IMPLEMENTASI *STABLE DIFFUSION* DAN *FINE-TUNING* LORA UNTUK PEMBUATAN LOGO

LAPORAN TUGAS AKHIR

Laporan Ini Disusun Untuk Memenuhi Salah Satu Syarat Memperoleh Gelar Sarjana Strata 1 (S1) Pada Program Studi S1 Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung

DISUSUN OLEH :

NAELA MUKAROMAH

NIM 32602100096

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM SULTAN AGUNG

SEMARANG

JANUARI 2025

FINAL PROJECT

IMPLEMENTATION OF *STABLE DIFFUSION* AND LORA *FINE-TUNING* FOR LOGO GENERATION

This report is prepared as part of the requirements for obtaining a Bachelor's Degree in the Informatics Engineering Program, Faculty of Industrial Technology, Sultan Agung Islamic University.

Arranged By :

NAELA MUKAROMAH

NIM 32602100096

INFORMATICS ENGINEERING DEPARTEMENT

FACULTY OF INDUSTRIAL TECHNOLOGY

SULTAN AGUNG ISLAMIC UNIVERSITY

SEMARANG

JANUARY 2025

LEMBAR PENGESAHAN

TUGAS AKHIR

Implementasi *Stable diffusion* Dan *Fine-Tuning* LoRA Untuk Pembuatan Logo

NAELA MUKAROMAH

32602100096

Telah dipertahankan di depan tim penguji ujian sarjana tugas akhir

Program Studi Teknik Informatika

Universitas Islam Sultan Agung

Pada tanggal : 24 Februari 2025

TIM PENGUJI UJIAN SARJANA

Dedy Kurniadi ST, M.Kom
NIDN. 0622058802
(Ketua Penguji) ..... 26 / 02 / 2024

Sam Farisa Chaerul Haviana, ST, M.Kom.
NIDN. 0628028602
(Anggota Penguji) ..... 6 / 03 / 2024

Ir. Sri Mulyono
NIDN.0626066601
(Pembimbing)

Semarang, ...

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	: Naela Mukaromah
NIM	: 32602100096
Judul Tugas Akhir	: **IMPLEMENTASI *STABLE DIFFUSION* DAN *FINE-TUNING* LORA UNTUK PEMBUATAN LOGO**

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 dikemudian 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

Yang Menyatakan

Naela Mukaromah

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Naela Mukaromah

NIM : 32602100096

Program Studi : Teknik Informatika

Fakultas : Teknologi Industri

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : **IMPLEMENTASI *STABLE DIFFUSION* DAN *FINE-TUNING* LORA UNTUK PEMBUATAN LOGO**

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

Yang Menyatakan

Naela Mukaromah

KATA PENGANTAR

Puji syukur peneliti ucapkan kepada Allah swt, alhamdulillah atas rahmat dan karuniaNya peneliti dapat menyelesaikan tugas akhir dengan judul "Implementasi *Stable diffusion* dan *Fine-tuning* LoRA Untuk Pembuatan Logo" untuk memenuhi salah satu syarat menyelesaikan studi sarjana pada program studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.

Dalam proses penyusunan Tugas Akhir ini, peneliti mendapat banyak sekali dukungan, bimbingan, serta motivasi dari berbagai pihak. Oleh karena itu, peneliti ingin mengucapkan rasa terima kasih yang sebesar-besarnya kepada :

1. Prof. Dr. H. Gunarto, SH., M.Hum, Selaku Rektor Universitas Islam Sultan Agung, yang telah memberikan kesempatan kepada peneliti untuk belajar di universitas ini.
2. Dr. Ir. Hj.Novi Marlyana, S.T., M.T, Selaku Dekan Fakultas Teknologi Industri yang telah memberikan kesempatan untuk belajar di fakultas ini
3. Moch Taufik, S.T., M.Kom, selaku Ketua Program Studi S1 Teknik Informatika yang telah memberikan dukungan akademik selama proses penelitian berlangsung.
4. Ir. Sri Mulyono, selaku dosen pembimbing yang telah memberikan bimbingan, arahan, serta masukan yang sangat berharga dalam penyusunan tugas akhir ini.
5. Orang tua penulis yang telah mengizinkan untuk menyelesaikan laporan ini serta selalu memberikan dukungan dan doa.

Dengan segala kerendahan hati, penulis menyadari masih terdapat kekurangan dari segi kualitas atau kuantitas maupun dari ilmu pengetahuan dalam penyuunan laporan. Sehingga penulis mengharapkan adanya kritik dan saran yang bersifat membangun.

Semarang, 20 Februari 2025

Naela Mukaromah

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

ABSTRAK

Pada pertumbuhan ekonomi di indonesia, UMKM memiliki kontribusi tinggi dalam meningkatkan tingkat pertumbuhan ekonomi. Pelaku UMKM masih memiliki keterbatasan untuk memperkuat identitas merek mereka. Salah satu elemen penting dalam identitas atau *branding* UMKM adalah logo, namun dalam proses pembuatannya membutuhkan biaya dan waktu yang lumayan besar. Penelitian ini bertujuan untuk mengembangkan sistem otomatis yang mampu menghasilkan ogo berdasarkan deskripsi teks menggunakan *Stable diffusion* dan *Low Rank Adaptation* (LoRA) *fine-tuning*. Evaluasi kinerja menggunakan evaluasi SSIM (*Structural Similarity Index Measure*) memperoleh *score* rata-rata 0,536 yang menunjukkan bahwa gambar cukup memiliki kesamaan dengan data uji. Sementara itu, evaluasi menggunakan CLIP-MMD (Contrastive Language-Image Pre*training* - *Max*imum Mean Discrepancy) memperoleh *score* 1,66 untuk data uji dan 1.60 untuk gambar yang dihasilkan oleh model baru. Dimana gambar dan teks yang dihasilkan oleh model baru menunjukkan kesesuaian lebih baik antara gambar dan deskripsi teks dibanding data uji. Hasil penelitian ini menunjukkan bahwa model *Stable diffusion* versi 1.5 dan LoRA *fine-tuning* dapat digunakan untuk sarana eksplorasi variasi logo dan pembuatan logo secara otomatis. Namun, masih diperlukan peningkatan dalam kualitas visual dan akurasi teks pada logo yang dihasilkan.

Kata kunci : *Stable diffusion, Low Rank Adaptation, text-to-image, Logo.*

In Indonesia's economic growth, UMKM (Micro, Small, and Medium Enterprises) contribute significantly to increasing economic development. However, many UMKM face challenges in strengthening their brand identity. One of the key elements of branding is the logo, but its creation requires considerable time and cost. This study aims to develop an automated system capable of generating logos based on text descriptions using Stable diffusion 1.5 and LoRA (Low-Rank Adaptation) fine-tuning. The system's performance was evaluated using SSIM (Structural Similarity Index Measure), which yielded an average score of 0.536, indicating a moderate similarity between the generate d images and test data. Additionally, CLIP-MMD (Contrastive Language-Image Pretraining - Maximum Mean Discrepancy) produced scores of 1.66 for test data and 1.60 for generate d images, demonstrating a better alignment between the generate d images and text descriptions. The findings indicate that Stable diffusion with LoRA fine-tuning can be utilized for automated logo generation. However, further improvements are required to enhance the visual quality and text accuracy of the generate d logos.

Keyword : Stable diffusion, Low Rank Adaptation, text-to-image, Logo.

BAB I
PENDAHULUAN

1.1 Latar Belakang

Pendapatan nasional negara salah satunya dipengaruhi oleh pertumbuhan ekonomi yang menjadi salah satu indikator keberhasilan suatu negara. Produk domestik bruto (PDB) sering digunakan untuk mengukur pertumbuhan ekonomi. Semakin tinggi PDB maka semakin bagus perekonomian negara tersebut. Usaha Mikro, Kecil, dan Menengah (UMKM) sangat berpengaruh dalam pertumbuhan ekonomi di indonesia dan diakui mempunyai peran penting dalam kontribusi pertumbuhan ekonomi. Dibuktikan dengan perkembangan UMKM dalam kurun waktu 10 tahun yang selalu meningkat. Selain itu, UMKM turut berperan dalam meratakan pendapatan dan meningkatkan kesejahteraan masyarakat di berbagai daerah, terutama di wilayah-wilayah yang masih memiliki tingkat kemiskinan yang tinggi (Halim, 2020).

Walaupun UMKM memiliki kontribusi tinggi dalam menyokong pertumbuhan ekonomi di indonesia, tentunya tetap ada tantangan yang dihadapi oleh pelaku UMKM. Tantangan yang sering dihadapi dibeberapa UMKM adalah keuangan terbatas dan kurangnya akses teknologi modern (Aftitah dkk., 2025). Keterbatasan finansial ini seringkali menjadi penghalang bagi pelaku UMKM untuk mengembangkan dan memperkuat identitas merek mereka. Saat ini, kondisi produk UMKM di indonesia dalam memperkuat diri mereka atau *branding* masih beragam. Terdapat banyak UMKM yang berkualitas namun belum kuat dalam *branding*. Padahal potensi UMKM dalam mengembangkan usahanya melalui *branding* sangat memungkinkan, karena banyak bisnis kecil UMKM yang berubah menjadi brand besar karena bandingnya yang kuat. Meskipun terdapat beberapa UMKM yang berhasil dalam *branding* yang dilakukannya sendiri, ternyata masih banyak UMKM yang belum bisa melakuakan *branding* sendiri (Faiz Muntazori & Listya, 2021). Maka langkah awal dan sederhana untuk memperkuat identitas UMKM adalah pembuatan logo, dimana logo berfungsi untuk meningkatkan daya saing dan mengenalkan produk ke pasar yang lebih luas. Selain itu, banyak UMKM yang kesulitan menentukan elemen visual yang paling cocok untuk merek mereka. Oleh

karena itu, diperlukan solusi yang memungkinkan UMKM untuk mengeksplorasi berbagai konsep desain logo sebelum memilih dan menggunakan desain akhir yang sesuai dengan identitas bisnis mereka **.**

Untuk menjawab tantangan diatas, salah satu teknologi yang menjadi tren saat ini dan diprediksi akan bertahan lama adalah teknologi AI. Teknologi AI telah memberikan perubahan yang cukup besar (Sony Maulana dkk., 2023) (Arly dkk., 2023). Teknologi ini menawarkan potensi besar dalam mengoptimalkan *branding* UMKM khususnya pada identitas visual. Dengan AI dapat mengatasi kendala biaya dan sumber daya yang sering kali dihadapi dalam pembuatan identitas visual. AI memungkinkan pelaku UMKM untuk memproduksi identitas visual logo tanpa vendor. Hal ini diperkuat oleh penelitian yang dilakukan oleh Cueves dkk dengan hasil penelitian bahwa generator logo menggunakan AI dinilai efektif (Cuevas dkk., 2023).

Penelitian ini mengembangkan sistem yang mampu menghasilkan logo berdasarkan deskripsi teks menggunakan Stable Diffusion versi 1.5 dan teknik *fine-tuning* LoRA (*Low-Rank Adaptation*). Sistem ini tidak hanya bertujuan sebagai alat generatif, tetapi juga sebagai sarana eksplorasi dan pendukung keputusan bagi pengguna dalam mengeksplorasi berbagai variasi desain logo sebelum memilih logo akhir yang sesuai dengan identitas bisnis mereka. Dengan menyediakan opsi visual yang beragam, sistem ini membantu pengguna untuk lebih memahami bagaimana logo mereka akan terlihat dalam berbagai bentuk dan gaya. Selain itu, sistem ini juga dapat menyesuaikan prompt atau deskripsi teks untuk mendapatkan hasil yang lebih sesuai dengan preferensi mereka. Dengan demikian, sistem ini tidak hanya berfungsi sebagai alat generatif tetapi juga sebagai alat bantu dalam proses pengambilan keputusan desain.

1.2 Perumusan Masalah

Bagaimana cara mengimplementasikan *Fine-tuning Low Rank Adaption* dan *Stable diffusion* untuk menghasilkan sistem yang berfungsi sebagai sarana eksplorasi dan pendukung keputusan dalam pembuatan logo berbasis deskripsi teks?

1.3 Pembatasan Masalah

Dalam penelitian ini terdapat beberapa batasan masalah, diantaranya sebagai berikut :

1. Penelitian ini hanya menggunakan *Stable diffusion* versi 1.5 dengan pendekatan *fine-tuning Low-Rank Adaptation* (LoRA) untuk menghasilkan logo.
2. Logo yang dihasilkan dalam penelitian ini difokuskan pada bidang makanan dan minuman, sehingga tidak mencakup kategori lain di luar konteks tersebut.
3. *Output* yang dihasilkan berupa gambar statis berdasarkan deskripsi teks yang diberikan, tanpa animasi atau elemen interaktif.
4. Sistem yang dikembangkan tidak dirancang untuk menghasilkan gambar dengan tingkat detail yang sangat tinggi atau kompleks, sehingga kualitas gambar bergantung pada model yang digunakan.
5. Pengguna tidak diberikan fitur untuk mengedit gambar secara langsung dalam sistem, sehingga hasil akhir bergantung pada *generate* otomatis dari model.
6. Sistem hanya menerima *input* deskripsi dalam bahasa Inggris, sehingga pengguna harus memasukkan teks dalam bahasa Inggris agar dapat diproses dengan baik.
7. Sistem hanya dapat menghasilkan logo berdasarkan deskripsi teks yang diberikan tanpa mempertimbangkan makna filosofis yang mendalam dari logo tersebu

1.4 Tujuan

Penelitian ini bertujuan untuk mengembangkan sistem yang dapat digunakan sebagai sarana eksplorasi dan pendukung keputusan dalam pembuatan logo berdasarkan deskripsi teks menggunakan Stable Diffusion versi 1.5 dan *fine-tuning Low Rank Adaption* (LoRA). Dengan adanya sistem ini, pengguna dapat memperoleh berbagai alternatif desain logo yang dapat digunakan sebagai referensi sebelum menentukan desain akhir.

1.5 Manfaat

Manfaat dari penelian tugas akhir ini adalah :

1. Mempermudah pelaku UMKM dalam proses pembuatan logo sebagai *branding* dari produknya, sehingga dapat membantu meningkatkan identitas visual usaha dengan cara yang terjangkau.
2. Meningkatkan pengetahuan dan sebagai referensi untuk penelitian selanjutnya terkait kecerdasan buatan, khususnya dalam penggunaan model generatif *Stable diffusion* dan LoRA *fine-tuning*.

1.6 Sistematika Penulisan

Berikut sistematika penulisan yang digunakan penulis dalam menyusun laporan penelitian tugas akhir :

BAB 1 : PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang pemilihan judul tugas akhir "Implementasi *Stable diffusion* dan *Fine-tuning* LoRA Untuk Pembuatan Logo", rumusan masalah, pembatasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB 2 : TINJAUAN PUSTAKA DAN DASAR TEORI

Bab ini memuat dasar teori dan penelitian terdahulu yang berfungsi sebagai sumber dan rujukan penelitian dalam pengembangan sistem, serta membantu penulis dalam memahami teori yang berkaitan.

BAB 3 : METODE PENELITIAN

Bab ini menjelaskan tentang proses tahapan- tahapan penelitian dimulai dari studi literatur, persiapan dataset, *preprocessing* dataset, pengembangan model, serta pengujian dan evaluasi model.

BAB 4: HASIL PENELITIAN DAN IMPLEMENTASI SISTEM

Bab ini menjelaskan hasil dari penelitian yang telah dilakukan, yakni hasil dari proses pembuatan sistem yang telah dibuat.

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

Penelitian yang berjudul "*TypeDance: Creating Semantic Typographic Logos from Image through Personalized Generation*" melakukan pembuatan sistem berbasis AI untuk membuat logo tipografi semantik. Sistem ini menggunakan generatif AI *diffusion model* dan *Contrastive Languange Image Pre-training* CLIP untuk menggabungkan elemen gambar dan menghasilkan logo. *Diffusion model* digunkan untuk menghasilkan gambar penggabungan antara elemen huruf dan gambar, serta mendukung penggabungan pada berbagai tingkat kedetailan huruf. Sedangkan CLIP digunakan untuk menyelaraskan teks, gambar, dan hasil desain, serta mengevaluasi seberapa sesuai *Output* dan *input* teks. Sistem yang dibuat disebut Typedance, dimana elemen huruf dan gambar digabungkan menjadikannya desain yang bermakna dan estetis. Hasil penelitian ini menunjukkan sistem yang dibuat dapat membantu desainer untuk menciptakan logo yang menggabungkan elemen tipografi dengan elemen gambar (Xiao dkk., 2024).

Penelitian yang berjudul "*Text-to-Image Generation Using Deep Learning*" membahas teknologi untuk menghasilkan gambar dari deskripsi teks menggunakan model deep learning. Salah satu pendekatan yang digunakan adalah Recurrent *Convolutional Generative Adversarial Network* (RC-GAN). Teknologi ini memungkinkan komputer mengubah deskripsi teks menjadi gambar visual yang sesuai, seperti menciptakan foto bunga dari deskripsi teks. Model ini dilatih dengan dataset gambar bunga dan deskripsi teks yang relevan. RC-GAN terdiri dari dua bagian utama, yaitu generator yang membuat gambar dari teks dan discriminator yang memeriksa apakah gambar yang dihasilkan realistis. Eksperimen menunjukkan bahwa model ini dapat menghasilkan gambar dengan kualitas yang baik, mengungguli metode sebelumnya dalam beberapa metrik seperti Inception *Score* (IS) dan PSNR (*Peak Signal-to-Noise Ratio*) (Ramzan dkk., 2022). Berdasarkan penelitian tersebut memungkinkan untuk pembuatan logo dari deskripsi teks, yang termasuk dari text to image.

Adapun penelitian yang membahas penyesuaian (*fine-tuning*) model *Stable diffusion* guna menghasilkan ikon bergaya tertentu untuk kebutuhan komersial, seperti ikon sekrup dan kabinet dapur. Penulis bereksperimen dengan berbagai cara pelatihan dan ukuran teks deskripsi, seperti teks pendek, teks panjang, dan penggunaan gambar referensi. Hasil yang didapatkan dalam penelitian ini model *Stable diffusion* dapat disesuaikan untuk menghasilkan ikon dengan gaya tertentu dan berkualitas baik. Model yang dilatih menggunakan deskripsi teks pendek dan gambar referensi menghasilkan gambar yang lebih akurat secara teknis, namun hasilnya tidak selalu sesuai dengan ekspektasi manusia dalam hal detail dan konsistensi gaya. Sebaliknya, model dengan deskripsi teks panjang lebih mampu menghasilkan ikon yang sesuai dengan deskripsi dan lebih mendekati gaya yang diinginkan, meskipun skornya sedikit lebih rendah (Sultan dkk., 2024).

Dalam penerapannya, *Low Rank Adaption* (LoRA) muncul sebagai salah satu teknik *fine-tuning* yang dapat digunakan untuk menghasilkan logo dari teks. Salah satu studi yang menunjukkan potensi ini dilakukan oleh Mahendra (2022), yang berhasil meningkatkan efisiensi model bahasa besar LlaMA 2-7B dengan teknik LoRA. Penulis memulai dengan mengumpulkan data berupa 60.500 artikel tentang kesehatan, pendidikan, sejarah, dan teknologi melalui teknik web scraping. Setelah itu, data tersebut diproses agar bersih dari elemen yang tidak penting, memiliki panjang teks yang seragam, dan diubah menjadi bentuk numerik supaya bisa dipahami oleh komputer. Kemudian penulis mengaplikasikan LoRA untuk mengurangi ukuran parameter model sebesar empat kali lipat, sehingga model menjadi lebih ringan dan lebih cepat digunakan tanpa mengurangi kualitasnya. Hasilnya, model yang telah dioptimalkan dengan LoRA hanya membutuhkan memori GPU sebesar 4,74 GB, jauh lebih kecil dibandingkan model aslinya yang memerlukan 23,1 GB. Selain itu, model ini juga mampu memproses data hingga 8 kali lebih cepat, sehingga sangat hemat waktu dan daya komputasi.

Kemudian pada penelitian yang berjudul "*A LoRA is Worth a Thousand Pictures*", metode yang digunakan adalah LoRA *fine-tuning* dan *Stable diffusion* 1.5 untuk membahas hubungan antara bobot LoRA dan representasi gaya seni.

Dataset yang digunakan berupa seni kecil, seperti karya seni dari genre seperti Art Nouveau, Baroque, Post-Impressionism, Renaissance, dan Ukiyo-e. *Fine-tuning* dilakukan pada u-net bagian self-attention, cross-attention, dan feed-*forward* neural network. Hasil penelitian menunjukkan bahwa LoRA dinilai merepresentasikan gaya seni lebih baik dibanding metode CLIP atau DINO. Selain itu, hasil clustering dan retrival lebih presisi untuk pengelompokkan seni(C. Liu dkk., 2024) .

Berdasarkan beberapa penelitian di atas, penulis dapat menyimpulkan bahwa *Stable diffusion* dan teknik *fine-tuning Low-Rank Adaptation* (LoRA) dapat dikombinasikan untuk membantu pembuatan logo dari teks. *Stable diffusion* berperan sebagai model generatif yang mampu mengubah deskripsi teks menjadi gambar dengan detail tinggi dan kualitas visual yang baik, sehingga memungkinkan pembuatan logo sesuai dengan deskripsi teks. LoRA berperan dalam proses pelatihan model. Dengan LoRA, model dapat dengan mudah disesuaikan untuk menambahkan gaya visual tertentu atau memenuhi kebutuhan spesifik logo tanpa harus melatih ulang seluruh model.

2.2 Dasar Teori

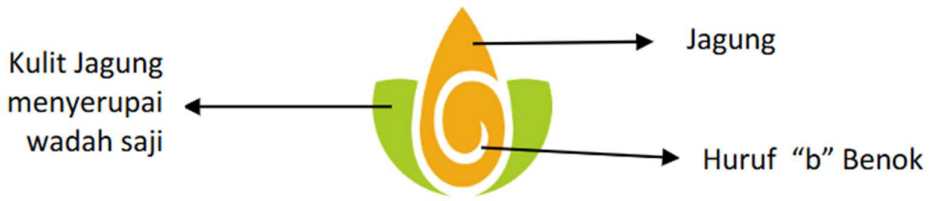
2.2.1 Logo

Logo merupakan salah satu bentuk identitas visual dari produk yang berbentuk ideogram, simbol, emblem, ikon, tanda yang digunakan sebagai representasi dari brand (Ainun dkk., 2023). Identitas visual produk bermanfaat untuk membedakan antara merk satu dengan lainnya. Logo digunakan untuk alat komuniksasi yang membentuk identitas dari suatu perusahaan, lembaga, produk dan lain-lain (Aulia dkk., 2021). Dengan adanya logo, orang akan mudah mengenali dan mengingat brand yang ditawarkan. Dengan memiliki logo yang khas dan konsisten, sebuah entitas dapat lebih mudah dikenali dan diingat oleh konsumen, sehingga membedakannya dari pesaing. Logo yang dirancang dengan baik mencerminkan nilai, visi, dan misi yang diusung oleh entitas tersebut, serta menjadi representasi grafis yang unik dalam menyampaikan pesan kepada audiens.

Selain itu, logo juga berperan dalam membangun citra merek yang profesional dan terpercaya, yang pada gilirannya dapat meningkatkan loyalitas

pelanggan dan memperluas jangkauan pasar. Oleh karena itu, penting bagi setiap entitas untuk merancang logo yang efektif dan sesuai dengan identitas yang ingin disampaikan, serta memastikan penggunaannya konsisten di berbagai media dan platform.

Bagi UMKM, logo memiliki peran yang lebih krusial karena menjadi elemen utama dalam membangun citra merek di tengah keterbatasan anggaran pemasaran dan sumber daya yang dimiliki. Pembuatan logo juga memberikan fleksibilitas bagi UMKM untuk menyesuaikan desain sesuai dengan karakteristik dan nilai bisnis yang ingin ditonjolkan. Selain itu, dengan strategi *branding* pembuatan identitas logo dapat meningkatkan penjualan dan daya tarik masyarakat (Zahara, 2024).



Gambar 2. 1 Logo UMKM Benok

Gambar diatas merupakan salah satu contoh logo dari UMKM. UMKM tersebut memiliki produk dengan bahan dasar dari jagung. Terlihat dari logo yang dimiliki merepresentasikan bahwa terdapat jagung yang menjadi bahannya (Wahmuda & Junaidi Hidayat, 2020).

2.2.2 *Text to Image*

Text to Image merupakan metode yang mengubah deskripsi teks menjadi representasi visual, yang digunakan untuk menghasilkan gambar berdasarkan deskripsi teks yang diberikan. Model ini mengeneralisasi kata-kata atau kalimat dengan elemen-elemen visual yang relevan. Teknologi ini memanfaatkan model deep learning yang nampu memahami dan menerjemahkan teks menjadi representasi visual yang sesuai dengan tingkat akurasi yang tinggi (Berrahal & Azizi, 2022).

Beberapa model *Text to Image* yang sering digunakan dalam proses generalisasi adalah sebagai berikut :

a. *Generative Adversarial Networks* (GANs), merupakan model yang bekerja dengan dua jaringan saraf, yaitu generator yang menciptakan gambar, dan discriminator yang menilai apakah gambar tersebut nyata atau tidak. GANs telah digunakan dalam berbagai aplikasi Text-to-Image, termasuk dalam pembuatan seni digital, desain produk, hingga rekonstruksi wajah berdasarkan deskripsi saksi dalam bidang forensik.
b. *Variational Autoencoders* (VAEs), merupakan model yang menggunakan teknik probabilistik untuk menghasilkan gambar dengan mempertimbangkan variasi dalam data. VAEs sering digunakan untuk menciptakan gambar yang lebih fleksibel dan dapat menghasilkan variasi visual yang luas dari teks yang sama.
c. *Diffusion Models*, merupakan model generatif yang bekerja dengan prinsip menyusun gambar secara bertahap dari *noise* acak hingga membentuk gambar yang sesuai dengan teks. Teknologi ini semakin populer karena mampu menghasilkan gambar dengan resolusi tinggi dan detail yang lebih kompleks dibandingkan model lainnya.

Masing-masing model tersebut memiliki peran penting dalam menghasilkna gambar yang realistis dan mampu menggeneralisasi teks ke dalam gambar lebih mendetail. Hal ini menjadi salah satu alasan penggunaan dari model AI tersebut. Alasan penggunaan AI ini adalah karena menghasilkan gambar yang lebih realistis dan akurat (Ramzan dkk., 2022).

Teknologi *Text-to-Image* kini telah diterapkan di berbagai industri, mulai dari seni digital, desain visual, industri game, hingga dunia pemasaran dan periklanan. Dalam konteks ini *Text to Image* berfungsi sebagai salah satu alat untuk membantu pelaku UMKM khususnya dalam memperkuat identitas, yaitu membuat logo untuk merek produk yang dimiliki.

2.2.3 *Fine-tuning*

Dalam machine learning, *fine-tuning* adalah teknik yang digunakan untuk memperbaiki kinerja model dengan cara bobot dan bias disesuaikan kembali pada model yang telah dilatih sebelumnya pada tugaas baru. Pada teknik ini biasanya digunakan pada model yang dilatih pada dataset besar kemudian digunakan pada tugas pengenalan citra atau teks yang lebih kecil. Tahapan *fine-tuning* dimulai dari pret-*train*ed model atau mengambil model yang sudah dilatih pada dataset besar sebagai model dasar. Selanjutnya, lapisan terakhir model diganti untuk menyesuaikannya dengan tugas baru. Lapisan terakhir adalah lapisan keluaran atau kelas. Melatih ulang lapisan terakhir selama proses *fine-tuning* digunakan untuk memastikan bahwa lapisan lain tetap menggunakan bobot dan bias yang sudah diatur sebelumnya. (Pradana dkk., 2024).

Fine-tuning bertujuan untuk melatih pre-*train*ed model yang berada pada top layer dan berfungsi untuk mengenali feature dari dataset yang ada (Sasongko dkk., 2023). *Fine-tuning* memiliki kemampuan untuk mengurangi jumlah daya komputasi yang mahal dan data berlabel yang diperlukan untuk menghasilkan model besar yang dapat disesuaikan dengan berbagai kasus penggunaan. Ada beberapa macam teknik *Fine-tuning*, salah satu nya adalah Low Rank Adaption.

Pada penelitian ini, *fine-tuning* digunakan untuk melatih model agar *Output* yang dihasilkan dapat berupa logo makanan. Teknik *fine-tuning* yang akan digunakan adalah teknik *Low Rank Adaptation* (LoRA).

2.2.4 LoRA (*Low Rank Adaption*)

Low Rank Adaptation (LoRA) adalah teknik untuk menyesuaikan model kecerdasan buatan (AI) yang sudah dilatih sebelumnya agar bisa digunakan untuk tugas atau bidang tertentu. LoRA bekerja dengan menambahkan komponen sederhana (matriks kecil) ke dalam model, sehingga model bisa belajar hal-hal baru tanpa perlu mengubah strukturnya secara besar-besaran. Reperesentasi LoRA dalam matimatis, sebagai berikut :

$$\Delta \mathcal{W} = A \,.\, B^T \tag{1}$$

Keterangan :

$\Delta \mathcal{W}$: Perubahan kecil yang akan kita tambahkan ke bagian tertentu dari model

A : Matriks dengan dimensi $d \times r$, dimana d adalah dimensi *input*, dan r adalah rank rendah yang ditentukan.

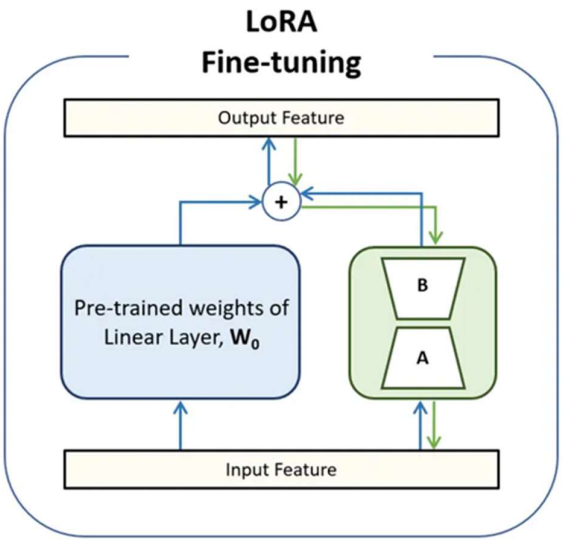
B : Matriks dengan dimensi $r \times d$, dimana r adalah rank rendah

$$W' = W + \Delta W \quad (2)$$

W : bagian model yang tidak berubah

W' : model yang sudah diperbarui dengan $\Delta \mathcal{W}$

Teknik ini memungkinkan pelatihan model menjadi lebih cepat dan hemat, karena hanya bagian kecil dari model yang dilatih ulang. Sementara itu, bagian utama model tetap tidak berubah, sehingga hasil sebelumnya tidak hilang. Dengan cara ini, LoRA dapat mengurangi kebutuhan perangkat keras (seperti memori atau GPU) hingga tiga kali lipat dibandingkan metode pelatihan biasa. Selain itu, LoRA menghasilkan kualitas yang sebanding dengan pelatihan penuh, tetapi dengan biaya dan waktu yang lebih efisien. Dengan menerapkan struktur low-rank pada pembaruan matriks, LoRA juga mampu menghasilkan hasil yang lebih baik, sekaligus mempertahankan atau bahkan menghasilkan kualitas yang setara dengan *fine-tuning* penuh (Hu dkk., 2022). Berikut adalah gambar dari arsitektur LoRA.



Gambar 2. 2 Arsitektur LoRA

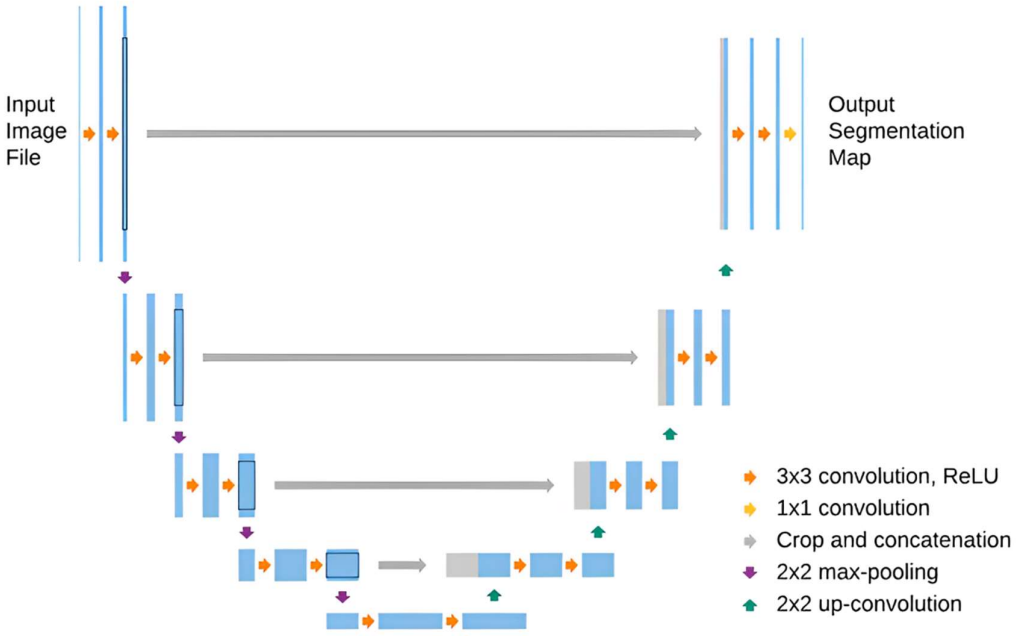
Gambar diatas menunjukkan konsep LoRA *Fine-tuning*, yaitu teknik untuk menyesuaikan model besar tanpa mengubah semua bobot aslinya. Dimana W_0 adalah model dasar, kemudian ditambahkan input feature berupa matriks kecil A dan B. Kedua matriks ini berfungsi untuk menangkap perubahan kecil yang diperlukan selama proses fine-tuning tanpa perlu mengubah keseluruhan bobot model. Fitur masukan kemudian dimodifikasi oleh matriks A dan B, lalu hasilnya digabungkan kembali dengan bobot asli dan menghasilkan output feature berupa model bobot yang sudah disesuaikan.

2.2.5 U-Net

U-Net adalah jenis model komputer yang sering digunakan untuk mengolah gambar, seperti membersihkan gambar dari gangguan (*noise*) atau membagi gambar menjadi bagian-bagian tertentu. Dikenalkan oleh Olaf Ronneberger dan timnya, khususnya untuk aplikasi medis seperti segmentasi organ pada citra MRI atau CT scan. Namun, U-Net juga banyak digunakan dalam berbagai, salah satunya untuk pengolahan gambar. Bentuknya seperti huruf "U", dengan bagian pertama mengecilkan gambar untuk memahami informasi penting, dan bagian kedua memperbesar gambar kembali sambil mempertahankan detail yang penting. Arsitektur U-net yang menyerupai huruf U menjadikannya istimewa, karena

memungkinkan jaringan untuk mengintergrasikan informasi konteks global dan detail lokal secara efisien (Mei, 2024).

Pada stable difussion versi 1.5 U-Net bekerja untuk memproses *noise* dalam proses denoising ketika menghasilkan gambar. U-Net akan memprediksi *noise* yang harus dihilangkan pada setiap prosesnya, hal ini memungkinkan gambar yang dihasilkan akurat dari random *noise* berdasarkan teks deskripsi.



Gambar 2. 3 Arsitektur U-Net

Gambar diatas merupakan arsitektur U-Net, yang merupakan model *Convolutional Neural Network* (CNN) yang digunakan untuk *image segmentation*. Model ini memiliki struktur berbentuk "U" dan terdiri dari dua jalur utama yaitu *contracting path* (*Encoder*) yang berada dibagian kiri atau *downsampling* dan *expanding path (Decoder)* yang berada dibagian kanan *upsampling.* Lebih jelasnya sebagai berikut :

1. *Input image file*

   Gambar *input* masuk ke model dalam bentuk tensor. Tensor ini kemudian melewati serangkaian konvolusi dan downsampling untuk mengekstrak fitur penting dari gambar.

2. *Contracting path (encoder) downsampling*

   Bagian ini bertujuan untuk mengekstrak fitur spasial dari gambar dengan menggunakan konvolusi dan *pooling***,** sehingga mendapatkan representasi yang lebih abstrak dari gambar. Langkah dalam *contracting path* :

   a. 3×3 *Convolution* + ReLU *Activation*

      Setiap blok melakukan dua kali konvolusi 3×3 untuk mengekstrak fitur dari gambar. Aktivasi ReLU (*Rectified Linear Unit*) digunakan untuk meningkatkan non-linearitas model.

   b. 2×2 *Max Pooling*

      Setelah konvolusi, dilakukan *max pooling* 2×2, yang mengurangi resolusi fitur map menjadi separuhnya. Tujuannya adalah untuk mengurangi ukuran gambar dan menangkap informasi global**.**

   c. Pengulangan langkah a & b

      Proses konvolusi dan max pooling ini dilakukan berulang kali**,** sehingga ukuran gambar semakin kecil tetapi fitur yang didapat semakin kompleks. Setiap kali terjadi downsampling, jumlah *channel* meningkat**,** misalnya dari 64 → 128 → 256 → 512.

3. *Bottleneck* (bagian tengah U-Net)

   Merupakan lapisan terdalam dari jaringan, yang bertindak sebagai *bottleneck* untuk menyaring informasi penting sebelum dilakukan upsampling. Sama seperti sebelumnya, terdapat dua lapisan konvolusi 3×3 + ReLU, tetapi tanpa *pooling*.

4. *Expanding path (decoder) upsampling*

   Bagian ini bertujuan untuk meregenerasi gambar dengan meningkatkan resolusi fitur map secara bertahap menggunakan *transposed convolution (up-convolution)*

5. *Output segmentation map*

   Pada tahap terakhir, dilakukan 1×1 *convolution* untuk mengubah jumlah *channel* menjadi jumlah kelas yang diinginkan. Hasil akhirnya adalah *segmentation map*, yaitu gambar yang sudah diberi label sesuai dengan kategori yang telah ditentukan.

2.2.6 *Diffusion Model*

Diffusion Models adalah model AI yang melakukan proses generalisasi untuk menghasilkan gambar dari data yang bersifat acak menjadi gambar yang lebih jelas. Cara kerja *Diffusion Models* melalui proses pembalikan *noise* dari distribusi Gaussian yang terdiri dari dua tahap, *forward process* (menambahkan *noise*) dan reverse *process* (mengurangi *noise*). Metode ini dapat menghasilkan citra yang baik dengan kompleksitas komputasi lebih rendah (Huang dkk., 2024). Berikut representasi rumus dari diffusion model:

Forward process

$$x_t = \sqrt{\overline{\alpha_t}}\, x_0 + \sqrt{1 - \overline{\alpha_t}}\, \in \quad (3)$$

Keterangan :

x_0 : data asli (gambar)

x_t : data setelah langkah t

$\overline{\alpha_t}$: skala data asli pada langkah t

$\in$: *noise* Gaussian acak

Reverse process

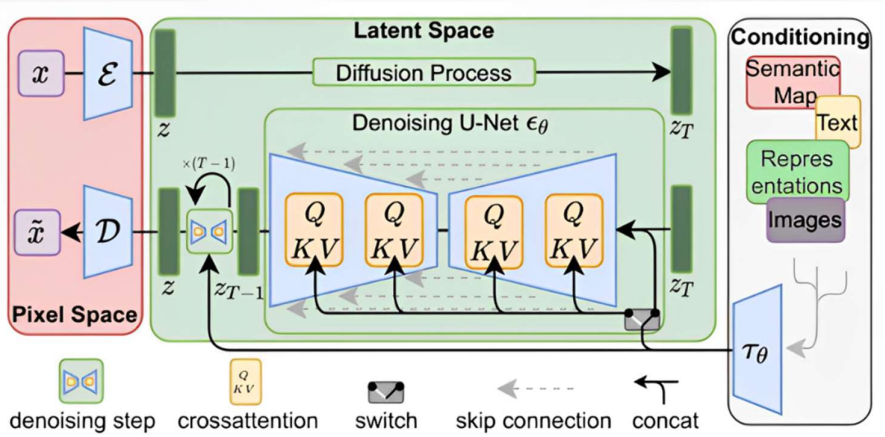
$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \sqrt{1 - \alpha_t}\, \in_\theta (x_t, t) \right) + \text{tambahan } noise \quad (4)$$

Keterangan :

x_{t-1} : data pada langkah sebelumnya

(x_t, t) : *noise* yang di prediksi

Diffusion model dalam deep learning merupakan model generatif yang memanfaatkan proses difusi untuk memodelkan distribusi data yang kompleks. Model ini terkenal karena dapat menghasilkan sampel berkualitas tinggi dan melakukan tugas-tugas seperti sintesis gambar dan penghilangan *noise* (Naseem, 2024).



Gambar 2. 4 Arsitektur diffusion model

Gambar diatas menunjukkan bagaimana Stable Diffusion bekerja untuk menghasilkan gambar dari teks atau kondisi lainnya. Berikut adalah penjelasan bagian-bagian utamanya:

1. *Pixel Space*

   Merupakan tahap awal dan akhir dalam proses generasi gambar. Gambar asli dikonversi ke bentuk lebih ringkas (kode laten) menggunakan encoder. Setelah proses selesai, hasil akhirnya dikonversi kembali ke gambar menggunakan *decoder*.

   $x =$ input gambar dalam bentuk resolusi

   ε = encoder, mengubah gambar asli x menjadi representasi z

   $\mathcal{D}$ = decoder, mengubah kembali representasi latent $\bar{z}$

2. Laten *Space*

   Model bekerja di ruang laten untuk membuat proses lebih efisien dibandingkan langsung di ruang piksel. Proses difusi terjadi di sini, di mana gambar secara bertahap diberi noise hingga menjadi acak sepenuhnya. Model kemudian membalikkan proses ini dengan menghapus noise secara bertahap menggunakan jaringan Denoising U-Net.

   z = laten

   z_T = laten yang sudah ditambah noise

   $\bar{z}$ = hasil denoising

3. *Diffusion process*

Menambahkan noise secara bertahap ke dalam representasi laten z hingga menjadi distribusi normal z_T

4. *Denoising* U-Net

   Menghilangkan noise dari z_T secara bertahap hingga mendapatkan kembali representsi laten yang lebih bersih z_T. Mekanisme denoising :

   - *Denoising Step***:** Ditunjukkan dengan ikon dua panah berlawanan dalam kotak kecil.
   - *Skip Connection* : Garis putus-putus untuk mempertahankan informasi resolusi tinggi dari lapisan awal ke lapisan akhir.
   - *Concat (Concatenation)* **:** Gabungan informasi dari berbagai tahap proses.

5. Mekanisme *Cross-Attention*

   Berfungsi untuk menyesuaikan representasi laten berdasarkan input tambahan seperti teks atau gambar referensi. Berikut beberapa blok yang digunakan untuk memahami dan memproses kondisi tambahan.

   Blok *Q* = *Query*

   Blok *K* = *Key*

   Blok V = *Value*

6. *Conditioning*

   Model menerima berbagai kondisi tambahan dan dikombinasikan melalui mekanisme *cross-attention* untuk menghasilkan gambar yang sesuai dengan input. Beberapa kondisi tersebut sebagai berikut :

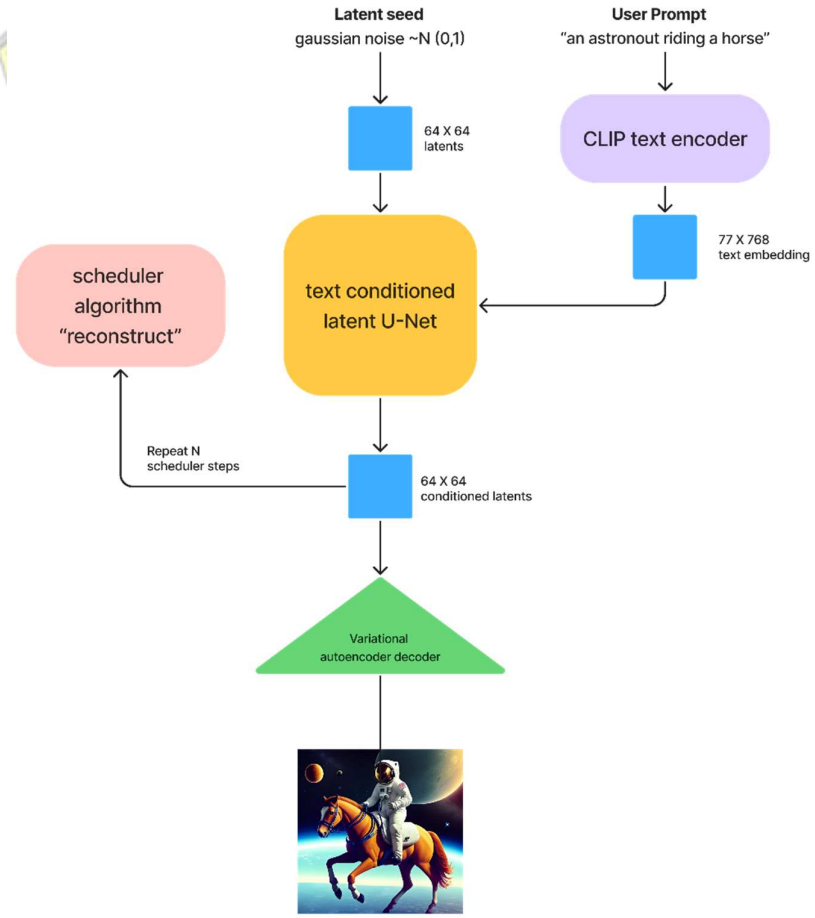
   - Teks (deskripsi gambar)
   - Peta Semantik (struktur tertentu dalam gambar)
   - Gambar (untuk melakukan penyempurnaan atau perubahan)
   - Representasi (seperti gaya tertentu)

2.2.7 *Stable diffusion* 1.5

Stable diffusion 1.5, yang merupakan bagian dari *Diffusion Models*, adalah model kecerdasan buatan yang dikembangkan oleh Stability AI menggunakan teknik *Latent Diffusion model* (LDM), dan memiliki kemampuan untuk mengkonversi teks menjadi gambar dengan baik (B. Liu dkk., 2024). Struktur

Stable diffusion diantaranya terdiri dari U-Net, CLIP *Text* Encoder, dan *Variational Autoencoder* (VAE). Cara kerja *Stable diffusion* 1.5 yaitu dimulai dengan random *noise*, kemudian melewati U-Net untuk dihilangkan *noise* nya secara bertahap dan mulai berbentuk gambar berdasarkan teks yang diberikan. Pada tahap ini model membutuhkan beberapa langkah untuk menghilangkan dan memperbaikin *noise*, sehingga mencapai hasil akhir yang diinginkan.

Model ini dipilih karena kemampuannya yang dapat menghasilkan gambar dan dapat dikombinasikan dengan teknik *fine-tuning* untuk mengubah gambar dengan gaya tertentu, sesuai dengan dataset yang ada. Terdapat penelitian yang melakukan beberapa teknik *fine-tuning* terhadap stable diffusion 1.5 dan menunjukkan fleksibilitas model dalam menyesuaikan dengan berbagai kebutuhan (Gambashidze dkk., 2024)



Gambar 2. 5 Alur kerja *Stable diffusion* 1.5

Ganbar diatas merupakan alur kerja dari *stable diffusion* 1.5 dalam menghasilkan gambar. Berikut penjelasan alur kerja nya :

6. *User input prompt*

   User memasukkan deskripsi teks yang ingin dimunculkan pada gambar. Pada alur diatas deskripsi tersebut adalah "*an astronout riding a horse*". Model akan memahami makna dari teks deskripsi agar dapat menghasilkan gambar yang sesuai. Teks tersebut kemudian dikodekan menggunakan *Contrastive Languange-Imange Pretraining* (CLIP).

7. *Encoding* teks dengan CLIP

   CLIP *text encoder* bertugas mengubah teks yang diberikan pengguna menjadi representasi numerik yang disebut text embedding**.** *Embedding* ini berbentuk matriks berukuran 77 × 768**,** yang berisi informasi semantik dari teks. *Embedding* ini nantinya akan digunakan sebagai kondisi tambahan dalam model untuk mengontrol bagaimana gambar dihasilkan, sehingga sesuai dengan deskripsi yang diminta.

8. Inisialisasi latent seed (Gaussian noise)

   Model memulai dengan membuat tensor laten berukuran 64x64 yang berisi noise gaussian N (0,1). Noise ini merupakan titik awal bagi model, yang nantinya akan dibersihkan secara bertahap untuk membentuk gambar yang bermakna. Representasi laten digunakan sebagai ganti gambar berukuran besar untuk menghemat memori dan meningkatkan efisiensi komputasi.

9. Proses *denoising Text Conditioned* Latent U-Net

   Text conditioned latent U-Net adalah model berbasis U-Net yang telah dilatih untuk menghilangkan noise dari latents secara bertahap. Pada tahap ini U-Net menggunakan 2 input utama yaitu latent yang masih ternoise dan *text embedding* dari CLIP yang membantu memahami gambar seperti apa yang harus dibuat. Mekanisme *cross-attention* memungkinkan model stable diffusion 1.5 menyesuaikan gambar berdasarkan teks, sehingga hasil akhirnya mencerminkan teks deskripsi.

10. *Scheduler Algorithm*

    Algoritma *scheduler* menentukan berapa banyak langkah yang dibutuhkan untuk menghilangkan *noise* dari latents. Proses ini dilakukan dalam ***N*** langkah iteratif, dengan setiap langkah menghapus sedikit noise sekaligus menyesuaikan bentuk gambar berdasarkan embedding teks. Semakin banyak langkah, semakin detail dan akurat gambar yang dihasilkan.

11. *Variational Autoencoder* (VAE) *Decoder*

    Setelah latents mencapai tahap akhir rekonstruksi, mereka masih dalam bentuk representasi laten berukuran 64 × 64**.** VAE *Decoder* bertugas mengonversi representasi laten ini kembali ke dalam gambar resolusi tinggi. Decoder ini memastikan bahwa gambar akhir memiliki kualitas yang baik tanpa mengurangi informasi yang penting.

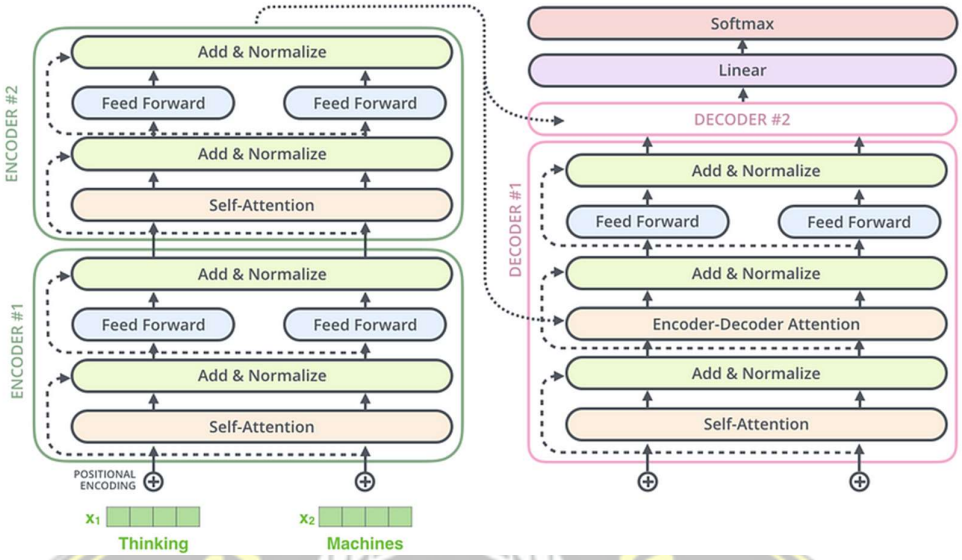
12. *Output*

    Setelah latents berhasil dikonversi, hasil akhirnya adalah gambar yang mencerminkan deskripsi teks yang diberikan. Pada alur diatas, model stable diffusion menghasilkan gambar seorang astronot yang sedang menunggang kuda**.** Gambar ini dibuat dengan mempertimbangkan informasi dari teks, serta melalui proses denoising yang mendalam.

2.2.8 Transformer

Transformer merupakan jenis model komputer yang digunakan untuk memahami dan memproses informasi, seperti teks, gambar, atau suara. Transformer memproses setiap komponen data secara bersamaan melalui mekanisme yang dikenal sebagai *self-attention*. *Self-attention* memungkinkan model untuk memberi perhatian pada setiap bagian data secara keseluruhan dan menghitung bagaimana hubungan antar bagian tersebut mempengaruhi pemahaman model terhadap informasi yang ada. Dengan memanfaatkan mekanisme *self-attention*, transformers dapat menangkap hubungan global dalam data, yang memungkinkan model ini untuk fokus pada elemen-elemen yang relevan dengan lebih mudah (Peebles & Xie, 2023). Untuk menangani informasi

yang sangat besar, kemampuan dari transformerd dinilai sangat bagus. Hal ini yang menjadikan transformer banyak digunakan oleh beberapa aplikasi.



Gambar 2. 6 Arsitektur transformer

Gambar diatas merupakan arsitektur dari transformer, yang terdiri dari encoder dan decoder yang digunakan untuk mengubah teks menjadi representasi vektor. Berikut adalah penjelasan dari arsitektur transformer :

1. Encoder

   Encoder merupakan yang mengubah input teks menjadi vektor numerik. Bagian ini terdiri dari beberapa lapisan atau *staked layers* yang diproses secara berulang. Beberapa elemen yang ada di *encoder* adalah sebagai berikut :

   a. *Positional encoding*

      Merupakan tempat pemberian informasi kata. Setiap kata dalam *input* diubah menjadi vektor numerik dan diberi nilai tambahan berdasarkan posisinya dalam kalimat. Bertujuan agar memahami urutan kata dalam teks.

   b. *Self attention layer*

      Berfungsi untuk menghitung hubungan antar kata dalam satu kalimat sehingga model dapat memahami konteksnya. Dengan cara setiap kata dalam input dikonversi menjadi *query* (Q), *key* (K), dan *value* (V) menggunakan 3 matriks terpisah. Kemudian *attention score* dihitung dengan rumus :

$$Attention\ (Q,K,V) = softmax\ \left(\frac{QK^T}{\sqrt{d_k}}\right) \quad (5)$$

Keterangan :

QK^T= menghitung kesamaan antara kata-kata

$\sqrt{d_k}$= normalisasi agar nilai tidak terlalu besar

$softmax$ = menghasilkan bobot perhatian yang lebih fokus.

Nilai akhir dari self-attention dikombinasikan kembali ke input awal. Kata yang memiliki hubungan erat akan memiliki bobot perhatian lebih tinggi.

c. *Add and Normalize*

Setelah self-attention, hasilnya ditambahkan kembali ke input asli melalui residual connection untuk mempertahankan informasi awal. *Layer Normalization* diterapkan untuk menstabilkan distribusi nilai, sehingga pelatihan lebih cepat dan stabil.

d. *Feed forward layer*

Mengubah representasi vektor hasil *self-attention* menjadi bentuk yang lebih abstrak.

e. *Output* dari *encoder*

Proses *self-attention, add and normalize*, dan *feed forward* diulang beberapa kali. Hasil akhir adalah representasi vektor yang menggambarkan seluruh kalimat *input*.

2. *Decoder*

Decoder merupakan bagian yang bekerja menggunakan representasi dari encoder untuk menghasilkan *output.* Beberapa elemen yang berada di decoder:

a. *Masked self attention layer*

Mirip dengan *self-attention* di encoder, tetapi memiliki masking yang bertujuan saat menghasilkan teks secara bertahap model tidak boleh melihat kata-kata di masa depan. Masking dilakukan dengan mengganti nilai yang tidak boleh dilihat dengan nilai sangat kecil (*negative infinity*) sebelum *softmax.*

b. *Encoder-decoder attention layer*

Berfungsi untuk menghubungkan informasi dari encoder ke *decoder* dengan cara menggunakan hasil dari *encoder* sebagai *key* (K) dan *value* (V)**,** sementara query berasal dari *decoder*. Hal ini memungkinkan model menyesuaikan perhatian pada bagian relevan dari input saat menghasilkan *output*.

c. *Add and normalize + feed forward*

Seperti di *encoder*, hasil dari *encoder-decoder attention* dilewatkan ke *residual connection* dan *feed forward layer.*

d. Linear + *softmax*

Merupakan hasil akhir dari decoder berbentuk vektor yang memiliki dimensi sesuai dengan jumlah kata dalam *vocabulary*. Lapisan *softmax* digunakan untuk mengubahnya menjadi probabilitas, sehingga model dapat memilih kata yang paling mungkin sebagai *output.*

2.2.9 *Prompt*

Teknologi Generatif AI dapat melakukan tugasnya dengan baik apabila terdapat perintah yang jelas. Perintah dalam Generatif AI biasa disebut sebagai *prompt*, yang berfungsi sebagai query untuk menghasilkan keluaran yang sesuai dengan permintaan (Hibatulwafi, 2024). Dalam hal ini, *prompt* memainkan peran penting dalam memberikan arahan spesifik kepada model untuk menghasilkan hasil yang diinginkan. Oleh karena itu, *prompt* menjadi elemen kunci dalam proses *generate Output* yang berkualitas dan relevan.

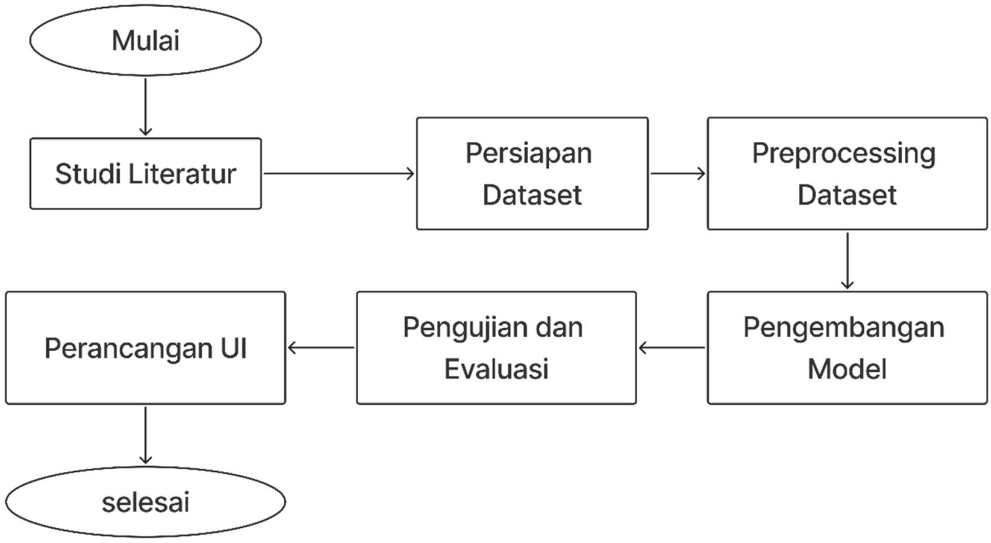
Dalam *Stable Diffusion*, terdapat dua jenis *prompt* utama, yaitu *prompt* positif dan *prompt* negatif. Prompt positif digunakan untuk menentukan elemen-elemen yang ingin ditampilkan dalam gambar yang dihasilkan, sedangkan *prompt* negatif berfungsi sebagai mekanisme panduan untuk menghindari elemen-elemen yang tidak diinginkan dalam produksi gambar.

Sebagai contoh, jika pengguna ingin membuat logo untuk restoran burger dengan gaya modern dan warna merah**,** maka *prompt* positif yang dapat digunakan adalah *"minimalist modern burger logo, bold typography, red and yellow color*

scheme, clean vector style, high quality, professional branding". Sementara itu, jika pengguna ingin menghindari elemen-elemen yang tidak diinginkan, seperti terlalu banyak detail, efek 3D yang berlebihan, atau warna yang tidak sesuai, maka prompt negatifnya bisa berupa *"low quality, blurry, too complex, photorealistic, 3D render, extra details, incorrect colors".* Dengan kombinasi prompt positif dan prompt negatif**,** logo yang dihasilkan dapat lebih sesuai dengan identitas merek dan kebutuhan desain yang diinginkan.

BAB III
METODE PENELITIAN

Penelitian ini bertujuan untuk mengembangkan sistem yang mampu menghasilkan logo berdasarkan deskripsi teks menggunakan *Stable diffusion* versi 1.5 dan *Fine-tuning Low Rank Adaption* (LoRA). Untuk mengembangkan model tersebut membutuhkan beberapa proses, gambar dibawah merupakan representasi alur metode penelitian dari pengembangan sistem.



Gambar 3. 1 Flowchart penelitian

3.1 Studi Literatur

Pada tahap ini, penelitian akan dimulai dengan kajian literatur untuk mendapatkan pemahaman yang mendalam mengenai konsep dan teknologi terkait. Kajian ini mencakup berbagai referensi dari penelitian terdahulu, jurnal ilmiah, buku, dan dokumentasi teknis yang relevan dengan pengembangan sistem *generate* logo berdasarkan teks deskripsi. Selain itu penulis juga melakukan kajian literatur terhadap metriks evaluasi untuk mengetahui kualitas dari gambar yang dihasilkan.

3.2 Persiapan dataset

Mengumpulkan dataset berupa gambar logo sebanyak 50 gambar. Dataset diperoleh dari berbagai platform, kemudian disimpan pada google drive. Pada pengumpulan dataset, dilakukan image *captioning* yang menghasilkan file txt. Image *captioning* adalah tahapan pemberian deskripsi pada gambar berdasarkan elemen yang terdapat didalamnya. Kemudian dataset tersebut dibagi menjadi data pelatihan dan data pengujian 40 data pelatihan 10 data pengujian. Dimana data pelatihan digunakan sistem untuk belajar sehingga dapat menghasilkan gaya yang sesuai dengan dataset, dan data pengujian digunakan sebagai pembanding.

3.3 *Preprocessing* dataset

Preprocessing dataset adalah mengolah data mentah menjadi data siap digunakan untuk *training*. Tujuan dilakukan *preprocessing* ini untuk meningkatkan variasi dataset dan mempermudah sistem untuk mengenali dataset karena data bersih dan konsisten. Dari data pelatihan yang sudah ada berupa gambar, selanjutnya di proses sebagai berikut :

a. *Resize*

   Resize adalah mengubah ukuran gambar yang ada secara keseluruhan, baik itu memperbesar atau memperkecil tanpa memotong. Tujuan dari adanya *Resize* adalah untuk menyamakan resolusi dataset.

b. *Cropping*

   Cropping adalah proses memotong bagian tertentu pada gambar. Tujuan dari proses ini adalah untuk mengkonsentrasikan perhatian pada bagian tertentu dari gambar dan menghilangkan bagian yang tidak penting.

c. *Color jitter*

   Color jitter adalah proses mengubah kecerahan, kontras, dan saturasi gambar secara acak. Bertujuan untuk membantu model menangani gambar dari berbagai kondisi pencahayaan.

d. *Rotation* dan *flipping* gambar

   Rotation dan *flipping* adalah melakukan pemutaran dan pembalikkan pada gambar dataset yang ada. Jenis pembalikkan yang terjadi pada proses ini yaitu

horizontal *flip*, vertical *flip* , dan diagonal *flip*. Bertujuan agar dataset lebih bervariasi.

3.4 Pengembangan model

3.4.1 *Stable diffusion* 1.5

Pada proses pengembangan model, *Stable diffusion* versi 1.5 berperan sebagai model dasar untuk menghasilkan gambar logo. Dalam penelitian ini, *Stable diffusion* versi 1.5 tidak secara langsung digunakan dengan bentuk aslinya. Namun, nantinya model *Stable diffusion* akan ditingkatkan menggunakan *fine-tuning* dengan teknik LoRA. Dengan begitu akan menghasilkan model baru yang dapat menghasilkan logo dengan gaya yang sesuai dengan dataset yang sudah dikumpulkan.

3.4.2 *Fine-tuning Low Rank Adaptation*

Pada tahapan ini dilakukan proses *fine-tuning* atau *training* terhadap model *Stable diffusion* versi 1.5 agar model dapat menghasilkan gambar dnegan gaya dan karakteristik tertentu berdasarkan dataset yang ada. Teknik *fine-tuning* yang digunakan adalah LoRA (*Low Rank Adaptation*). Dengan *fine-tuning* LoRA, model dapat menyesuaikan tanpa harus melatih ulang seluruh parameter model. Namun hanya melakukan pelatihan pada beberapa layer tertentu dari parameter yang relevan.

Dengan menerapkan *fine-tuning* menggunakan LoRA, diharapkan model dapat membuat logo dengan gaya yang lebih spesifik dan sesuai dengan kebutuhan pengguna sambil tetap menjaga efisiensi proses pelatihan dan inferensi.

3.5 Pengujian dan Evaluasi

3.5.1 Pengujian

Setelah melalui tahap pengembangan, selanjutnya adalah tahapan pengujian. Dimana pada tahap ini sistem akan menghasilkan gambar dari model yang sudah dilatih. Untuk menghasilkan gambar, *prompt* yang digunakan berupa teks yang sama dengan caption dari data pengujian. Alasan penggunaan teks yang

sama adalah untuk memudahkan proses perbandingan subjektif antara gambar hasil model baru dan gambar data pengujian.

3.5.2 Evaluasi

Pada evaluasi model, menggunakan matriks SSIM (*Structural Similarity Index Measure*) dan CLIP *score* dengan penjelasan sebagai berikut :

a. SSIM (*Structural Similarity Index Measure*)

SSIM merupakan metrik evaluasi yang digunakan untuk mengukur kesamaan antara dua gambar dengan mempertimbangkan bagaimana manusia melihat kualitas gambar (Sumarna dkk., 2021).. Berbeda dengan metrik seperti *Mean Squared Error* (MSE) yang hanya menghitung perbedaan piksel secara numerik, SSIM mempertimbangkan tiga aspek utama dalam sebuah gambar:

- *Luminance*, mengukur perbedaan tingkat kecerahan antara dua gambar
- *Contrast*, menilai perbedaan kontras antara gambar
- *Structure*, mengukur kesamaan pola dan struktur

Evaluasi menggunakan SSIM memiliki rentang *score* dari -1 sampai dengan 1. *Score* yang mendekati angka 1 menunjukkan bahwa gambar memiliki kesamaan. Perhitungan SSIM dilakukan dengan rumus berikut :

$$SSIM\ (x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)\ (\sigma_x^2 + \sigma_y^2 + C_2)} \quad (6)$$

Keterangan :

$x, y =$ gambar yang dibandingkan

$2\mu_{x,}\mu_y$ = rata-rata intensitas pixel

σ_x^2, σ_y^2= variansi

σ_{xy} = kovariansi antar kedua gambar

$C_1, \mathrm{C2} =$ konstanta stabilisasi

a. CLIP-MMD (*CLIP Maximum Mean Discrepancy*)

CLIPmetode evaluasi yang digunakan untuk mengukur kesesuaian antara gambar yang dihasilkan oleh model AI dengan deskripsi teks yang diberikan oleh pengguna. CMMD mengukur jarak antara distribusi *embedding* gambar yang dihasilkan oleh model dengan distribusi *embedding* gambar dari dataset

referensi (Jayasumana et al., 2023). Metrik ini menggabungkan dua konsep utama :

1) CLIP (*Contrastive Language-Image Pretraining*), merupakan model AI yang telah dilatih untuk memahami hubungan antara teks dan gambar dengan cara mengonversi keduanya ke dalam bentuk vektor di ruang embedding yang sama.
2) MMD (*Maximum Mean Discrepancy*), merupakan teknik statistik yang digunakan untuk mengukur perbedaan antara dua distribusi probabilitas, dalam hal ini distribusi embedding gambar dan embedding teks.

Dalam CLIP-MMD, variable utama yang diambil yaitu :

- *Embedding* teks, merupakan representasi vektor dari teks deskripsi yang diberikan pengguna. Dihasilkan oleh model CLIP, yang mengubah teks menjadi vektor dalam ruang embedding yang sama dengan gambar. Digunakan sebagai referensi untuk mengevaluasi seberapa sesuai gambar hasil AI dengan teks deskripsi awal
- *Embedding* gambar hasil *generate,* merupakan representasi vektor dari gambar yang dihasilkan oleh model AI. Dikonversi ke dalam ruang embedding oleh CLIP, sehingga dapat dibandingkan dengan embedding teks. Semakin dekat distribusinya dengan embedding teks, semakin sesuai gambar dengan deskripsi yang diberikan pengguna.
- *Embedding* gambar referensi, merupakan representasi vektor dari gambar referensi. Digunakan untuk membandingkan hasil generasi AI dengan gambar nyata dalam dataset uji. Jika gambar referensi tersedia, evaluasi akan mempertimbangkan seberapa dekat distribusi gambar hasil AI dengan gambar nyata yang relevan
- Kernel *function*

  Merupakan *function* yang mengukur kesamaan antara dua embedding, baik antara gambar dan teks maupun antara gambar hasil generasi dan gambar referensi. Digunakan dalam perhitungan *Maximum Mean Discrepancy* (MMD) untuk menilai perbedaan distribusi *embedding*.

Perhitungan CLIP-MMD dilakukan dengan rumus berikut :

$$MMD^2(E_g, E_r) = \mathbb{E}_{E_g,\acute{E}_g}[k(E_g\acute{E}_g)] - 2\mathbb{E}_{E_g,E_r}[k(E_g,E_r)] + \mathbb{E}_{E_r,E_r}[k(E_r,\acute{E}_r)] \quad (7)$$

Keterangan :

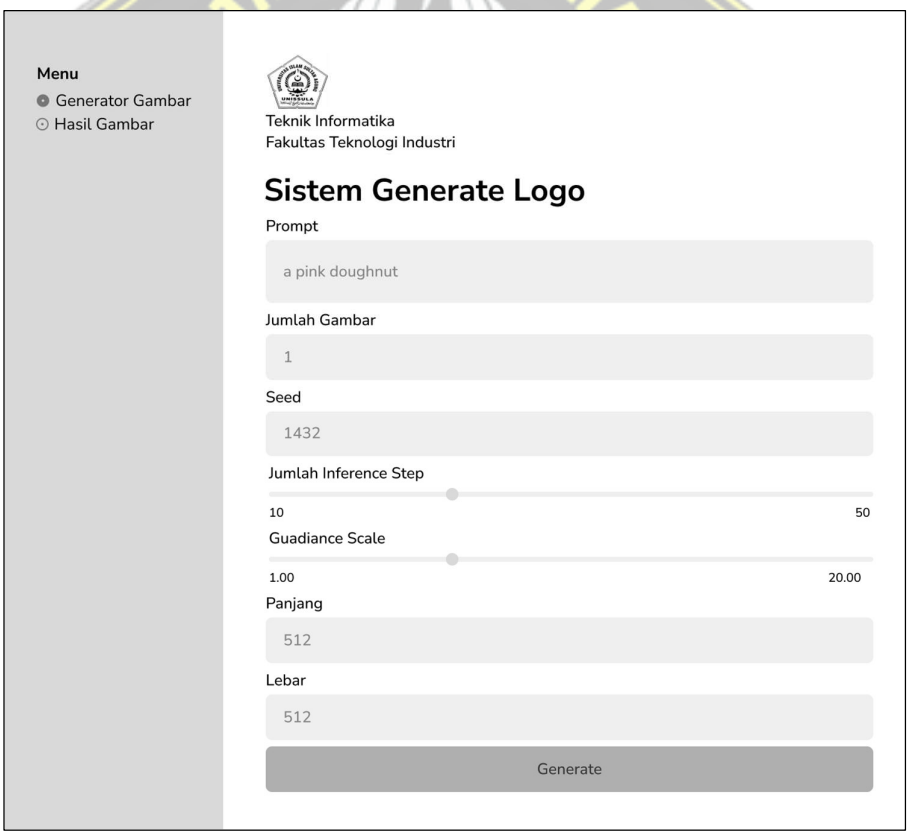
E_g = distribusi *embedding* gambar hasil AI

E_r = distribusi *embedding* gambar referensi

$k(x, y)$ = fungsi kernel yang mengukur kesamaan antara dua *embedding*

3.6 Perancangan *User Interface*

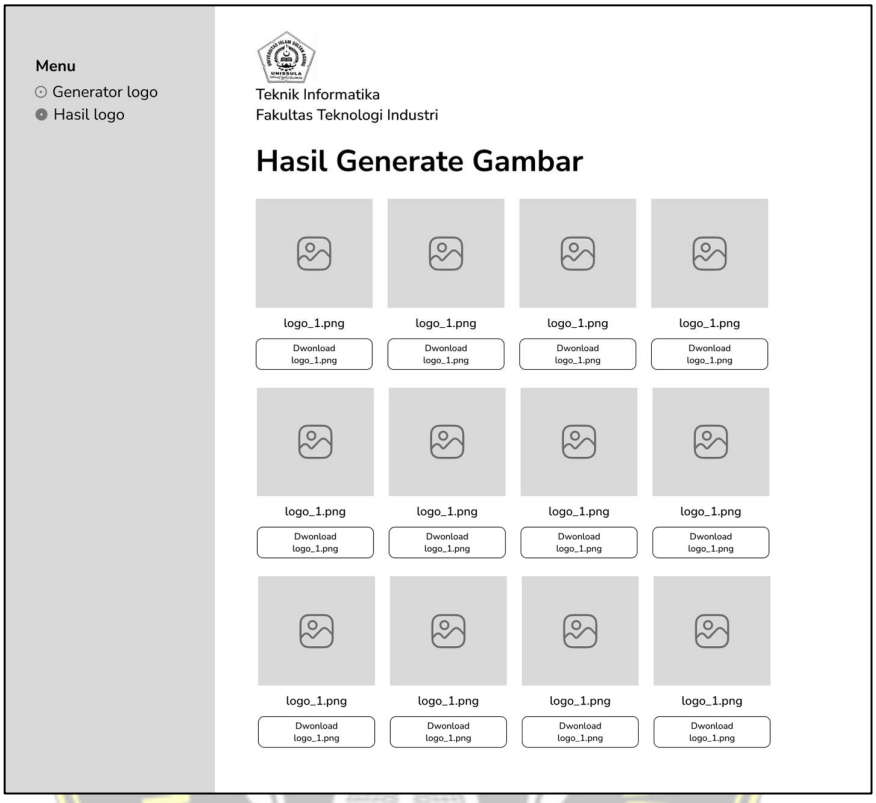
Pada tahap ini penulis merancang *user interface* untuk penggunaan sistem. Tujuan adanya *user interface* dari sistem adalah untuk mempermudah pengguna jika ingin menghasilakn logo dari sistem ini, yaitu sistem dengan model *Stable diffusion* versi 1.5 yang telah di*fine-tuning* menggunakan teknik LoRA.



Gambar 3. 2 *Wireframe* streamlit generator logo

Gambar diatas merupakan *wireframe* dari sistem *generate* logo pada bagian navigasi bar generator gambar yang akan dibuat nantinya. Pada saat menggunakan sistem, pengguna harus memasukkan beberapa parameter seperti :

- *Prompt* : berisikan deskripsi teks yang menggambarkan logo yang ingin dihasilkan.
- Jumlah gambar: digunakan untuk menentukan banyaknya gambar yang ingin dihasilkan dalam sekali proses.
- *Seed*: angka acak untuk memastikan replikasi gambar yang sama.
- Jumlah *inference step*: digunakan untuk menentukan jumlah iterasi dalam proses menghasilkan gambar. Semakin tinggi *inference step* maka gambar akan semakin detail namun prosesnya akan membutuhkan waktu yang lebih lama.
- *Guidance scale*: digunakan untuk mengontrol seberapa kuat gambar harus mengikuti *prompt*.
- Panjang & lebar: digunakan untuk menentukan ukuran gambar yang ingin dihasilkan.



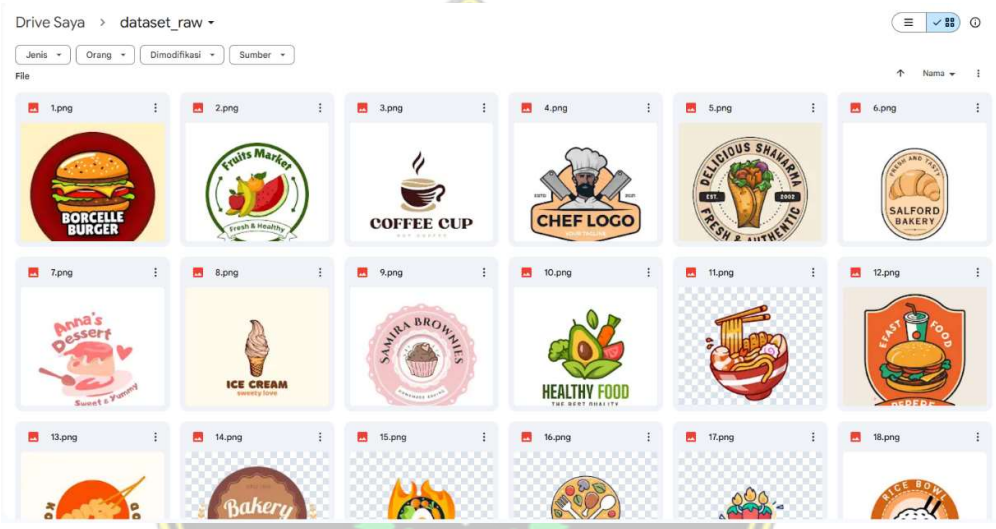
Gambar 3. 3 *Wireframe* streamlit hasil logo

Gambar diatas merupakan *wireframe* dari tampilan navigasi bar hasil gambar yang berhasil dibuat. Pada bagian tersebut, pengguna dapat melihat hasil logo yang dibuatnya dan dapat menyimpan hasilnya dengan menekan klik pada *button dwonload*.

BAB IV
HASIL DAN ANALISIS PENELITIAN

4.1 Persiapan dataset

Dataset yang berhasil dikumpulkan oleh penulis berupa gambar logo yang diperoleh dari platform canva dan freepik berjumlah 50 gambar. Kemudian dilakukan beberapa proses sebelum digunakan ke tahap selanjutnya. Proses tersebut berupa *captioning* dan pembagian dataset.



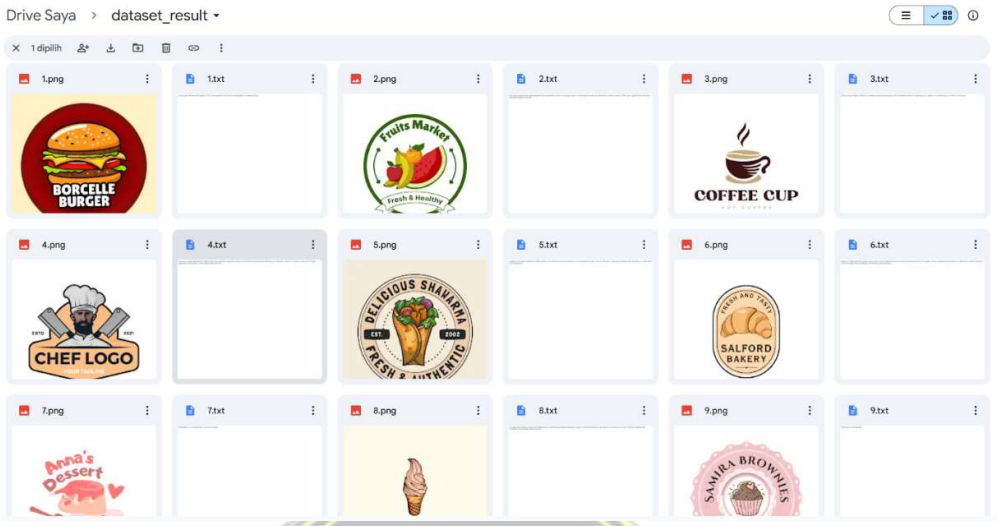
Gambar 4. 1 Dataset

Gambar diatas merupakan hasil dari pengumpulan dataset yang belum dibagi menjadi data pelatihan dan data pengujian. Dataset tersebut tersimpan pada folde dengan nama dataset_raw dan berisikan 50 gambar yang terdiri dari gambar logo makanan yang bersumber dari beberapa platform.

Tabel 4. 1 hasil *captioning*

No.	url gambar	*caption*
1.	/content/drive/MyDrive/dataset_result/1.png	*The logo features a cartoon-style burger with a sesame seed bun, lettuce, tomato, cheese, and a beef patty. The burger is set against a red circular background with a darker red border. Below the burger, "BORCELLE BURGER" is displayed in bold, white, uppercase letters with a black outline. The design is vibrant and modern, emphasizing a playful and appetizing theme.*
2.	/content/drive/MyDrive/dataset_result/10.png	*healthy food logo design*
3.	/content/drive/MyDrive/dataset_result/11.png	*a bowl of noodles with chops and chops*
4.	/content/drive/MyDrive/dataset_result/12.png	*fast food logo design*
5.	/content/drive/MyDrive/dataset_result/13.png	*a logo for a restaurant with a fork and knife*
6.	/content/drive/MyDrive/dataset_result/14.png	*bakery logo design*
…	….	...
50.	/content/drive/MyDrive/dataset_result/9.png	*santa brown's home baking*

Gambar diatas merupakan hasil pemberian teks deskripsi pada gambar sesuai dengan elemen yang ada didalam gambar tersebut. Pada proses ini, penulis menggunakan model *Bootstrapping Language Image Pretraining* (BLIP) untuk membantu proses *captioning*. BLIP berperan untuk memberikan teks deskripsi dan memasangkan antara gambar dan file deskripsinya.



Gambar 4. 2 Hasil *captioning*

Gambar diatas merupakan tampilan dari hasil proses *captioning* dari dataset awal. Pada folder dataset_result berisi 50 gambar dan 50 teks deskripsi dalam bentuk txt. Selain itu, pada folder tersebut dataset yang ada sudah dipasangkan dengan teks deskripsi.

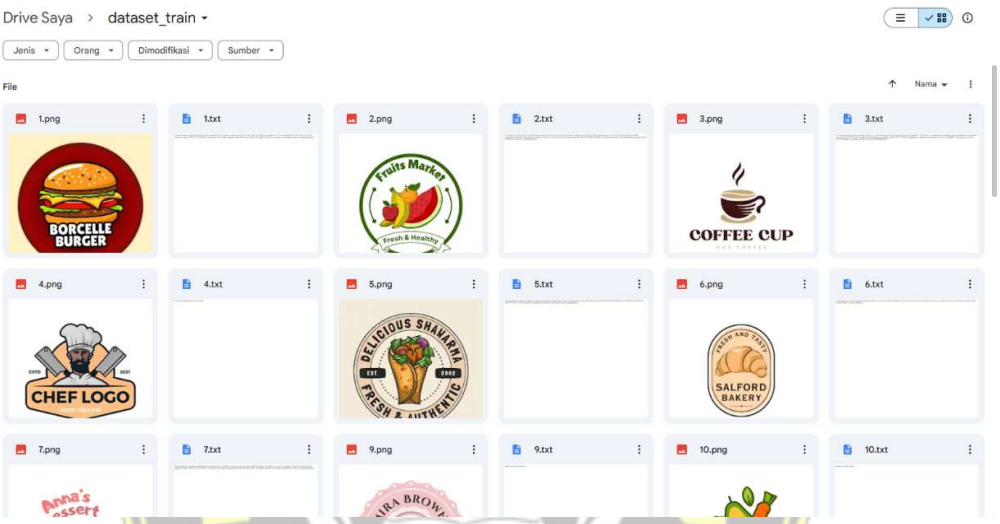
Tabel 4. 2 Total dataset

No	Dataset	Jumlah	Jumlah caption
1.	Data Pelatihan	40	40
2.	Data pengujian	10	10
Total		**50**	**50**

Tabel diatas menunjukkan jumlah pembagian dataset. Terdapat data pelatihan, yang digunakan untuk melatih model. Kemudian data pengujian yang digunakan untuk pengujian hasil gambar model setelah *fine-tuning*.

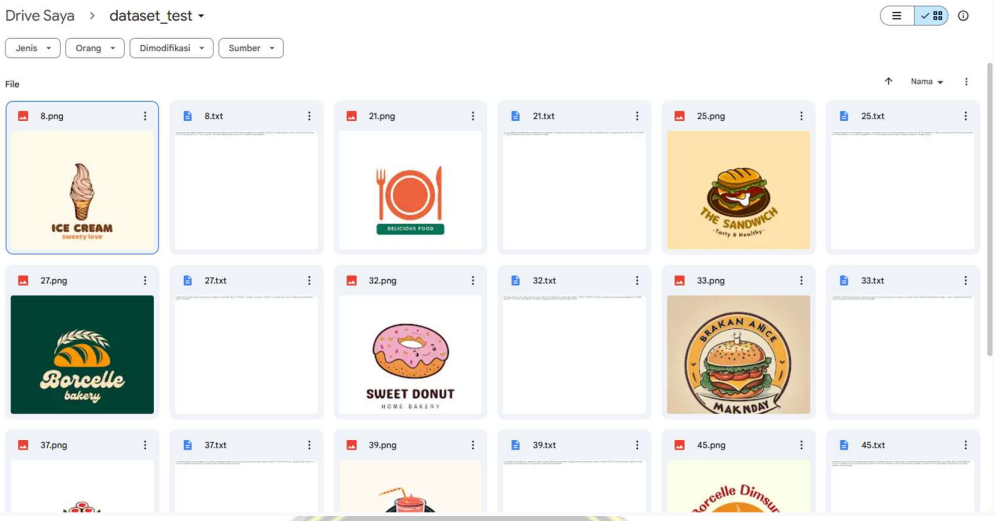
```
# membagi dataset menjadi pelatihan dan pengujian (80%
pelatihan, 20% pengujian)
train_df, test_df = train_test_split(df, test_size=0.2,
random_state=42) # atur agar dataset yang  terbagi konsisten

# menyimpan DataFrame ke CSV
train_df.to_csv('/content/drive/MyDrive/dataset_train/dataset_tr
ain.csv', index=False)
test_df.to_csv('/content/drive/MyDrive/dataset_test/dataset_test
.csv', index=False)

# memindahkan gambar dan caption ke folder pelatihan dan
pengujian
for _, row in train_df.iterrows():
    # menyalin gambar dan caption ke folder pelatihan
    shutil.copy(row['image'], os.path.join(TRAIN_DIR,
os.path.basename(row['image'])))
    shutil.copy(row['image'].replace('.png', '.txt'),
os.path.join(TRAIN_DIR,
os.path.basename(row['image']).replace('.png', '.txt')))

for _, row in test_df.iterrows():
    # menyalin gambar dan caption ke folder pengujian
    shutil.copy(row['image'], os.path.join(TEST_DIR,
os.path.basename(row['image'])))
    shutil.copy(row['image'].replace('.png', '.txt'),
os.path.join(TEST_DIR,
os.path.basename(row['image']).replace('.png', '.txt')))

# menampilkan beberapa baris pertama DataFrame pelatihan dan
pengujian
print("Training Data Sample:")
print(train_df.head())

print("Testing Data Sample:")
print(test_df.head())
```

Code diatas merupakan *code* untuk proses pembagian dataset. Dataset yang sudah diberikan deskripsi atau caption kemudian dibagi menjadi data pengujian dan data pelatihan dengan presentasi pembagian 80% data pelatihan dan 20% data pengujian. Data tersebut juga disimpan dalam bentuk csv.



Gambar 4. 3 Hasil pembagian data pelatihan

Gambar diatas merupakan hasil dari pembagian dataset, yaitu data pelatihan. Dataset tersebut tersimpan pada folder dengan nama dataset_*train*. Didalamnya terdapat 40 data pelatihan yang nantinya akan digunakan pada proses *training* model.



Gambar 4. 4 Hasil pembagian data pengujian

Gambar diatas merupakan hasil dari pembagian dataset, yaitu data pengujian. Dataset tersebut tersimpan pada folder dengan nama dataset_test. Didalamnya terdapat 10 data pengujian yang nantinya akan digunakan pada proses pengujian terhadap hasil gambar model setelah *fine-tuning*.

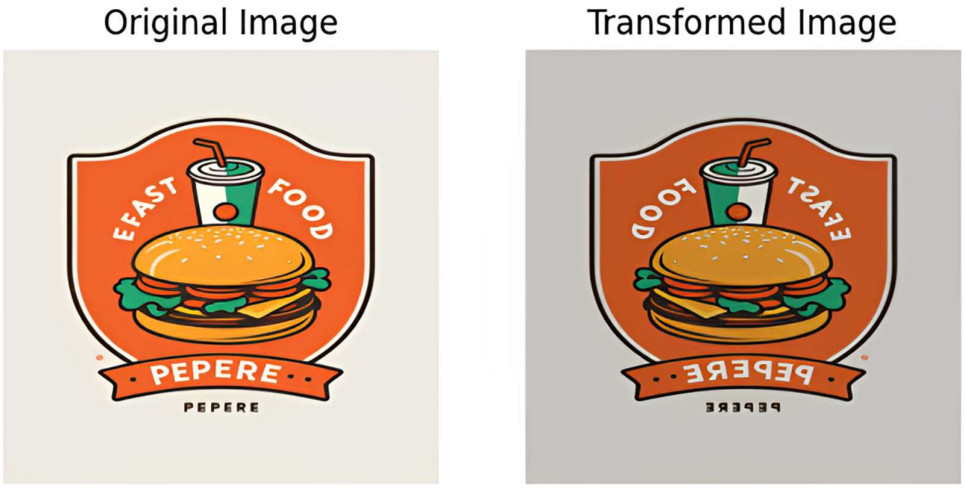
4.2 *Preprocessing* Dataset

Dataset yang berhasil diberikan caption selanjutnya dilakukan *preprocessing* untuk meningkatkan variasi dataset dan mempermudah sistem untuk mengenali dataset karena data bersih dan konsisten.

```
import matplotlib.pyplot as plt
from torchvision import transforms
from PIL import Image
import numpy as np

# Membuka folder gambar
img = Image.open("/content/drive/MyDrive/dataset_train/12.png")

# Mendefinisikan transformasi untuk preprocessing
visualize_transforms = transforms.Compose(
    [
        transforms.Resize((512, 512),
interpolation=transforms.InterpolationMode.BILINEAR),
```

```
        transforms.RandomHorizontalFlip(),
        transforms.ColorJitter(brightness=0.2, contrast=0.2,
saturation=0.2),
        transforms.ToTensor(),
    ]
)

# Menerapkan transformasi ke gambar
img_transformed = visualize_transforms(img)

# Mengubah tensor kembali menjadi gambar agar bisa ditampilkan
img_transformed = img_transformed.permute(1, 2, 0).numpy()  #
Menata ulang dimensi agar sesuai untuk ditampilkan
img_transformed = (img_transformed * 255).astype(np.uint8)  #
Mengembalikan nilai piksel ke rentang 0-255

# Menampilkan gambar asli
plt.subplot(1, 2, 1)
plt.imshow(img)
plt.title("Original Image")
plt.axis('off')

# Menampilkan gambar yang telah diproses
plt.subplot(1, 2, 2)
plt.imshow(img_transformed)
plt.title("Transformed Image")
plt.axis('off')
plt.show()
```

Code diatas merupakan *code* untuk tahapan *preprocessing* dataset. *Preprocessing* dataset tersebut meliputi pengubahan ukuran, pembalikkan gambar, dan pengubahan warna.



Gambar 4. 5 Hasil *preprocessing* dataset

Gambar diatas menunjukkan hasil *preprocessing* pada dataset, gambar telah melalui beberapa tahapan proses sebelum proses pelatihan model. Dengan adanya *preprocessing* diharapkan dapat model mampu mengenali pola dengan lebih baik.

4.3 Pengembangan Model

Pada tahap ini penulis telah melakukan *fine-tuning* pada model *Stable diffusion*. Tujuan dilakukan proses ini adalah untuk melatih agar model dapat menghasilkan gambar yang sesuai dengan dataset. Adapaun penulis melakukan uji coba terhadap 500 iterasi *training* pada model *Stable diffusion*.

```
!accelerate launch
/content/drive/MyDrive/script_training/train_text_to_image_lora_
csv.py \
  --pretrained_model_name_or_path="stable-diffusion-v1-5/stable-
diffusion-v1-5" \
  --
train_data_dir="/content/drive/MyDrive/dataset_train/dataset_tra
in.csv" \
  --image_column="image" \
  --caption_column="captions" \
  --dataloader_num_workers=8 \
  --resolution=512 \
  --random_flip \
  --train_batch_size=1 \
  --gradient_accumulation_steps=4 \
  --max_train_steps=500 \
  --learning_rate=2e-5 \
  --max_grad_norm=1 \
  --lr_scheduler="cosine_with_restarts" \
  --lr_warmup_steps=100 \
  --output_dir="/content/drive/MyDrive/lora_outputs" \
  --checkpointing_steps=100 \
  --validation_prompt="cartoon style burger logo featuring a
juicy cheeseburger with lettuce, tomato, and a sesame seed bun.
Encircled text reads 'Delicious Burgers' in bold retro font. A
ribbon banner displays 'EST. 2024' on a beige textured
background." \
  --seed=777 \
  --rank=16 \
  --report_to tensorboard \
  --mixed_precision fp16 \
  --snr_gamma=5.0
```

Gambar diatas menunjukkan parameter yang digunakan saat pelatihan model. Parameter tersebut berfungsi untuk mengontrol proses pelatihan model dan berpengaruh pada kesesuaian bobot terhadap data. Berikut adalah beberapa paramaeter yang digunakan :

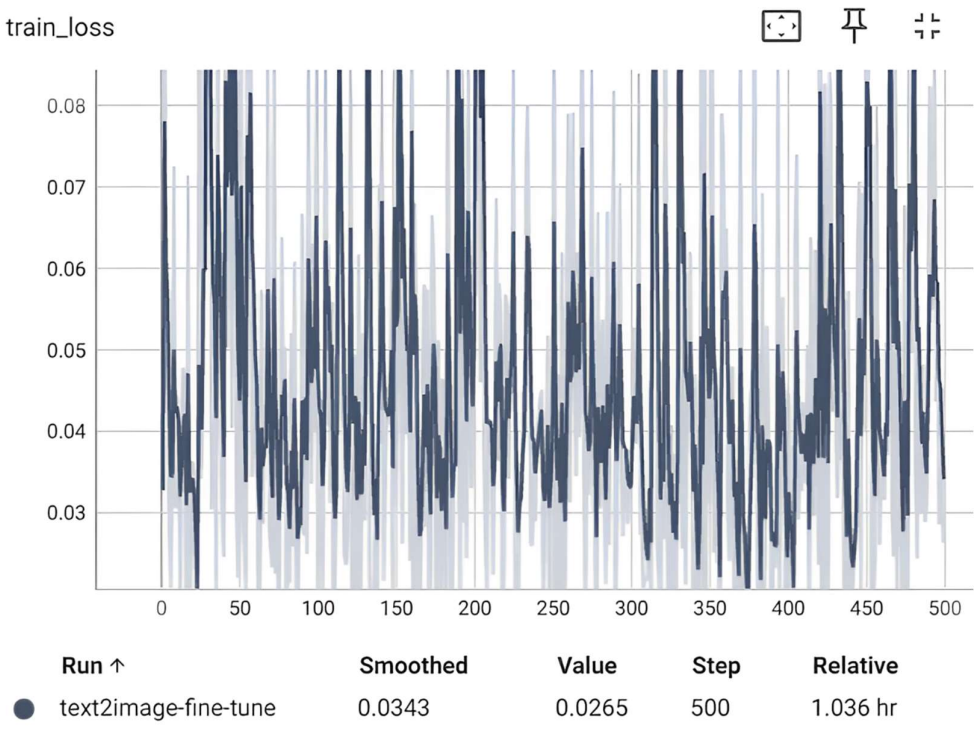
Tabel 4. 3 Parameter pelatihan

Parameter	Keterangan
Model *Stable diffusion* versi 1.5	Model dasar yang digunakan sebelum dilatih dengan dataset baru. Model ini sudah dilatih sebelumnya dan akan disesuaikan dengan data baru menggunakan lora *fine-tuning*.
Dataset	Lokasi dataset yang berisi gambar dan teks deskripsi dalam format CSV. Yang didalamnya terdapat kolom image berisikan gambar dan kolom captions berisikan teks deskripsi dari gambar.
Worker	Jumlah worker yang digunakan untuk memuat data selama *training*. Semakin banyak worker, semakin cepat data diproses, terutama jika dataset besar.
Resolusi gambar	Ukuran gambar yang digunakan dalam *training*, diubah menjadi 512x512 piksel. Dengan adanya resolusi yang seragam dapat mempermudah model dalam memahami pola dalam gambar.
Random flip	Membalik gambar secara horizontal secara acak sebagai bagian dari augmentasi data. Membantu model agar tidak hanya mengenali satu pola orientasi tertentu dalam gambar.
Train batch	Jumlah gambar yang diproses sekaligus dalam satu langkah pelatihan. Semakin kecil, semakin sedikit memori yang digunakan.
Gradient accumulation	Mengakumulasi gradien langkah sebelum memperbarui bobot model. Digunakan untuk mensimulasikan batch size yang lebih besar tanpa menghabiskan terlalu banyak memori.
Max train	Jumlah total langkah *training* sebelum proses berhenti. Digunakan untuk menentukan durasi *training* agar tidak berjalan terlalu lama atau terlalu singkat.

Learning rate	Menentukan seberapa besar perubahan bobot model dalam setiap langkah *training*. Learning rate yang terlalu tinggi bisa membuat model tidak stabil, sedangkan terlalu rendah bisa membuat *training* berjalan lambat.
Max grad norm	Membatasi perubahan bobot agar model tetap stabil dan tidak mengalami lonjakan besar saat proses pelatihan
Lr *scheduler*	Mengatur kecepatan belajar agar menurun secara bertahap dengan pola tertentu, supaya model bisa lebih stabil dalam belajar.
Lr *warmup*	Jumlah langkah awal di mana learning rate secara perlahan meningkat hingga mencapai nilai target. Hal ini dilakukan agar model tidak langsung melakukan perubahan besar di awal *training* yang bisa menyebabkan ketidakstabilan.
Checkpoint	Menentukan setiap berapa langkah *training* model akan menyimpan checkpoint. Checkpoint berguna agar *training* bisa dilanjutkan jika terjadi gangguan atau jika ingin melakukan evaluasi di titik tertentu.
Validation prompt	Deskripsi teks yang digunakan untuk menguji apakah model telah belajar dengan baik. Model akan mencoba menghasilkan gambar berdasarkan teks ini, sehingga bisa dievaluasi apakah hasilnya sesuai dengan yang diharapkan.
Seed	*Seed* untuk memastikan *training* menghasilkan hasil yang sama setiap kali dijalankan. Berguna untuk replikasi dan debugging.
Rank lora	Menentukan seberapa banyak bagian model yang akan diubah selama pelatihan. Angka yang lebih tinggi membuat model bisa belajar lebih banyak variasi dari data baru.
Tensorboard	Mengirimkan data *training* ke tensorboard untuk memantau performa *training* melalui grafik dan metrik. Berguna untuk analisis dan evaluasi.

Precision	Menggunakan format angka 16-bit agar pelatihan lebih cepat dan tidak memakan banyak memori, terutama saat menggunakan GPU.
Snr gamma	Membantu model lebih fokus pada bagian penting dalam gambar dan mengabaikan bagian yang tidak relevan. Ini bisa membuat pelatihan lebih stabil.

Ketika melakukan proses pelatiham, penulis memantau loss yang ada selama proses pelatihan berlangsung melalui grafik yang berhasil dikirim kan melalui tensorboard.

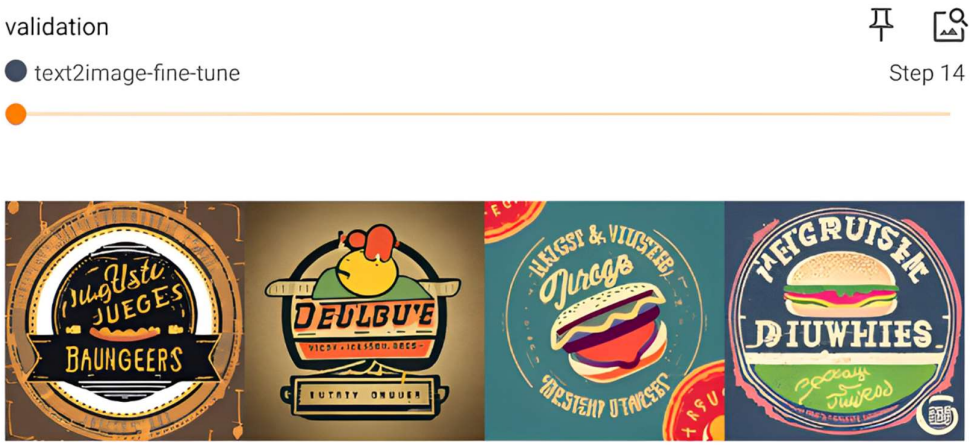


Gambar 4. 6 Grafik loss

Gambar diatas menunjukkan grafik loss pada saat proses *training* berlangsung. Berdasarkan grafik tersebut, terlihat bahwa nilai loss mengalami penurunan secara bertahap beriringan dengan bertambahnya langkah iterasi. Hal tersebut menunjukkan bahwa proses *training* model berhasil mempelajari data yang ada. Namun juga terdapat kondisi dimana loss mengalami fluktuasi, dimana pada

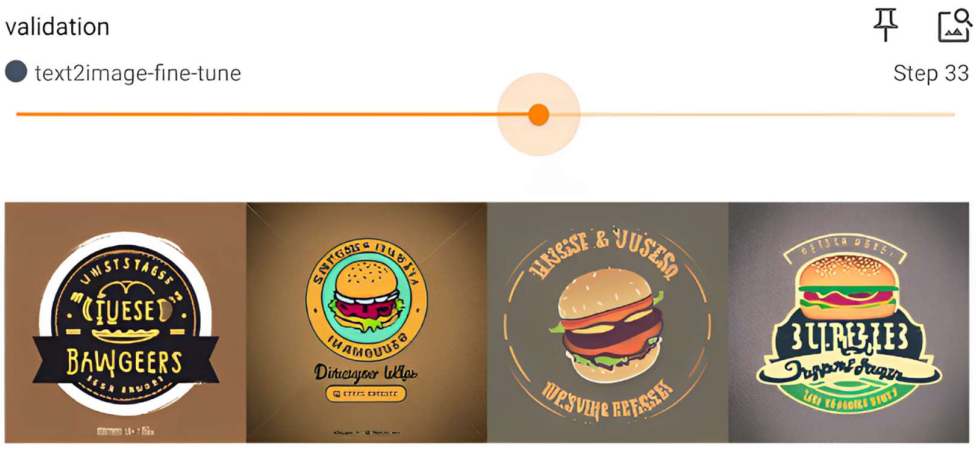
proses *training* nilai loss naik dan turun secara drastis. Ini menunjukkan adanya ketidakstabilan dalam proses model mempelajari data. Secara keseluruhan, grafik tersebut menunjukkan bahwa model belajar dengan baik, tetapi masih mengalami fluktuasi loss yang dapat mengakibatkan terhambatnya konvergensi model.

Selain grafik loss, terdapat juga validasi *prompt* yang dihasilkan selama proses pelatihan berlangsung pada setiap iterasinya. Validasi *prompt* digunakan untuk



Gambar 4. 7 Validasi *step* 14

Hasil validasi menunjukkan bahwa model telah berhasil menghasilkan gambar namun masih belum merepresentasikan burger dengan baik. jika dilihat dengan subjektif gambar tersebut masih ada yang kurang sempurna galam membentuk gambar.



Gambar 4. 8 Validasi *step* 33

Gambar diatas merupakan hasil validasi gambar *step* 33 dari model yang telah dilatih menggunakan teknik LoRA *fine-tuning*. Gambar tersebut menggunakan *prompt* "*cartoon-style burger logo featuring a juicy cheeseburger with lettuce, tomato, and a sesame seed bun. Encircled text reads 'Delicious Burgers' in bold retro font*". Secara umum, model berhasil menghasilkan suatu logo dengan bentuk dan warna yang cukup konsisten. Namun teks yang ditampilkan pada logo tidak terbaca dengan jelas.

4.4 Pengujian dan Evaluasi

4.4.1 Pengujian

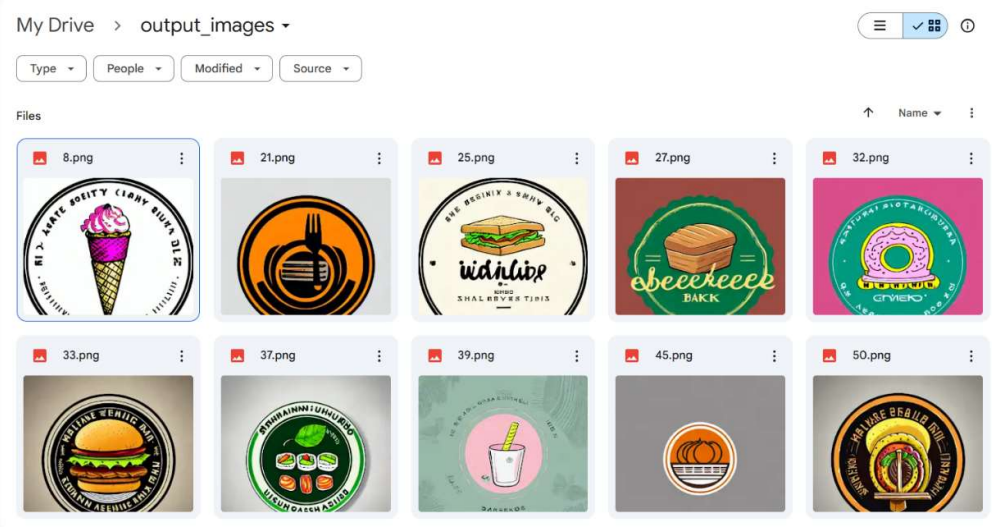
Pada tahap pengujian, model akan melakukan *generate* gambar berdasarkan dataset pengujian. Model akan menggunakan teks deskripsi yang sama dengan dataset pengujian dan menghasilkan gambar yang sesuai dengan deskripsi teks. Berikut adalah teks deskripsi atau *caption* pada data pengujian.

Tabel 4. 4 *Caption* data pengujian

Nama file	*Prompt*
8.png	*A playful logo with a white background. A simple, hand-drawn illustration of a pink soft-serve ice cream cone is centered. The text "Ice Cream" is written in a bold, sans-serif font below the cone,*

	and "sweety love" is in a cute, cursive font. The overall design conveys a sense of fun, sweetness, and happiness.
21.png	*The logo features a stylized plate icon, flanked by a fork and knife. It's rendered in a bold, geometric style with an orange hue. Below the icon, a dark green banner reads 'DELICIOUS FOOD' in white, sans-serif font. The design is minimalist and modern*
25.png	*A minimalist logo with a warm background. A simple, hand-drawn illustration of a classic sandwich is centered. The text "The Sandwich" is written in a bold, sans-serif font below the sandwich, and "Tasty & Healthy" is in a smaller, handwritten font. The overall design conveys a sense of simplicity, freshness, and deliciousness.*
27.png	*A golden-brown bread loaf with curved slashes, topped by a wheat stalk. Below, "Borcelle" in elegant cursive and "bakery" in a smaller script. Set on a dark green background for a warm, inviting feel.*
32.png	*This logo features a cute, cartoon-style pink donut with colorful sprinkles and a smiling face. Below, "SWEET DONUT" is written in bold brown uppercase letters, followed by "HOME BAKERY" in a clean, minimalist font. The design is playful and inviting, perfect for a bakery brand.*
33.png	*A minimalist, circular logo with a warm background. A simple, hand-drawn illustration of a burger is centered. The text "Brakan Anice Maknday" is written in a bold, sans-serif font around the burger, conveying a sense of deliciousness and nostalgia.*
37.png	*A minimalist logo with a white background. A simple, hand-drawn illustration of four sushi rolls on a plate with green leaves is centered. The text "Sushichan - Japanese Food" is written in a bold, sans-serif font below the sushi, conveying a sense of freshness and authenticity.*

39.png	*A minimalist, circular logo with a pastel color palette. A simple, hand-drawn illustration of a pink smoothie in a glass with a straw is centered. The text "Smoothie Station" is written in a bold, sans-serif font around the glass, conveying a sense of freshness and coolness.*
45.png	*A minimalist, circular logo with a vibrant orange background. A detailed illustration of a brown dim sum steamer filled with three steaming dumplings takes center stage. The text "Borcelle Dimsum" is elegantly curved around the circle in a modern cursive font, while "Since 2024" is placed below in a smaller, sans-serif font. The overall design conveys a sense of warmth, tradition, and deliciousness.*
50.png	*A minimalist, circular logo with a warm background. A simple, hand-drawn illustration of a kebab wrapped in a tortilla is centered. The text "Kebab Borcelle" is written in a bold, sans-serif font around the kebab, and "Enak, Lezat & Bergizi" is written below, conveying a sense of deliciousness and freshness.*



Gambar 4. 9 Hasil pengujian

Gambar diatas menunjukkan hasil pengujian menggunakan model baru. Secara visual, gambar yang dihasilkan sudah menyerupai logo. Ini menunjukkan model mampu menghasilkan desain yang bervasiasi. Namun, pada bagian teks model masih susah untuk mengenalinya sehingga membuat teks yang dimunculkan tidak terbaca dengan baik.

4.4.2 Evaluasi

Pada bagian ini, gambar yang dihasilkan oleh model pada tahap pengujian dilakukan evaluasi. Evaluasi berfungsi untuk menilai seberapa sesuai dan bagus kualitas dari gambar yang dihasilkan model.

1) SSIM (*Structural Similarity Index Measure*)

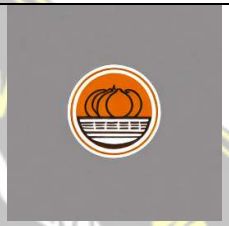
   Hasil *score* evaluasi menggunakan metrik SSIM mendapatkan rata-rata sebesar 0.5365, dimana *score* tersebut berada dalam kategori cukup baik. *Score* tersebut menunjukkan bahwa gambar hasil dari model baru terdapat kemiripan dengan data uji, namun terdapat sedikit perbedaan yang terlihat.

Gambar 5. 1 Evaluasi SSIM

Untuk melihat lebih detailnya perbandingan gambar hasil model baru dan data uji, disajikan pada tabel berikut.

Tabel 5. 1 Hasil Evaluasi SSIM

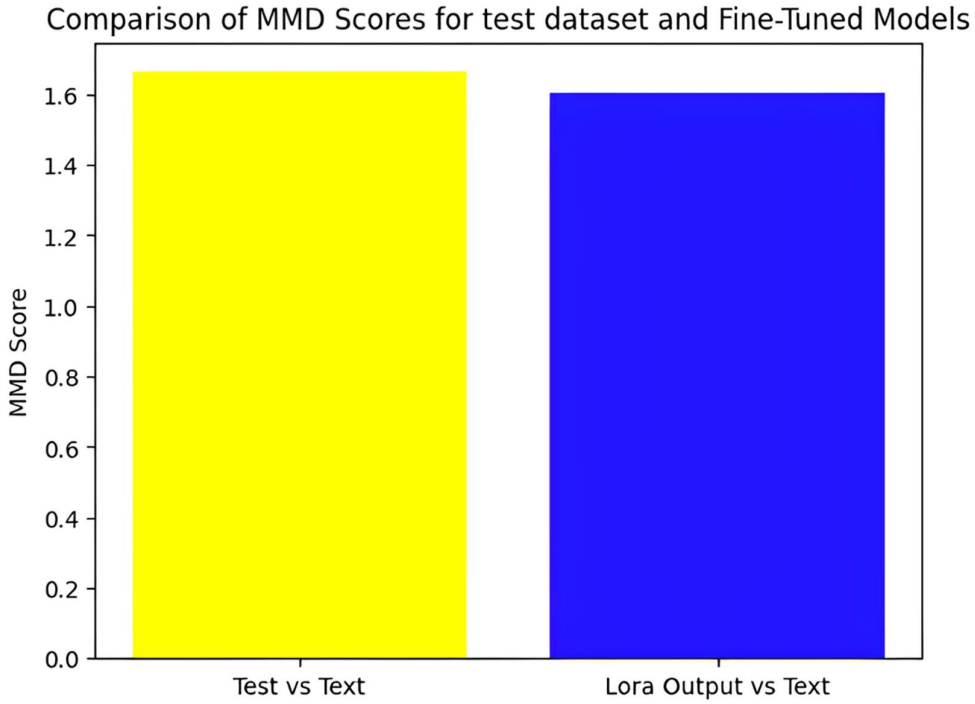
No	Nama file	Dataset	Hasil Pengujian	*Score* SSIM
1.	8.png			0.51
2.	21.png			0.56
3.	25.png			0.57
4.	27.png			0.51
5.	32.png			0.54
6.	33.png			0.65

7.	37.png			0.45
8.	39.png			0.53
9.	45.png			0.59
10.	50.png			0.41

2) CLIP-MMD (*CLIP Maximum Mean Discrepancy*)

Score yang diperoleh ketika dilakukan evaluasi menggunakan CLIP-MMD direpresentasikan pada diagram dibawah.

```
CMMD Score (Test Dataset vs Text): 1.6639246940612793
CMMD Score (Lora Output vs Text): 1.6033869981765747
Model fine-tuned menghasilkan gambar yang lebih sesuai dengan prompt!
```

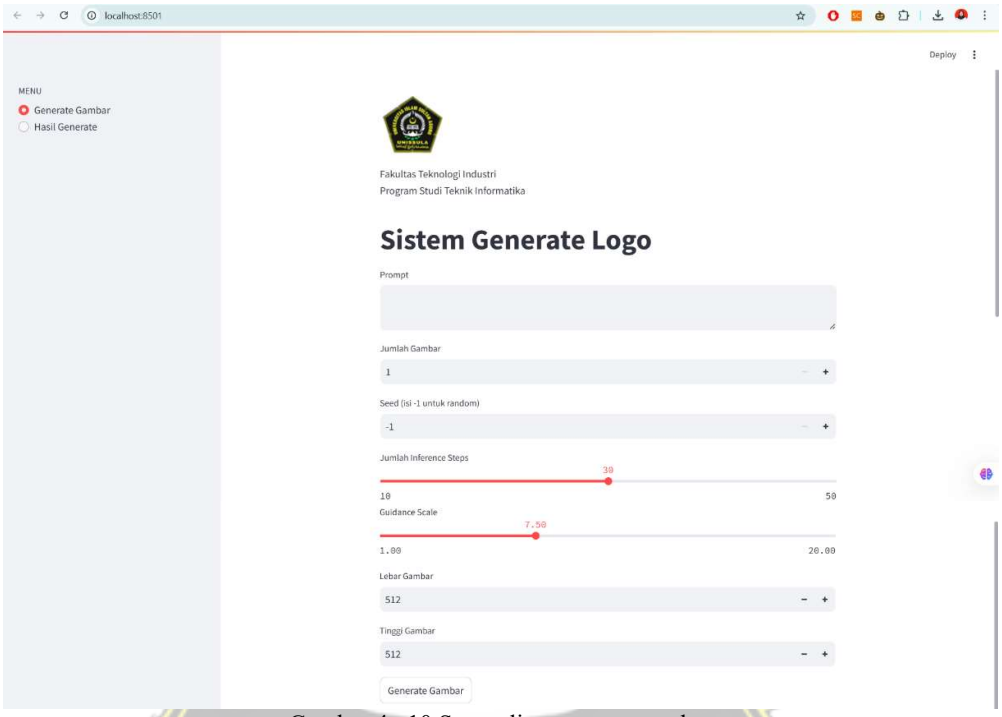


Gambar 5. 2 Diagram hasil CLIP-MMD

Gambar diatas menunjukkan *score* yang diperoleh CLIP-MMD dalam membandingkan kesesuaian gambar dengan teks deskripsi yang dihasilkan oleh model baru dan data uji. Dataset pengujian mendapatkan *score* 1.663 sedangkan model baru menghasilakan *score* 1,603 . Hal ini menunjukkan bahwa gambar yang dihasilkan oleh model baru menghasilkan gambar yang lebih sesuai dengan teks deskripsi, meskipun hanya memiliki perbedaan yang tipis.

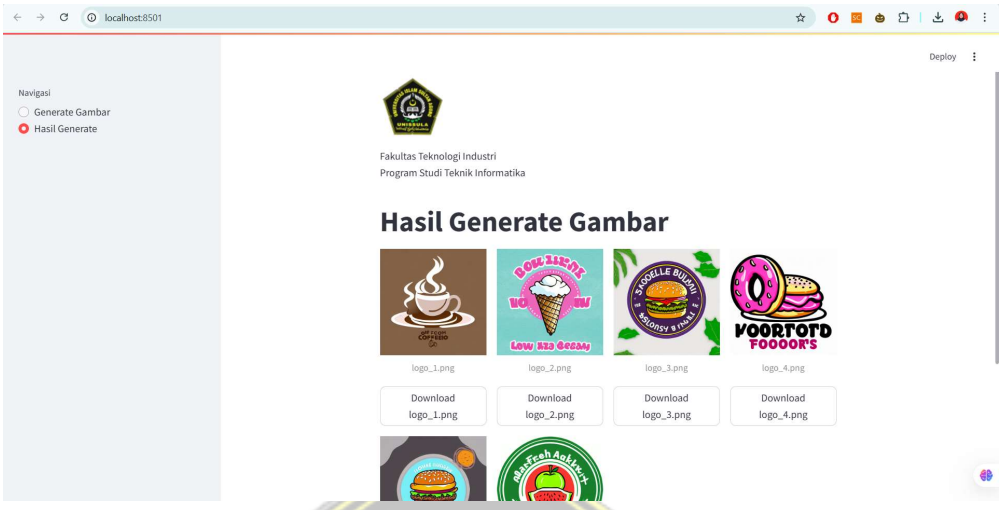
4.5 Perancangan Sistem

Berikut adalah tampilan website dari sistem *generate* logo yang siap digunakan oleh pengguna. Pengguna dapat melakukan genarate gambar pada website streamlitw yang telah dibuat.



Gambar 4. 10 Streamlit generator gambar

Pada gambar diatas pengguna dapat memasukkan beberapa parameter seperti *prompt*, jumlah gambar, *seed*, jumlah *inference step*, *Guidance scale*, lebar gambar, dan tinggi gambar.



Gambar 4. 11 Streamlit hasil *generate*

Gambar diatas merupakan tampilan website pada navbar hasil *generate* gambar menggunakan. Pada bagian ini pengguna dapat melihat hasil dan dwonload logo yang diinginkan.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian ini, dapat disimpulkan bahwa model *Stable diffusion* dapat dilakukan *fine-tuning* dengan teknik LoRA. Model yang di *fine-tuning* dapat menghasilkan gambar berupa logo berdasarkan deskripsi teks yang dapat digunakan pengguna untuk mengeksplorasi berbagai konsep desain logo serta mendukung keputusan akhir logo. Model dinilai dapat mempelajari pola dari dataset pelatihan yang ada dan menghasilkan gambar logo yang cukup baik. Hal ini ditunjukkan dengan hasil evaluasi menggunakan metrik SSIM memperoleh *score* rata-rata 0,536 dengan kategori cukup baik. *Score* tersebut menunjukkan bahwa gambar model baru menyerupai dengan data uji namun masih ada sedikit perbedaan yang terlihat. Kemudian hasil evaluasi menggunakan metrik CLIP-MMD memperoleh *score* 1,612. Dimana *score* tersebut menggambarkan bahwa hasil gambar oleh model baru sesuai dengan teks deskripsi yang ada. Selain itu model mengalami kesulitan untuk memunculkan teks pada gambar. Hal tersebut bisa disebabkan oleh beberapa aspek seperti jumlah dataset yang digunakan masih terbatas, dataset yang digunakan kurang beragam, dan jumlah iterasi ketika proses pelatihan masih kurang. Kekurangan dalam penelitian ini diharapkan dapat diperbaiki di penelitian selanjutnya.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, terdapat beberapa hal yang dapat dikembangkan untuk penelitian selanjutnya agar sistem generasi logo berbasis Stable Diffusion semakin optimal. Pertama, tingkat kesesuaian antara teks deskripsi dan hasil gambar perlu ditingkatkan dengan menerapkan teknik tambahan yang memastikan bahwa elemen visual yang dihasilkan sesuai dengan prompt yang diberikan. Selain itu, peningkatan kualitas visual dari logo yang dihasilkan dapat dilakukan dengan memperbesar jumlah dataset pelatihan serta mengeksplorasi

parameter pelatihan, seperti jumlah iterasi yang lebih banyak, agar model dapat mempelajari karakteristik desain logo dengan lebih baik.

Selanjutnya, penggunaan versi model Stable Diffusion yang lebih baru dapat menjadi solusi untuk meningkatkan resolusi dan pemahaman model terhadap teks input, sehingga desain logo yang dihasilkan menjadi lebih presisi dan sesuai dengan ekspektasi pengguna. Terakhir, pengembangan sistem dengan dukungan bahasa Indonesia juga perlu dipertimbangkan agar pengguna dapat memberikan deskripsi dalam bahasa Indonesia secara langsung tanpa harus menerjemahkannya ke dalam bahasa Inggris. Dengan adanya pengembangan ini, diharapkan sistem dapat lebih fleksibel, akurat, dan mudah digunakan oleh lebih banyak pengguna.

DAFTAR PUSTAKA

Aftitah, F. N., K, J. L., Hasanah, K., Lailatul, N., Bina, U., & Informatika, S. (2025). *Pengaruh UMKM Terhadap Pertumbuhan Ekonomi Di Indonesia Pada Tahun 2023 Pemerintah mendukung UMKM melalui program seperti Kredit Usaha Rakyat (KUR), meskipun penyalurannya tahun 2023 belum memenuhi target . UMKM kini terus*. *3*, 32–43.

Ainun, N., Maming, R., & Wahida, A. (2023). Pentingnya Peran Logo Dalam Membangun Branding Pada Umkm. *Jesya*, *6*(1), 674–681. https://doi.org/10.36778/jesya.v6i1.967

Arly, A., Dwi, N., & Andini, R. (2023). Implementasi Penggunaan Artificial Intelligence Dalam Proses Pembelajaran Mahasiswa Ilmu Komunikasi di Kelas A. *Prosiding Seminar Nasional*, 362–374.

Aulia, F., Afriwan, H., & Faisal, D. (2021). Konsistensi Logo Dalam Membangun Sistem Identitas. *Gorga : Jurnal Seni Rupa*, *10*(2), 439. https://doi.org/10.24114/gr.v10i2.28131

Berrahal, M., & Azizi, M. (2022). Optimal text-to-image synthesis model for generating portrait images using generative adversarial network techniques. *Indonesian Journal of Electrical Engineering and Computer Science*, *25*(2), 972–979. https://doi.org/10.11591/ijeecs.v25.i2.pp972-979

Cuevas, E., Avalos, O., & Gálvez, J. (2023). *Analysis and Comparison of Metaheuristics*.*1063*(November),21–24. https://link.springer.com/10.1007/978-3-031-20105-9

Faiz Muntazori, A., & Listya, A. (2021). Branding UMKM Produk Kopi Bang Sahal melalui Desain Logo. *SENADA : Semangat Nasional Dalam Mengabdi*, *1*(3), 342–351.

Gambashidze, A., Kulikov, A., Sosnin, Y., & Makarov, I. (2024). *Aligning Diffusion Models with Noise-Conditioned Perception*. 1–13. http://arxiv.org/abs/2406.17636

Halim, A. (2020). Pengaruh Pertumbuhan Usaha Mikro, Kecil dan Menengah Terhadap Pertumbuhan. *Ekonomi*, *2*(September), 158.

Hibatulwafi, F. (2024). *Fenomena Penggunaan Generative AI dalam Perilaku*

Pencarian Informasi Praktisi Teknologi. *31*(2), 141–155. https://doi.org/10.37014/medpus.v31i2.5222

Hu, E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). Lora: Low-Rank Adaptation of Large Language Models. *ICLR 2022 - 10th International Conference on Learning Representations*, 1–26.

Huang, P., Gao, X., Huang, L., Jiao, J., Li, X., Wang, Y., & Guo, Y. (2024). Chest-Diffusion: A Light-Weight Text-to-Image Model for Report-to-CXR Generation. *Proceedings - International Symposium on Biomedical Imaging*, 1–5. https://doi.org/10.1109/ISBI56570.2024.10635417

Jayasumana, S., Ramalingam, S., Veit, A., Glasner, D., Chakrabarti, A., & Kumar, S. (2023). *Rethinking FID: Towards a Better Evaluation Metric for Image Generation*. https://doi.org/10.1109/CVPR52733.2024.00889

Liu, B., Wang, C., Cao, T., Jia, K., & Huang, J. (2024). *Towards Understanding Cross and Self-Attention in Stable Diffusion for Text-Guided Image Editing*. https://doi.org/10.1109/CVPR52733.2024.00747

Liu, C., Takikawa, T., & Jacobson, A. (2024). *A LoRA is Worth a Thousand Pictures*. http://arxiv.org/abs/2412.12048

Mahendra, A. (2022). *Implementasi low-rank adaptation of large languange model (lora) untuk effisiensi large language model*. *8*(3), 110–118.

Mei, S. (2024). *U-Nets as Belief Propagation: Efficient Classification, Denoising, and Diffusion in Generative Hierarchical Models*. http://arxiv.org/abs/2404.18444

Naseem, B. (2024). *Temporal Scene Generation with Stable Diffusion*. huggingface.co. https://huggingface.co/blog/Bilal326/stable-diffusion-project

Peebles, W., & Xie, S. (2023). Scalable Diffusion Models with Transformers. *Proceedings of the IEEE International Conference on Computer Vision*, 4172–4182. https://doi.org/10.1109/ICCV51070.2023.00387

Pradana, A. G., Setiadi, D. R. I. M., & Muslikh, A. R. (2024). Fine tuning model Convolutional Neural Network EfficientNet-B4 dengan augmentasi data untuk klasifikasi penyakit kakao. *Journal of Information System and Application Development*, *2*(1), 01–11.

https://doi.org/10.26905/jisad.v2i1.11899

Ramzan, S., Iqbal, M. M., & Kalsum, T. (2022). Text-to-Image Generation Using Deep Learning †. *Engineering Proceedings*, *20*(1), 1–6. https://doi.org/10.3390/engproc2022020016

Sasongko, T. B., Haryoko, H., & Amrullah, A. (2023). Analisis Efek Augmentasi Dataset dan Fine Tune pada Algoritma Pre-Trained Convolutional Neural Network (CNN). *Jurnal Teknologi Informasi dan Ilmu Komputer*, *10*(4), 763–768. https://doi.org/10.25126/jtiik.20241046583

Sony Maulana, M., Nurmalasari, Rheno Widianto, S., Dewi Ayu Safitri, S., & Maulana, R. (2023). Pelatihan Chat Gpt Sebagai Alat Pembelajaran Berbasis Artificial Intelligence Di Kelas. *Jurnal Penelitian dan Pengabdian Masyarakat Jotika*, *3*(1), 16–19. https://doi.org/10.56445/jppmj.v3i1.103

Sultan, Y., Ma, J., & Liao, Y.-Y. (2024). *Fine-Tuning Stable Diffusion XL for Stylistic Icon Generation: A Comparison of Caption Size*.

Sumarna, H. B., Utami, E., & Hartanto, A. D. (2021). Tinjauan Literatur Sistematik tentang Structural Similarity Index Measure untuk Deteksi Anomali Gambar. *Creative Information Technology Journal*, *7*(2), 75. https://doi.org/10.24076/citec.2020v7i2.248

Wahmuda, F., & Junaidi Hidayat, M. (2020). Redesain Logo dan Media Promosi sebagai. *Jurnal Desain Komunikasi Visual & Multimedia*, *6*(2), 147–159. http://publikasi.dinus.ac.id/index.php/andharupa

Xiao, S., Wang, L., Ma, X., & Zeng, W. (2024). TypeDance: Creating Semantic Typographic Logos from Image through Personalized Generation. *Conference on Human Factors in Computing Systems - Proceedings*, 1–24. https://doi.org/10.1145/3613904.3642185

Zahara, R. L. (2024). Strategi Branding Logo dan Kemasan Produk Makanan Tradisional “Enye Ibu Iya.” *Ars: Jurnal Seni Rupa dan Desain*, *27*(1), 7–12. https://doi.org/10.24821/ars.v27i1.6757