindari kebingunguan, dan mempercepat proses analisis atau pelatihan model.

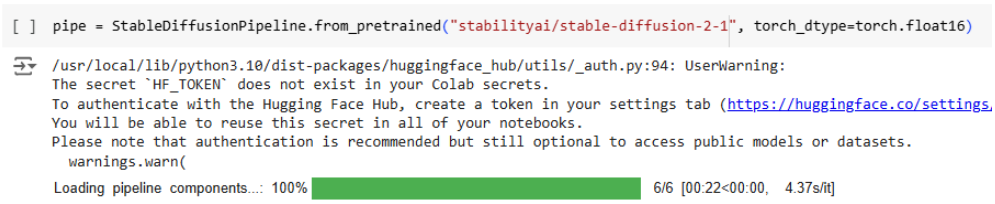
```
[ ]  def get_image_features(image):
         img_tensor = preprocess(image).unsqueeze(0).to(device)
         with torch.no_grad():
             image_features = clip_model.encode_image(img_tensor)
         return image_features

     similarity_scores_gen_vs_ref = []
     reference_image_features = [get_image_features(ref_img) for ref_img in reference_images]

     for i, gen_img in enumerate(generated_images):
         gen_img_features = get_image_features(gen_img)
         ref_similarity_scores = [
             torch.nn.functional.cosine_similarity(gen_img_features, ref_features).item()
             for ref_features in reference_image_features
         ]
         avg_ref_similarity = sum(ref_similarity_scores) / len(ref_similarity_scores)
         similarity_scores_gen_vs_ref.append(avg_ref_similarity)

     print("\nPerbandingan Skor Kesamaan (Gambar Hasil Generasi vs Gambar Referensi):")
     for i, avg_ref_score in enumerate(similarity_scores_gen_vs_ref):
         print(f"Gambar {i+1} - Skor rata-rata kesamaan dengan gambar referensi: {avg_ref_score:.2f}")
```

Gambar 4. 3 Kode Penggunaan Clip Score

Setelah dataset diatur, evaluasi dilakukan dengan menggunakan model CLIP untuk mengukur kesamaan (similarity) antara gambar referensi dan gambar yang dihasilkan oleh model. Proses ini melibatkan perbandingan embedding teks dan gambar, yang memungkinkan penilaian sejauh mana gambar yang dihasilkan sesuai dengan deskripsi yang diberikan. CLIP mengevaluasi atribut visual, seperti ekspresi wajah, gaya pakaian, serta elemen-elemen lainnya, untuk memastikan bahwa gambar yang dihasilkan mendekati atau mencerminkan karakteristik yang diinginkan sesuai dengan *prompt* yang diberikan.

4.1.2 Inisialisasi Model dengan Stable Diffusion v2.1

Stable Diffusion v2.1 digunakan sebagai model utama untuk generasi gambar. Model ini dilengkapi dengan kemampuan menangani input berbasis teks, namun memiliki tantangan seperti inkonsistensi *output* pada atribut visual spesifik yang diatasi melalui *fine-tuning* dengan metode *DreamBooth.*

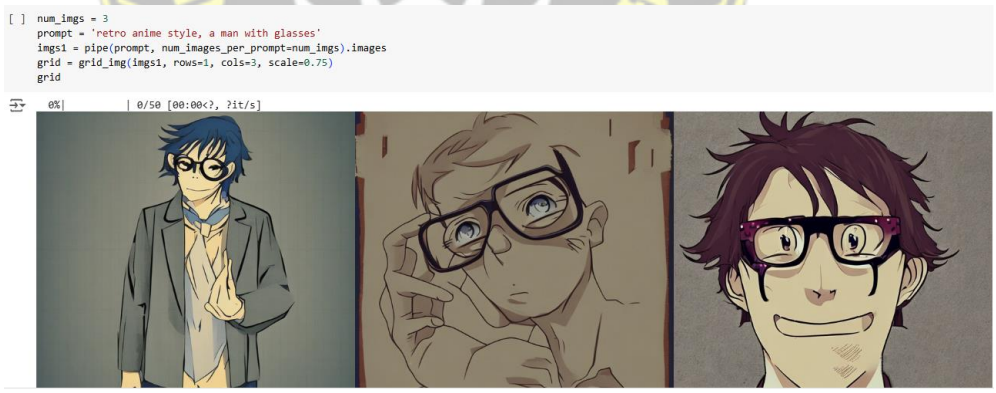


Gambar 4. 4 Penggunaan Stable Diffusion 2.1

Untuk mengakses model dari Hugging Face**,** token autentikasi diperlukan jika menggunakan akun pribadi atau model privat. Namun, model publik seperti `stabilityai/stable-diffusion-2-1` tidak memerlukan token khusus. Model diload dalam tipe data `float16` untuk mengurangi penggunaan memori GPU dan meningkatkan efisiensi.

4.2 Penggunaan Prompt

Prompt adalah deskripsi teks yang digunakan untuk mengarahkan proses generasi gambar. Penelitian ini mengeksplorasi berbagai variasi *prompt* untuk menguji fleksibilitas dan kemampuan model dalam menghasilkan gambar yang sesuai dengan deskripsi yang diberikan.



Gambar 4. 5 Generasi Model Awal

Hasilnya adalah tiga gambar yang dirender dalam gaya anime retro dengan subjek utama seorang pria berkacamata. Gambar-gambar tersebut kemudian divisualisasikan dalam format grid untuk analisis visual lebih lanjut. Proses ini memungkinkan evaluasi yang lebih efisien terhadap kesesuaian elemen-

elemen visual dalam setiap gambar, seperti atribut yang diinginkan, yang dapat diperiksa secara bersamaan dalam satu tampilan.

```
CLIP Score Gambar 1: 0.328125
CLIP Score Gambar 2: 0.33349609375
CLIP Score Gambar 3: 0.3212890625
Rata-rata CLIP Score: 0.32763671875
```

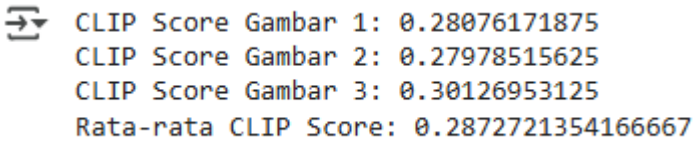
Gambar 4. 6 Pengujian Clip Score

Dalam eksperimen ini, skor yang diperoleh untuk masing-masing gambar adalah 0.3281, 0.3335, dan 0.3213, dengan rata-rata CLIP Score sebesar 0.3276. Nilai ini menunjukkan tingkat kesesuaian yang moderat antara *prompt* dan gambar yang dihasilkan. Namun, secara visual, gambar yang dihasilkan memiliki warna yang kurang mencerminkan gaya retro yang diharapkan. Elemen warna seperti tone hangat, pudar, atau estetika khas tahun 80-an tidak sepenuhnya terlihat pada hasil gambar. Untuk mengatasi hal ini, diperlukan langkah tambahan, seperti memperbaiki *prompt* dengan deskripsi yang lebih spesifik terkait warna retro, melakukan pasca-pemrosesan gambar menggunakan filter warna hangat dan efek grain, atau bahkan melakukan *fine-tuning* model menggunakan dataset dengan gaya warna retro. Dengan pendekatan ini, hasil generasi gambar diharapkan dapat lebih mencerminkan deskripsi gaya retro yang diinginkan.

4.2.1 *Prompt* dalam Bahasa Indonesia

Gambar 4. 7 Generasi Gambar dengan *Prompt* Berbahasa Indonesia

Prompt berbahasa Indonesia digunakan untuk menguji kemampuan model dalam memahami dan menghasilkan gambar berdasarkan deskripsi lokal. Hasil menunjukkan bahwa model dapat menangkap elemen umum dari deskripsi yang diberikan, tetapi kualitas detail visual dan spesifikasinya sedikit lebih rendah dibandingkan dengan *prompt* berbahasa Inggris.



Gambar 4. 8 Nilai Clip Score

Sebagai contoh, untuk *prompt "gaya anime retro, seorang pria berkacamata"*, model menghasilkan tiga gambar dengan CLIP Score masing-masing 0.2808, 0.2798, dan 0.3013, dengan rata-rata CLIP Score sebesar 0.2873.

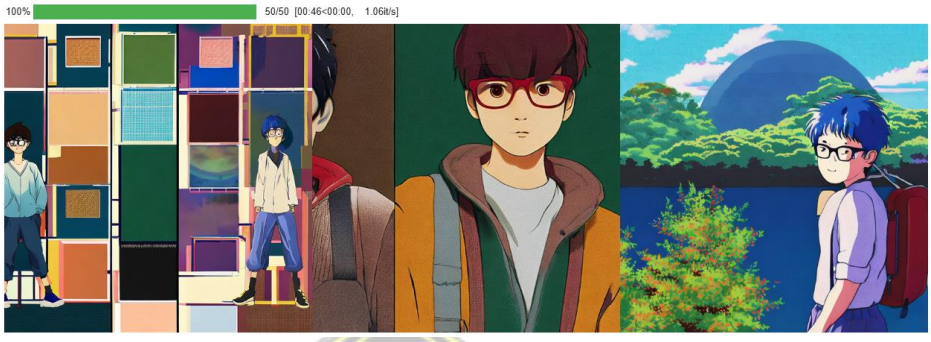
Nilai CLIP Score ini lebih rendah dibandingkan hasil dari *prompt* berbahasa Inggris, menunjukkan bahwa model memiliki keterbatasan dalam menangkap nuansa deskripsi dalam bahasa Indonesia, terutama terkait elemen detail dan atribut visual spesifik. Secara visual, gambar yang dihasilkan tetap relevan dengan deskripsi umum, namun gaya dan warna retro yang diharapkan tidak sepenuhnya tercapai. Untuk penelitian selanjutnya, penggunaan *prompt* dalam bahasa Inggris akan diutamakan guna memastikan kualitas dan akurasi gambar yang lebih tinggi, mengingat model lebih optimal dalam memahami deskripsi dalam bahasa Inggris.

4.2.2 Negative prompt

Negative *prompt* diterapkan untuk menghilangkan elemen yang tidak diinginkan, seperti detail wajah yang tidak realistis, anatomi tubuh yang salah, atau distorsi visual lainnya. Dengan negative prompt, hasil generasi menjadi lebih bersih dan sesuai dengan deskripsi yang diinginkan.

```
[ ]  neg_prompt = 'bad anatomy, ugly face, low quality, NSFW content, Harsh lines, overly sharp edges, or digital precision, Modern, futuristic'
```

Gambar 4. 9 Penggunaan Negative Prompt



Gambar 4. 10 Hasil setelah menggunakan Negative Prompt

Negative *prompt* yang diterapkan adalah "*bad anatomy, ugly face, low quality, NSFW content, harsh lines, overly sharp edges, or digital precision, modern, futuristic*". Parameter ini dirancang untuk meningkatkan kualitas gambar yang dihasilkan dengan menghindari berbagai elemen yang tidak diinginkan. Dalam penerapannya, parameter ini berfungsi untuk mencegah anatomi tubuh yang tidak realistis (*bad anatomy*), memastikan wajah karakter tetap estetis dan tidak mengalami distorsi (*ugly face*), serta meningkatkan ketajaman visual dengan mengeliminasi detail yang rendah atau kabur (*low quality*). Selain itu, parameter ini juga digunakan untuk menjaga keamanan konten dengan menghindari elemen yang bersifat eksplisit atau tidak pantas (*NSFW content*), serta untuk mencegah tampilan yang terlalu keras atau tegas dengan mengurangi garis yang terlalu tajam (*harsh lines, overly sharp edges*). Dengan menggunakan *prompt* ini, hasil akhir tetap sesuai dengan standar etika dan estetika yang diharapkan, serta terhindar dari elemen visual yang tidak sesuai dengan gaya yang diinginkan (*modern, futuristic*).

```
100%|████████████████████████████████████| 338M/338M [00:03<00:00, 101MiB/s]
CLIP Score Gambar 1: 0.333984375
CLIP Score Gambar 2: 0.33056640625
CLIP Score Gambar 3: 0.334228515625
Rata-rata CLIP Score: 0.3329264322916667
```

Gambar 4. 11 Clip Score

Meskipun ada peningkatan, perbedaan tersebut tidak terlalu signifikan, dengan *prompt* berbahasa Inggris menunjukkan sedikit kenaikan dari 0.31 menjadi 0.32, hanya bertambah 0.01, yang bisa dianggap sebagai peningkatan yang sangat kecil. Hal ini mengindikasikan bahwa meskipun negative *prompt* memberikan dampak positif dalam kualitas gambar

4.2.3 Spesifikasi Elemen Prompt

Pendekatan penggunaan *prompt* yang lebih spesifik melibatkan penambahan elemen-elemen detail yang mendalam, seperti warna, posisi, pola, atau atribut visual lainnya yang diminta dalam deskripsi. Dengan memperkenalkan informasi lebih terperinci dalam prompt, model dapat menghasilkan gambar yang lebih akurat dan sesuai dengan harapan. Penambahan detail semacam ini juga membantu dalam mempertajam fokus pada atribut visual tertentu, mengurangi kemungkinan hasil yang ambigu, dan meningkatkan keselarasan antara deskripsi teks dan gambar yang dihasilkan, sehingga memberikan representasi yang lebih jelas dan lebih mendalam sesuai dengan preferensi pengguna.

Spesifikasi *prompt* yang mencakup "*the color are earthy and natural with soft gradients, giving a vintage film-like quality to the atmosphere*" dirancang untuk memberikan nuansa visual yang hangat dan alami, dengan gradasi warna lembut yang menciptakan kesan kedalaman dan kehangatan pada gambar. Penggunaan warna yang bersifat *earthy*, seperti cokelat, hijau daun, atau krem, menguatkan kesan natural dan organik yang sering ditemukan dalam estetika retro. Gradasi warna yang halus menambah kesan transisi yang lembut antara satu warna dengan warna lainnya, memberikan atmosfer yang lebih nostalgik dan lembut, seolah-olah gambar tersebut diambil dari film klasik dengan kualitas visual yang lebih tua. Kesan "*vintage film-like*" ini menambah elemen retro, karena gaya warna ini sering kali dikaitkan dengan film-film lama yang menggunakan teknik pencahayaan dan pengolahan warna tertentu untuk menciptakan tampilan yang lebih hangat dan bernuansa nostalgia. Dengan demikian, spesifikasi *prompt* ini mendukung penciptaan

gambar yang tidak hanya visual, tetapi juga atmosferik, yang menambah kedalaman emosional dan kesan retro yang kuat pada gambar yang dihasilkan.

Gambar 4. 12 Hasil Gambar dengan *Prompt* Spesifik

```
Perbandingan Skor Kesamaan (Gambar yang Dihasilkan vs Gambar Referensi):
Gambar 1 - Skor rata-rata kesamaan dengan gambar referensi: 0.69
Gambar 2 - Skor rata-rata kesamaan dengan gambar referensi: 0.60
Gambar 3 - Skor rata-rata kesamaan dengan gambar referensi: 0.65
```

Gambar 4. 13 Perbandingan Gambar dengan Referensi

Perbandingan skor kesamaan dengan gambar referensi menunjukkan bahwa Gambar 1 memiliki rata-rata skor tertinggi sebesar 0.69, diikuti oleh Gambar 3 dengan skor 0.65, dan Gambar 2 dengan skor 0.60. Hasil ini mengindikasikan bahwa model lebih cenderung menghasilkan elemen visual yang lebih mendekati gambar referensi, dengan Gambar 1 menunjukkan tingkat kesamaan yang paling kuat. Skor ini mencerminkan sejauh mana elemen-elemen visual pada gambar yang dihasilkan sesuai dengan deskripsi dan atribut yang diberikan dalam referensi, yang berarti model berhasil menangkap lebih baik detail-detail yang diminta dalam prompt.

4.3 Pengaturan Parameter dan Pelatihan Model

a. Seed

Gambar 4. 14 Gambar dengan seed 777

Untuk memastikan konsistensi gambar yang dihasilkan, parameter seed digunakan. Dengan menetapkan nilai seed yang sama, generator dapat menghasilkan gambar yang serupa meskipun dilakukan pada waktu yang berbeda. Sebagai contoh, gambar 4.14 menunjukkan hasil gambar dengan seed 777.

b. Inference steps

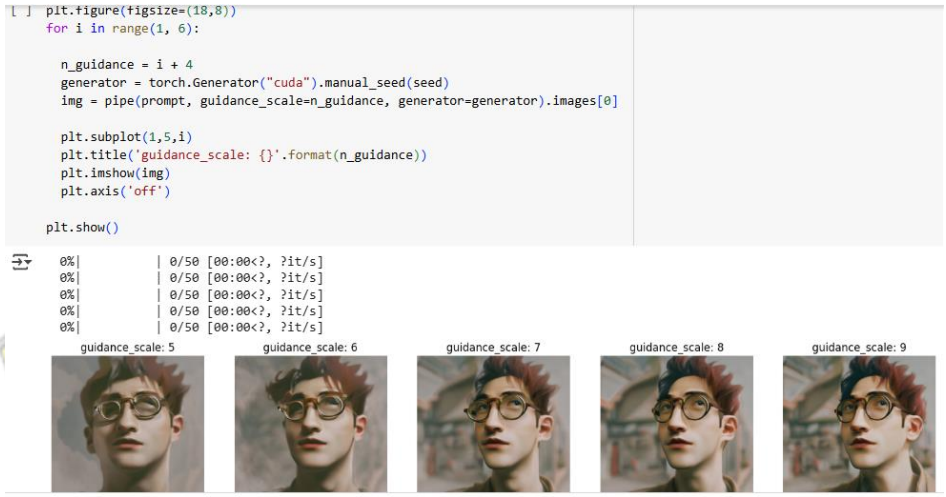
```
[ ] import matplotlib.pyplot as plt

    plt.figure(figsize=(18,8))
    for i in range(1, 6):
      n_steps = i * 10
      generator = torch.Generator('cuda').manual_seed(seed)
      imginference = pipe(prompt, num_inference_steps=n_steps, generator=generator).images[0]
      plt.subplot(1, 5, i)
      plt.title('num_inference_steps: {}'.format(n_steps))
      plt.imshow(img)
      plt.axis('off')
    plt.show()
```

```
0%|          | 0/10 [00:00<?, ?it/s]
0%|          | 0/20 [00:00<?, ?it/s]
0%|          | 0/30 [00:00<?, ?it/s]
0%|          | 0/40 [00:00<?, ?it/s]
0%|          | 0/50 [00:00<?, ?it/s]
```

Gambar 4. 15 Nilai inference steps (10, 20, 30, 40, 50)

Jumlah langkah dalam pembuatan gambar ditentukan oleh parameter inference steps. Nilai yang lebih tinggi menghasilkan gambar dengan detail lebih baik, namun memerlukan waktu lebih lama. Gambar 4.15 menunjukkan hasil dengan beberapa nilai inference steps, seperti 10, 20, 30, 40, dan 50.

c. *Guidance scale* (CFG)



Gambar 4. 16 Nilai guidance scale (CFG) dari 5 hingga 9

Guidance scale atau CFG mengatur sejauh mana generator mengikuti deskripsi dalam prompt. Semakin kecil nilai CFG, gambar yang dihasilkan lebih bebas dan kreatif. Sebaliknya, nilai CFG yang lebih besar akan menghasilkan gambar yang lebih sesuai dengan deskripsi yang diberikan. Gambar 4.16 menampilkan contoh penggunaan nilai CFG antara 5 hingga 9.

d. *Image size* (dimensions)

```
[ ]  seed = 777
     generator = torch.Generator("cuda").manual_seed(seed)
     h, w = 512, 512
     img = pipe(prompt, height=h, width=w, generator=generator).images[0]
     img
```

Gambar 4. 17 Mengatur Dimensi Gambar Menjadi 512x512

Dimensi gambar mempengaruhi resolusi dan kualitas visual gambar tersebut. Dalam Stable Diffusion 1.5, ukuran gambar yang digunakan adalah 512x512 piksel. Pengaturan dimensi ini membutuhkan penyesuaian pada parameter tinggi dan lebar, seperti yang ditunjukkan dalam gambar 4.17 dengan dimensi 512x512.

Pemilihan dimensi 512x512 piksel pada model Stable Diffusion 2.1 mempertimbangkan keseimbangan antara kualitas gambar dan efisiensi komputasi. Ukuran ini mampu menghasilkan gambar berkualitas tinggi sambil menjaga waktu proses tetap singkat. Bentuk persegi (rasio aspek 1:1) memudahkan model dalam menangani gambar dalam ruang laten, karena tidak ada distorsi yang disebabkan oleh perbedaan rasio aspek. Dengan dimensi persegi, model dapat lebih optimal dalam memproses informasi visual dan menghasilkan gambar dengan ketajaman yang sesuai tanpa kehilangan detail penting. Selain itu, dimensi 512x512 memberikan kemudahan dalam penerapannya pada berbagai konteks atau kebutuhan visual, menjadikannya pilihan yang sesuai dalam menghasilkan gambar yang konsisten dan berkualitas.

4.3.1 *Fine-tuning* dengan DreamBooth

Metode *fine-tuning* DreamBooth diterapkan untuk menyesuaikan model dengan dataset spesifik. Proses ini memungkinkan model menangkap detail visual gaya retro, seperti pola pakaian dan ekspresi wajah yang unik, guna mengatasi tantangan inkonsistensi *output*.

```
!python3 train_dreambooth.py \
  --pretrained_model_name_or_path="stabilityai/stable-diffusion-2-1" \
  --output_dir="/content/file/MyDrive/output2" \
  --revision="main" \
  --with_prior_preservation --prior_loss_weight=1.0 \
  --seed=777 \
  --resolution=512 \
  --train_batch_size=1 \
  --train_text_encoder \
  --mixed_precision="fp16" \
  --use_8bit_adam \
  --gradient_accumulation_steps=1 \
  --learning_rate=1e-6 \
  --lr_scheduler="constant" \
  --lr_warmup_steps=80 \
  --num_class_images=20 \
  --sample_batch_size=4 \
  --max_train_steps=1000 \
  --instance_prompt="1980s-inspired anime style" \
  --instance_data_dir="/content/file/MyDrive/Anime/finale" \
  --class_data_dir="/content/file/MyDrive/Anime/finale" \
  --class_prompt="1980s-inspired anime style"
```

Gambar 4. 18 Finetuning dengan Dreambooth

Dalam pelatihan ini, *fine-tuning* dilakukan dengan menggunakan instance *prompt* bertema gaya anime yang terinspirasi dari era 1980-an, yang menekankan elemen-elemen khas dari periode tersebut. Fokus utama dari *prompt* ini adalah pada penggunaan palet warna hangat yang sering ditemukan dalam karya-karya anime era 1980-an, tekstur lembut yang memberi kesan retro, serta atribut visual yang menggambarkan nuansa nostalgia, seperti desain karakter dan latar belakang yang mencerminkan ciri khas animasi pada masa itu. Pendekatan ini bertujuan untuk membuat model lebih sensitif dalam menghasilkan gambar yang konsisten dengan gaya visual tersebut.

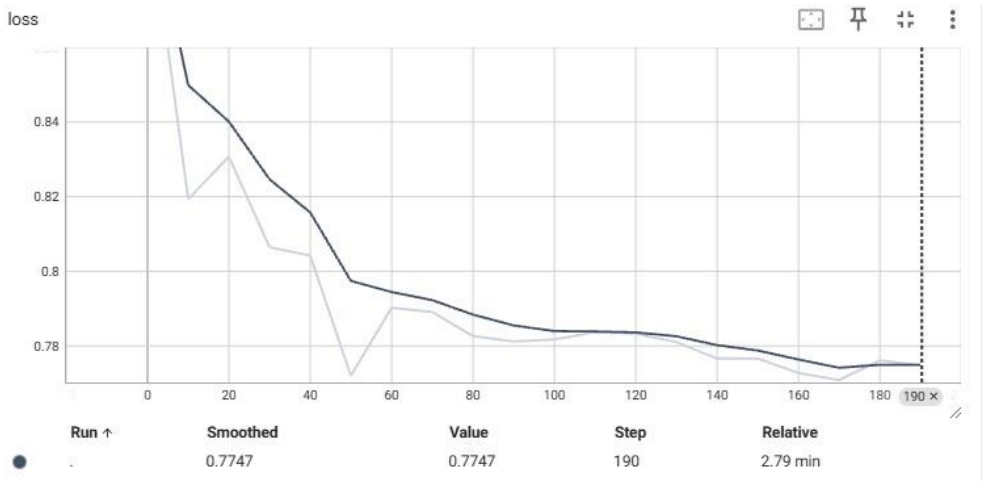
1. `--seed=777`: Pengaturan nilai seed memastikan reprodusibilitas hasil eksperimen. Dengan menggunakan nilai seed yang tetap, model akan menghasilkan hasil yang konsisten setiap kali pelatihan dijalankan, mencegah variasi yang tidak diinginkan akibat kondisi acak saat proses pelatihan.

2. `--resolution=512`: Dimensi gambar 512x512 digunakan untuk menjaga keseimbangan antara kualitas gambar dan penggunaan memori. Ukuran ini menghindari penggunaan memori yang berlebihan dan memungkinkan model untuk bekerja dalam batas kapasitas GPU yang terbatas, tanpa mengorbankan resolusi gambar yang dihasilkan.
3. `--train_batch_size=1`: Pengaturan *batch* size ke 1 bertujuan untuk mengurangi beban memori GPU, terutama saat bekerja dengan model besar dan dataset yang memiliki gambar dengan resolusi tinggi. Hal ini memungkinkan model untuk berjalan tanpa mengalami kekurangan memori, meskipun mengurangi efisiensi pelatihan yang dapat terjadi pada *batch* size lebih besar.
4. `--train_text_encoder`: Mengaktifkan pelatihan encoder teks memastikan bahwa representasi teks yang diberikan dalam bentuk deskripsi dapat dioptimalkan untuk menghasilkan gambar yang lebih akurat. Hal ini penting untuk mencegah model menghasilkan *output* yang tidak relevan dengan deskripsi yang diminta.
5. `--mixed_precision="fp16"`: Penggunaan presisi campuran (*mixed precision*) dengan tipe data fp16 mengurangi penggunaan memori GPU, memungkinkan pelatihan dengan *batch* size yang lebih besar tanpa membebani kapasitas memori. Ini juga mempercepat pelatihan dengan mengurangi waktu yang diperlukan untuk setiap iterasi.
6. `--use_8bit_adam`: Mengaktifkan optimizer Adam dengan 8-bit precision memungkinkan penghematan memori yang signifikan dan mempercepat pelatihan tanpa mengorbankan akurasi, yang sangat penting dalam pelatihan model besar seperti yang digunakan dalam pengolahan gambar.
7. `--gradient_accumulation_steps=1`: Pengaturan ini memastikan bahwa perhitungan gradien dilakukan setiap langkah pelatihan, yang menjaga kontrol pada update bobot model tanpa menambah beban memori secara berlebihan, meskipun dengan *batch* size yang kecil.
8. `--learning_rate=1e-6`: Learning rate yang sangat kecil ini diterapkan untuk mencegah model dari perubahan yang terlalu besar pada parameter

selama pelatihan, yang dapat menyebabkan model tidak stabil atau bahkan gagal untuk konvergen. Pengaturan ini bertujuan untuk meningkatkan kestabilan pelatihan pada dataset yang lebih kecil atau lebih spesifik.

9. --`lr_scheduler="constant"`: Menggunakan scheduler dengan langkah lr yang tetap (*constant*) bertujuan untuk menjaga laju pembelajaran konstan sepanjang pelatihan, menghindari fluktuasi yang bisa merugikan pelatihan model, terutama pada tahap awal yang sensitif terhadap perubahan learning rate.
10. --`lr_warmup_steps=80`: Dengan memanaskan laju pembelajaran selama 80 langkah pertama, model akan secara bertahap beradaptasi dengan dataset, mengurangi kemungkinan kesalahan besar pada awal pelatihan yang dapat menyebabkan ketidakstabilan atau pelatihan yang buruk.
11. --`num_class_images=20`: Pengaturan ini memastikan ada 20 gambar untuk setiap kelas dalam dataset, yang berguna untuk menyediakan contoh yang cukup untuk pembelajaran model, namun tidak terlalu banyak sehingga menghindari masalah *overfitting*.
12. --`sample_batch_size=4`: Pengaturan ini menentukan ukuran *batch* sampel gambar yang dihasilkan per iterasi, memberikan ukuran *batch* yang cukup untuk evaluasi model selama pelatihan tanpa membebani memori GPU terlalu banyak.
13. --`max_train_steps=1000`: Menetapkan batasan jumlah langkah pelatihan yang maksimal membantu mengontrol waktu pelatihan dan mencegah model berlatih terlalu lama, yang bisa menyebabkan *overfitting*.

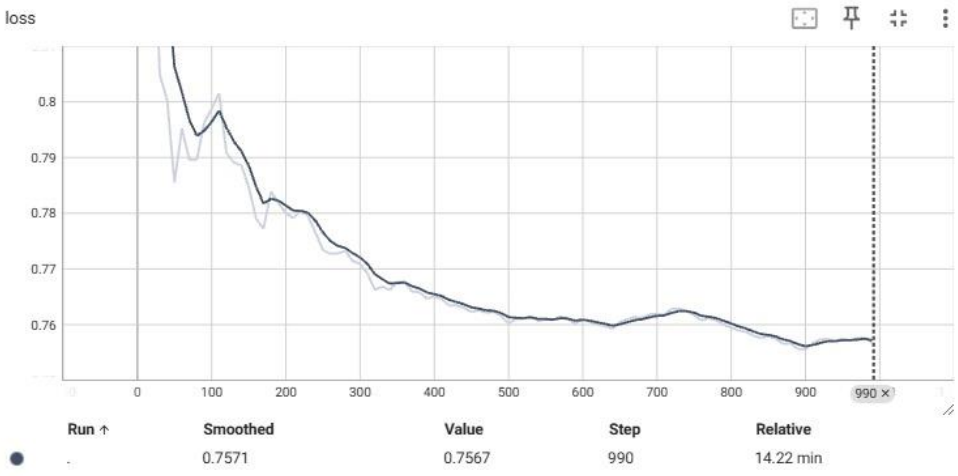
4.3.2 Grafik *Loss* Selama Proses Fine-Tuning



Gambar 4. 19 Grafik *Loss* Iterasi 200 Dataset 20

Gambar 4.19 menunjukkan grafik *loss* pada iterasi ke-200 dengan *dataset* berisi 10 sampel, menggunakan *learning rate scheduler* berbasis *cosine*, berbeda dengan eksperimen lainnya yang menggunakan *constant learning rate.* Pada eksperimen ini, nilai *loss* yang telah dihaluskan (*smoothed*) dan nilai aktual sama, yaitu 0.7747. Proses pelatihan berlangsung selama 2.79 menit dengan ukuran *batch size* sebesar 4.

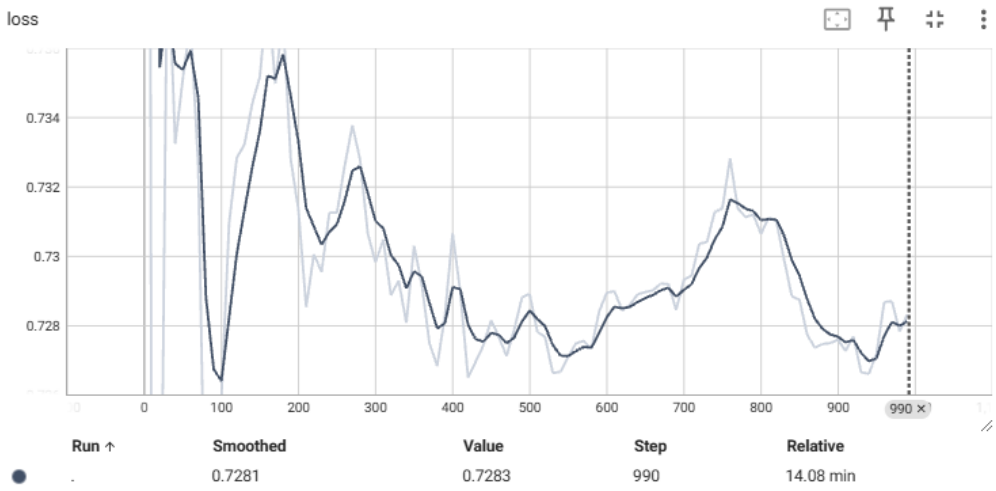
Penggunaan *cosine scheduler* memungkinkan model menyesuaikan learning rate secara bertahap, yang dapat meningkatkan stabilitas dan konvergensi selama pelatihan. Namun, jumlah dataset yang terbatas tetap menjadi kendala utama, karena data yang sedikit membatasi kapasitas model dalam menangkap pola yang lebih kompleks. Hal ini menunjukkan bahwa meskipun teknik pengaturan *learning rate* yang lebih adaptif dapat membantu, jumlah data tetap menjadi faktor krusial dalam efektivitas pelatihan model.



Gambar 4. 20 Grafik *Loss* Iterasi 1000 Dataset 30

Gambar 4.20 menampilkan grafik *loss* pada iterasi ke-1000 dengan dataset berisi 30 sampel, menggunakan *learning rate* konstan dan *batch size* sebesar 2. Nilai *loss* yang dihaluskan adalah 0.7571, sementara nilai aktualnya sedikit lebih rendah, yaitu 0.7567. Dibandingkan dengan eksperimen sebelumnya, waktu pelatihan meningkat secara signifikan menjadi 14.22 menit akibat jumlah iterasi yang lebih banyak dan *batch size* yang lebih kecil.

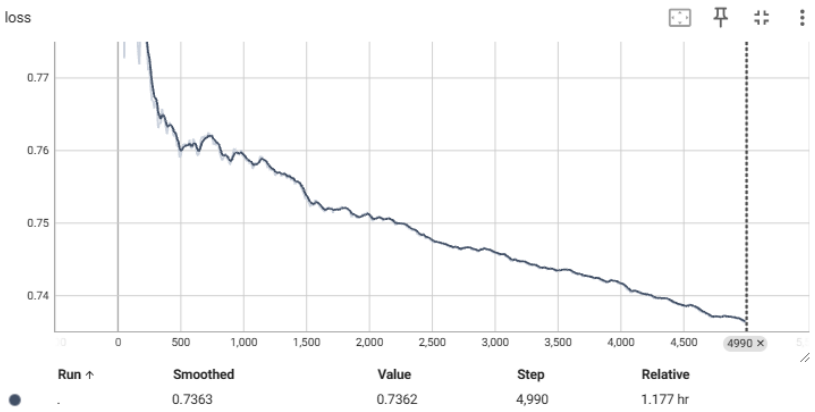
Peningkatan jumlah dataset memberikan hasil yang lebih stabil, menunjukkan bahwa model semakin mampu mengenali pola dengan lebih baik. Dengan lebih banyak data, model dapat menggeneralisasi informasi secara lebih akurat, mengurangi fluktuasi *loss*, dan meningkatkan performa keseluruhan. Meskipun waktu pelatihan lebih lama, hasil ini mengindikasikan bahwa jumlah data yang lebih besar berkontribusi positif terhadap stabilitas dan efektivitas pembelajaran model.



Gambar 4. 21 Grafik *Loss* Iterasi 1000 Dataset 50

Gambar 4.21 memperlihatkan hasil pelatihan model pada iterasi ke-1000 dengan dataset yang lebih besar, yaitu 50 sampel, menggunakan *learning rate* konstan dan *batch size* sebesar 4. Nilai loss yang dihaluskan mencapai 0.7281, dengan nilai aktual sebesar 0.7283. Meskipun jumlah dataset bertambah, waktu pelatihan tetap relatif efisien, yaitu sekitar 2.79 menit.

Dengan dataset yang lebih besar, model menunjukkan performa yang lebih baik dalam menurunkan *loss*, menandakan peningkatan kemampuan generalisasi. Penambahan jumlah sampel membantu model mengenali pola lebih akurat dan mengurangi *overfitting*, sehingga menghasilkan hasil pelatihan yang lebih stabil dibandingkan eksperimen sebelumnya.



Gambar 4. 22 Grafik *Loss* Iterasi 5000 Dataset 50

Gambar 4.22 menggambarkan grafik *loss* pada iterasi ke-5000 dengan dataset 50 sampel. Eksperimen ini tetap menggunakan *learning rate* konstan, tetapi dengan *batch size* yang lebih besar, yaitu 6. Nilai *loss* yang dihaluskan berada pada angka 0.7383, sementara nilai aktualnya sedikit lebih rendah, yaitu 0.7362. Waktu relatif yang digunakan untuk iterasi ini masih sekitar 2.79 menit. Meskipun iterasi lebih panjang dan ukuran *batch size* lebih besar, perbedaan nilai *loss* dibandingkan iterasi sebelumnya tidak terlalu signifikan, menunjukkan bahwa model telah mencapai titik stabil dalam pembelajaran.

4.3.3 Variasi Max Train

Tabel 4. 1 Hasil Pelatihan Model

Max Train	Ukuran *Output*	Dataset	Checkpoint
2000	25 GB	10	500
200	4.8 GB	20	-
1000	4.8 GB	30	-
1000	4.8 GB	50	-
5000	4.8 GB	50	-

Pada Tabel 4.1, parameter *Max Train* menentukan jumlah maksimum iterasi atau langkah pelatihan yang dilakukan pada dataset yang diberikan. Nilai ini berpengaruh langsung terhadap durasi pelatihan serta kualitas model

yang dihasilkan. Dari variasi yang tercantum, Max Train berkisar antara 200 hingga 5000, dengan ukuran output yang bervariasi antara 4.8 GB hingga 25 GB. Semakin besar nilai Max Train, semakin lama model dilatih, yang biasanya meningkatkan kualitas hasil tetapi juga memerlukan sumber daya komputasi yang lebih besar.

Sebagai contoh, pada Max Train 200, ukuran output tercatat sebesar 4.8 GB, sementara pada Max Train 5000, ukuran output tetap 4.8 GB, meskipun jumlah dataset meningkat dari 20 ke 50 dan tidak terdapat checkpoint yang digunakan. Namun, pada Max Train 2000, ukuran output meningkat signifikan hingga 25 GB dengan jumlah dataset 10 dan checkpoint 500. Hal ini menunjukkan bahwa jumlah iterasi serta konfigurasi dataset dan checkpoint memiliki dampak signifikan terhadap ukuran model akhir. Selain itu, dataset yang digunakan bervariasi dari 10 hingga 50, tetapi tidak selalu berhubungan langsung dengan ukuran output atau jumlah checkpoint yang digunakan.

4.3.4 Regularisasi L2

```python
class CLIPWithL2Regularization(nn.Module):
    def __init__(self, original_model, l2_lambda):
        super(CLIPWithL2Regularization, self).__init__()
        self.clip_model = original_model
        self.l2_lambda = l2_lambda

    def encode_image(self, image):
        image_features = self.clip_model.encode_image(image)
        return image_features

    def encode_text(self, text):
        text_features = self.clip_model.encode_text(text)
        return text_features

    def compute_l2_penalty(self):
        l2_penalty = 0
        for param in self.clip_model.parameters():
            l2_penalty += torch.sum(param ** 2)
        return self.l2_lambda * l2_penalty
```

Gambar 4. 23 Regularisasi L2 pada Clip Encoder

Selama pelatihan, regularisasi L2 diterapkan untuk menghindari ketergantungan model yang berlebihan pada dataset pelatihan yang terbatas. Dengan memberikan penalti pada bobot model yang memiliki nilai besar, regularisasi L2 membantu model untuk menjaga keseimbangan dalam pemrosesan data, sehingga meningkatkan kemampuan model dalam melakukan generalisasi. Hal ini mengurangi kemungkinan terjadinya *overfitting*, yaitu kondisi di mana model terlalu terikat pada detail spesifik dari data pelatihan dan kehilangan kemampuan untuk bekerja dengan data baru yang tidak terlihat sebelumnya.

4.4 Evaluasi dan Analisis *Output*

4.4.1 Pengujian *Output* Berdasarkan Jumlah Dataset

Gambar yang dihasilkan dievaluasi menggunakan *prompt* yang sama namun dengan jumlah dataset yang berbeda (10, 20, dan 30 gambar). Evaluasi ini dilakukan untuk menilai dampak jumlah dataset terhadap konsistensi atribut visual dan kesesuaian *output* dengan deskripsi prompt.

a. Dataset 10 Iterasi 2000

Gambar 4. 24 Gambar Hasil dengan 10 Dataset

Pewarnaan dalam gambar yang dihasilkan telah menunjukkan peningkatan konsistensi dengan gaya *retro anime*. Palet warna hangat dan atribut tekstur lembut berhasil diterapkan. Namun, terdapat inkonsistensi visual yang signifikan pada dua dari tiga gambar yang dihasilkan, di mana muncul banyak orang di bagian latar belakang. Hal ini mengindikasikan

bahwa model menghadapi tantangan dalam mempertahankan fokus pada subjek utama karena keterbatasan variasi data dalam dataset. Jumlah dataset yang rendah meningkatkan risiko *overfitting* pada pola yang tidak diinginkan, seperti pengulangan elemen latar belakang.

b. Dataset 20

Gambar 4. 25 Gambar Hasil dengan 20 Dataset

Ketika jumlah dataset ditingkatkan menjadi 20 gambar, hasil generasi menunjukkan perbaikan pada tingkat detail dan fokus visual. Sebagian besar gambar berhasil menggambarkan subjek utama dengan latar belakang yang lebih terkontrol. Namun, pada satu gambar, elemen latar belakang masih memuat banyak orang yang tidak relevan dengan prompt. Hal ini menunjukkan bahwa meskipun peningkatan jumlah dataset membantu mengurangi ketergantungan model pada pola acak, dataset yang lebih besar diperlukan untuk mengatasi inkonsistensi secara menyeluruh.

c. Dataset 30

Gambar 4. 26 Gambar Hasil dengan 30 Dataset

Dengan jumlah dataset sebesar 30 gambar, hasil generasi menjadi jauh lebih stabil dan rapi. Pewarnaan tetap konsisten dengan gaya retro, dan elemen dekoratif seperti pola pakaian berhasil ditampilkan dengan lebih presisi. Tidak ada lagi gambar yang memuat elemen latar belakang yang mengganggu, seperti banyaknya orang yang tidak relevan. Hal ini menunjukkan bahwa jumlah dataset yang lebih besar memungkinkan model untuk belajar pola atribut visual secara lebih baik tanpa terlalu terpengaruh oleh pola yang tidak diinginkan, sehingga risiko *overfitting* dapat diminimalkan.

d. Dataset 50 dengan iterasi 5000

Gambar 4. 27 Gambar Hasil dengan 50 Dataset Dengan Iterasi 5000

Dengan jumlah dataset sebesar 50 gambar, hasil generasi menjadi semakin stabil dan teratur. Pewarnaan tetap konsisten dengan gaya retro, namun satu hal yang berbeda adalah latar belakang yang hanya dapat digenerasi sebagai kebun bunga. Elemen dekoratif, seperti pola pakaian, masih berhasil ditampilkan dengan lebih presisi, tetapi model kesulitan untuk menghasilkan latar belakang selain kebun bunga. Hal ini menunjukkan bahwa meskipun jumlah dataset yang lebih besar memungkinkan model untuk belajar pola atribut visual dengan lebih baik, terdapat batasan pada variasi latar belakang yang bisa dihasilkan, mengindikasikan adanya pengaruh kuat dari pola latar belakang yang dominan dalam dataset.

e. Dataset 50 dengan iterasi 1000

Gambar 4. 28 Gambar Hasil dengan 50 Dataset Dengan Iterasi 1000

Dengan jumlah dataset sebesar 50 gambar dan iterasi sebanyak 1000, hasil generasi menunjukkan kualitas yang lebih baik dibandingkan dengan iterasi 5000. Detail visual, seperti tekstur dan bentuk objek, tetap terjaga dengan baik, sementara pewarnaan lebih seimbang tanpa *overfitting* pada pola tertentu. Sebaliknya, pada iterasi 5000, meskipun model semakin memahami pola dalam dataset, hasil generasi cenderung menjadi terlalu spesifik dan kehilangan variasi, terutama dalam elemen seperti latar belakang dan komposisi. Hal ini menunjukkan bahwa meskipun jumlah iterasi yang lebih tinggi dapat meningkatkan akurasi model dalam mengenali detail dataset, terlalu banyak iterasi justru dapat mempersempit ruang kreativitas model, menyebabkan hasil yang kurang bervariasi dan lebih terikat pada pola dominan dalam dataset.

4.4.2 Analisis Overfitting pada Model

Overfitting terjadi ketika model terlalu menyesuaikan diri dengan dataset pelatihan, sehingga kehilangan kemampuan untuk menggeneralisasi pola pada data baru (Zeng *dkk.*, 2024). Dalam eksperimen ini, beberapa faktor yang mempengaruhi overfitting dianalisis, termasuk jumlah dataset, iterasi pelatihan, serta penggunaan regularisasi L2.

1. Dampak Jumlah Dataset terhadap Overfitting

   Berdasarkan hasil yang diperoleh pada berbagai eksperimen, jumlah dataset memiliki dampak signifikan terhadap risiko overfitting. Pada

eksperimen dengan dataset yang lebih kecil (10 atau 20 sampel), model menunjukkan kecenderungan untuk terlalu menyesuaikan diri dengan pola dalam data pelatihan. Hal ini terlihat pada hasil generasi yang mengandung elemen latar belakang berulang yang tidak diinginkan, serta kurangnya variasi dalam komposisi visual.

Ketika jumlah dataset ditingkatkan menjadi 30 dan 50 sampel, hasil generasi menjadi lebih stabil. Model lebih mampu mengenali pola atribut visual tanpa terlalu terpaku pada pola spesifik dalam dataset pelatihan. Namun, pada eksperimen dengan iterasi tinggi (5000) meskipun menggunakan dataset 50 sampel, model justru mengalami overfitting terhadap pola latar belakang tertentu, seperti hanya mampu menghasilkan latar belakang berbentuk kebun bunga. Hal ini menunjukkan bahwa meskipun jumlah dataset lebih besar dapat mengurangi overfitting, jumlah iterasi yang terlalu tinggi dapat mempersempit variasi hasil model.

2. Pengaruh Jumlah Iterasi terhadap Overfitting

   Seiring bertambahnya jumlah iterasi, model semakin menyesuaikan parameter-parameter internalnya terhadap dataset pelatihan. Namun, ketika iterasi mencapai angka yang terlalu tinggi, seperti 5000, model mulai kehilangan kemampuan untuk menghasilkan variasi dalam outputnya.

   Pada eksperimen dengan iterasi 1000, hasil generasi tetap konsisten dengan gaya visual yang diinginkan tanpa menunjukkan pola yang terlalu spesifik atau repetitif. Sebaliknya, pada iterasi 5000, meskipun model semakin memahami detail dataset, variasi latar belakang menjadi sangat terbatas. Hal ini menunjukkan bahwa model telah mengalami overfitting terhadap pola tertentu dalam dataset, sehingga kehilangan fleksibilitas dalam menghasilkan variasi baru.

4.4.3 Penggunaan Regularisasi L2

a. Sebelum menggunakan L2

Gambar 4. 29 Hasil Generalisasi Sebelum Menggunakan L2

```
Perbandingan Skor Kesamaan (Gambar yang Dihasilkan vs Gambar Referensi):
Gambar 1 - Skor rata-rata kesamaan dengan gambar referensi: 0.73
Gambar 2 - Skor rata-rata kesamaan dengan gambar referensi: 0.70
Gambar 3 - Skor rata-rata kesamaan dengan gambar referensi: 0.68
```

Gambar 4. 30 Perbandingan Score Kesamaan

Gambar 4.29 dan gambar 4.30 menunjukkan hasil generalisasi sebelum penerapan regularisasi L2, di mana gambar yang dihasilkan memiliki kesamaan yang cukup tinggi dengan gambar referensi, dengan nilai skor kesamaan berturut-turut 0.73, 0.70, dan 0.68. Nilai-nilai skor ini mencerminkan bahwa meskipun model dapat menghasilkan gambar yang sesuai dengan deskripsi umum, variasi tersebut menunjukkan bahwa gambar yang dihasilkan masih bisa ditingkatkan untuk lebih mendekati referensi yang diinginkan.

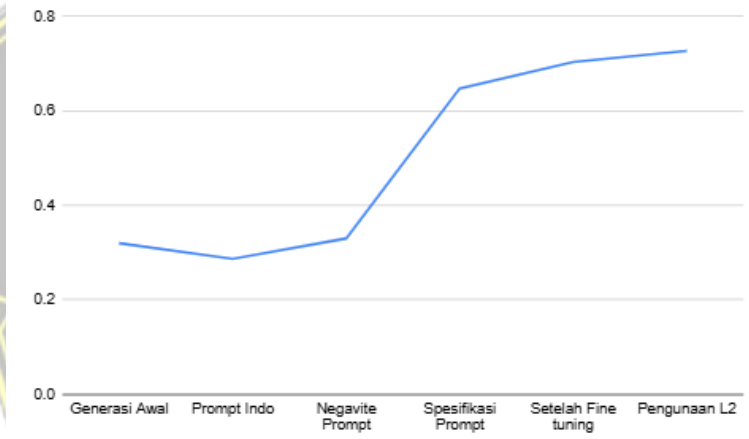
b. Sesudah menggunakan L2

Gambar 4. 31 Hasil Generalisasi Setelah Menggunakan L2

```
Perbandingan Skor Kesamaan (Gambar yang Dihasilkan vs Gambar Referensi):
Gambar 1 - Skor rata-rata kesamaan dengan gambar referensi: 0.70
Gambar 2 - Skor rata-rata kesamaan dengan gambar referensi: 0.73
Gambar 3 - Skor rata-rata kesamaan dengan gambar referensi: 0.75
```

Gambar 4. 32 Perbandingan Score Kesamaan

Gambar 4.31 dan gambar 4.32 menunjukkan hasil generalisasi setelah penerapan regularisasi L2, di mana penggunaan L2 berhasil meningkatkan kesesuaian gambar dengan *prompt* tanpa mengurangi variasi visual. Dengan nilai skor kesamaan berturut-turut 0.70, 0.73, dan 0.75, gambar-gambar yang dihasilkan menunjukkan kesesuaian yang lebih baik dengan gambar referensi, baik dalam hal detail maupun keseluruhan visual.



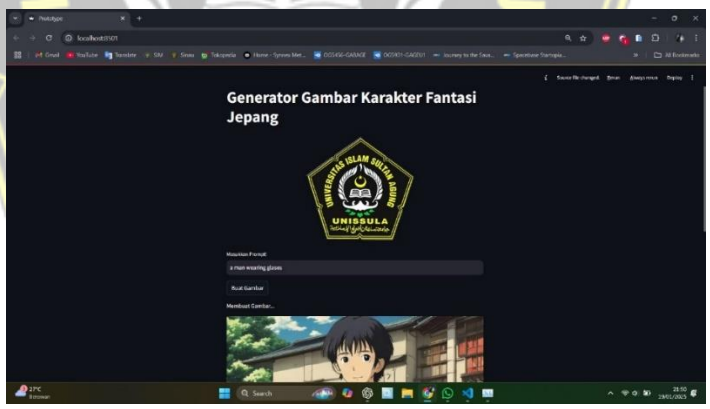
Gambar 4. 33 Perbandingan Clip Score Pada Tiap Tahap

Gambar 4.33 menunjukkan perbandingan CLIP score pada setiap tahap, dengan grafik yang menunjukkan peningkatan signifikan setelah penerapan regularisasi L2. Peningkatan ini menunjukkan bahwa regularisasi L2 tidak hanya memperbaiki kualitas kesesuaian gambar, tetapi juga menjaga variasi visual yang diinginkan, memastikan gambar yang dihasilkan lebih sesuai dengan *prompt* yang diberikan.

4.5 Implementasi Sistem Berbasis Web

Gambar 4. 34 Tampilan Awal di Local Menggunakan Streamlit

Sistem generasi gambar ini diimplementasikan dalam bentuk aplikasi web dengan menggunakan Streamlit, yang dirancang untuk memudahkan pengguna dalam menghasilkan gambar karakter Jepang bergaya retro berdasarkan deskripsi teks.



Gambar 4. 35 Tampilan Saat Menajalankan Model di Local Menggunakan Streamlit

Di dalam *website*, gambar yang dihasilkan tidak hanya mencakup karakter itu sendiri, tetapi juga dapat memuat aksi atau kegiatan yang sedang dilakukan oleh karakter, seperti naik sepeda atau melakukan kegiatan lain yang disesuaikan dengan deskripsi. Selain itu, sistem juga memungkinkan pengguna untuk menghasilkan latar belakang yang sesuai dengan cerita atau

konteks, seperti pasar atau desa, memberikan gambaran yang lebih hidup dan kontekstual.

Gambar 4. 36 Hasil Generalisasi Local Menggunakan Streamlit

Dengan kemampuan ini, pengguna dapat dengan mudah menciptakan scene yang lebih dinamis dan detail, di mana karakter dan latar belakang bekerja sama untuk menciptakan visual yang sesuai dengan narasi yang diinginkan. Sistem ini memungkinkan kebebasan dalam menciptakan gambar sesuai dengan keinginan pengguna hanya dengan memasukkan deskripsi teks, sehingga memperkaya pengalaman interaktif dalam pembuatan gambar berbasis AI.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini berhasil membangun generator gambar karakter Jepang bergaya retro menggunakan *diffusion models* dengan pendekatan *fine-tuning* DreamBooth. Model ini mampu mengintegrasikan deskripsi teks sebagai input dengan elemen visual khas gaya retro, seperti palet warna pastel, tekstur lembut, dan pola geometris yang sederhana. Hasilnya adalah gambar yang realistis dan relevan dengan deskripsi pengguna, mencerminkan estetika retro yang diinginkan. *Fine-tuning* DreamBooth memungkinkan penyesuaian elemen-elemen ini agar sesuai dengan atribut visual yang diminta.

Pemrosesan data, termasuk pengumpulan dan pengolahan dataset, memainkan peran penting dalam menjaga konsistensi dan variasi gaya. Jumlah dataset yang memadai terbukti menjadi faktor kunci dalam menghasilkan generalisasi model yang baik, mencegah model terlalu bergantung pada pola spesifik dalam data pelatihan. Selain itu, penerapan regularisasi L2 membantu mencegah overfitting, dengan peningkatan kualitas hasil sebesar 0.02 dalam uji coba CLIP. Evaluasi hasil menunjukkan bahwa dengan dataset yang lebih kaya dan beragam, model dapat menghasilkan gambar dengan detail yang lebih konsisten. Berdasarkan hasil uji coba CLIP, terjadi peningkatan skor dari sekitar 0.30-an menjadi 0.70-an, menunjukkan peningkatan yang signifikan dalam kesesuaian antara gambar yang dihasilkan dan deskripsi input. Namun, masih terdapat kendala dalam penerjemahan prompt, yang menyebabkan beberapa gambar tidak sepenuhnya merepresentasikan instruksi teks. Masalah ini menjadi tantangan yang perlu dipelajari lebih lanjut dalam penelitian selanjutnya.

5.2 Saran

Penelitian selanjutnya dapat mengeksplorasi variasi gaya visual lain, seperti futuristik atau minimalis, untuk memperluas aplikasi model generatif ini dalam industri kreatif. Selain itu, *fine-tuning* model dengan dataset yang

lebih besar dan lebih beragam dapat dilakukan untuk meningkatkan kualitas hasil, terutama dalam mengatasi inkonsistensi visual. Integrasi sistem dengan fitur interaktif, seperti pengeditan *prompt* secara langsung atau penyempurnaan gambar, juga disarankan untuk memberikan fleksibilitas lebih besar kepada pengguna.

Terakhir, optimalisasi waktu inferensi dengan memanfaatkan metode distilasi model atau arsitektur generatif yang lebih efisien dapat membantu meningkatkan kecepatan generasi gambar tanpa mengorbankan kualitas visual.

Daftar Pustaka

Anderson, J. dan Akram, N. (2024) "Denoising Diffusion Probabilistic Models (DDPM) Dynamics: Unraveling Change Detection in Evolving Environments," *Innovative Computer Sciences Journal*, 10(1), hal. 1–10.

Bengesi, S. *dkk.* (2024) "Advancements in Generative AI: A Comprehensive Review of GANs, GPT, Autoencoders, Diffusion Model, and Transformers.," *IEEE Access* [Preprint].

Berahmand, K. *dkk.* (2024) *Autoencoders and their applications in machine learning: a survey*, *Artificial Intelligence Review*. Springer Netherlands. Tersedia pada: https://doi.org/10.1007/s10462-023-10662-6.

Budinugroho, G. dan Islam, M.A. (2023) "Perancangan Buku Ilustrasi Infografis Sejarah Konsol Video Game Era 1980-an dan 1990-an," *BARIK-Jurnal S1 Desain Komunikasi Visual*, 4(3), hal. 75–89.

Cao, Y. *dkk.* (2023) "AnimeDiffusion: Anime Face Line Drawing Colorization via Diffusion Models," *arXiv preprint arXiv:2303.11137* [Preprint].

Chen, J. *dkk.* (2024) "Textdiffuser: Diffusion models as text painters," *Advances in Neural Information Processing Systems*, 36.

Croitoru, F.-A. *dkk.* (2023) "Diffusion models in vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9), hal. 10850–10869.

Everaert, M.N. *dkk.* (2023) "Diffusion in style," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, hal. 2251–2261.

Firdaus, A. (2023) "Keberhasilan Diplomasi Publik Jepang Melalui Budaya Populer: Tantangan Terhadap Identitas Nasional Generasi Muda Indonesia," *Jurnal Pendidikan dan Pengajaran*, 1(2), hal. 98–119. Tersedia pada: https://pijar.saepublisher.com/index.php/jpp/article/view/24.

Han, Z. *dkk.* (2024) "Parameter-efficient fine-tuning for large models: A comprehensive survey," *arXiv preprint arXiv:2403.14608* [Preprint].

Hidalgo, R. *dkk.* (2023) "Personalizing text-to-image diffusion models by fine-tuning classification for AI applications," in *Proceedings of SAI Intelligent Systems Conference*. Springer, hal. 642–658.

Hutagalung, M.A.I. (2024) “Penalized Maximum Likelihood Estimation dengan Algoritma Gradient descent pada Model Regresi Logistik Multinomial,” *IJM: Indonesian Journal of Multidisciplinary*, 2(6), hal. 673–683.

Jung, C. *dkk.* (2024) “FlowAVSE: Efficient Audio-Visual Speech Enhancement with Conditional Flow Matching,” hal. 2210–2214. Tersedia pada: https://doi.org/10.21437/interspeech.2024-701.

Lailiyah, S.M. (2024) “LAPORAN KP-PENERAPAN ILUSTRASI DAN PERANCANGAN ASET GRAFIS DALAM PROSES KERJA DI STUDIO MOARA MALANG.”

Li, Y. *dkk.* (2023) “SnapFusion: Text-to-Image Diffusion Model on Mobile Devices within Two Seconds,” *Advances in Neural Information Processing Systems*, 36(NeurIPS 2023), hal. 1–17.

Liu, X. *dkk.* (2024) “LaDiffGAN: Training GANs with Diffusion Supervision in Latent Spaces,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, hal. 1115–1125.

Luo, C. (2022) “Understanding diffusion models: A unified perspective,” *arXiv preprint arXiv:2208.11970* [Preprint].

Pang, L. *dkk.* (2024) “AttnDreamBooth: Towards Text-Aligned Personalized Text-to-Image Generation,” *arXiv preprint arXiv:2406.05000* [Preprint].

Parmar, G. *dkk.* (2024) “One-step image translation with text-to-image models,” *arXiv preprint arXiv:2403.12036* [Preprint].

Prasad, A. *dkk.* (2024) “Stable Diffusion Image Processing,” *Library Progress International*, 44(3), hal. 5917–5925.

Rakitin, D., Shchekotov, I. dan Vetrov, D. (2024) “Regularized Distribution Matching Distillation for One-step Unpaired Image-to-Image Translation.” Tersedia pada: http://arxiv.org/abs/2406.14762.

Rao, K.M. dan Patel, T. (2024) “Enhancing Control in Stable Diffusion Through Example-based Fine-Tuning and Prompt Engineering,” in *2024 5th International Conference on Image Processing and Capsule Networks (ICIPCN)*. IEEE, hal. 887–894.

Ruiz, N. *dkk.* (2023) “Dreambooth: Fine tuning text-to-image diffusion models for

subject-driven generation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, hal. 22500–22510.

Shi, J. (2024) "InstantBooth : Personalized Text-to-Image Generation without Test-Time Finetuning," hal. 8543–8552.

Siddique, N. *dkk.* (2021) "U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications," *IEEE Access*, 9, hal. 82031–82057. Tersedia pada: https://doi.org/10.1109/ACCESS.2021.3086020.

Sutedy, M.F. dan Qomariyah, N.N. (2022) "Text to image latent diffusion model with dreambooth fine tuning for automobile image generation," in *2022 5th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. IEEE, hal. 440–445.

Weng, W. dan Zhu, X. (2021) "UNet: Convolutional Networks for Biomedical Image Segmentation," *IEEE Access*, 9, hal. 16591–16603. Tersedia pada: https://doi.org/10.1109/ACCESS.2021.3053408.

Wu, Q. *dkk.* (2023) "Uncovering the Disentanglement Capability in Text-to-Image Diffusion Models," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2023-June, hal. 1900–1910. Tersedia pada: https://doi.org/10.1109/CVPR52729.2023.00189.

Xiang, W. *dkk.* (2023) "Denoising diffusion autoencoders are unified self-supervised learners," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, hal. 15802–15812.

Yang, L. *dkk.* (2023) "Diffusion Models: A Comprehensive Survey of Methods and Applications," *ACM Computing Surveys*, 56(4). Tersedia pada: https://doi.org/10.1145/3626235.

Zeng, W. *dkk.* (2024) "Infusion: Preventing customized text-to-image diffusion from overfitting," in *Proceedings of the 32nd ACM International Conference on Multimedia*, hal. 3568–3577.