ARTIFICIAL FINGERPRINTING BERBASIS *CONVOLUTIONAL NEURAL NETWORK* UNTUK ATRIBUSI GAMBAR WAJAH SINTETIK DARI STYLEGAN2

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

DISUSUN OLEH:

ALVIN YUSUF RIZIQ
NIM 32602100002

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG
2025

ARTIFICIAL FINGERPRINTING BASED ON CONVOLUTIONAL NEURAL NETWORK FOR THE ATTRIBUTION OF SYNTHETIC FACIAL IMAGES GENERATED BY STYLEGAN2

FINAL PROJECT

Proposed to complete the requirement to obtain a bachelor's degree (S1)
at Informatics Engineering Departement of Industrial Technology Faculty
Sultan Agung Islamic University

ARRANGED BY:

ALVIN YUSUF RIZIQ
32602100002

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

LEMBAR PENGESAHAN TUGAS AKHIR

***ARTIFICIAL FINGERPRINTING* BERBASIS *CONVOLUTIONAL NEURAL NETWORK* UNTUK ATRIBUSI GAMBAR WAJAH SINTETIK DARI STYLEGAN2**

ALVIN YUSUF RIZIQ

NIM 32602100002

Telah dipertahankan di depan tim penguji ujian sarjana tugas akhir
Program Studi Teknik Informatika
Universitas Islam Sultan Agung
Pada tanggal :...26 Juni 2025...

TIM PENGUJI UJIAN SARJANA :

Imam Much Ibnu Subroto, S.T., M.Sc., Ph.D NIDN. 0613037301 (Ketua Penguji)	.....	3-7-2025
Ghufron, S.T., M.Kom NIDN. 0609108802 (Anggota Penguji)	.....	.....
Ir. Sri Mulyono, M. Eng NIDN. 0626066601 (Pembimbing)	.....	.....

Semarang, 30 Juni 2025
Mengetahui,
Kaprodi Teknik Informatika
Universitas Islam Sultan Agung

Moch. Taufik, S.T., MIT
NIK. 210604034

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama	:	Alvin Yusuf Riziq
NIM	:	32602100002
Judul Tugas Akhir	:	*ARTIFICIAL FINGERPRINTING* BERBASIS *CONVOLUTIONAL NEURAL NETWORK* UNTUK ATRIBUSI GAMBAR WAJAH SINTETIK DARI STYLEGAN2

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, 23-07-2025

Yang Menyatakan,

Alvin Yusuf Riziq

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama	: Alvin Yusuf Riziq
NIM	: 32602100002
Program Studi	: Teknik Informatika
Fakultas	: Teknologi industri
Alamat Asal	: Demak, Jawa Tengah

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : "*Artificial Fingerprinting* Berbasis *Convolutional Neural Network* Untuk Atribusi Gambar Wajah Sintetik Dari Stylegan2" 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, 23-07-2025

Yang menyatakan,

Alvin Yusuf Riziq

KATA PENGANTAR

Segala puji dan syukur saya panjatkan ke hadirat Allah SWT atas limpahan rahmat dan karunia-Nya, sehingga saya dapat menyelesaikan Tugas Akhir ini dengan judul "*Artificial Fingerprinting* Berbasis *Convolutional Neural Network* Untuk Atribusi Gambar Wajah Sintetik Dari Stylegan2". Tugas Akhir ini disusun sebagai salah satu syarat kelulusan dalam menempuh studi serta untuk memperoleh gelar sarjana (S-1) pada Program Studi Teknik Informatika, Fakultas teknologi Industri, Universitas Islam Sultan Agung Semarang.

Dalam penyusunan Tugas Akhir ini, saya mendapatkan banyak bantuan, baik dalam aspek materi maupun teknis dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, saya ingin mengucapkan terima kasih kepada :

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, SH., M.Hum yang telah memberikan kesempatan penulis menimba ilmu di kampus ini
2. Dekan Fakultas Teknologi Industri Ibu Dr. Ir. Hj. Novi Marlyana, S.T., M.T., IPU., ASEAN Eng
3. Dosen Pembimbing Bapak Ir. Sri Mulyono, M. Eng yang telah meluangkan waktu, membimbing, 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, 23-07-2025

Alvin Yusuf Riziq

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Model generatif seperti StyleGAN2 mampu menghasilkan citra sintetik yang menyerupai gambar nyata, namun menimbulkan risiko pelanggaran hak cipta akibat penggunaan data tanpa izin. Penelitian ini mengusulkan metode *artificial fingerprinting* berbasis *Convolutional Neural Network* (CNN) untuk menyisipkan jejak digital tersembunyi pada gambar wajah guna mendukung atribusi citra hasil *generate*. Sistem terdiri dari *encoder* dengan arsitektur menyeruapai U-Net untuk penyisipan dan *decoder* untuk ekstraksi *fingerprint*. Dengan *dataset* FFHQ, evaluasi menggunakan *Binary Cross Entropy* dan *Mean Squared Error* menunjukkan bahwa metode ini efektif dalam menyisipkan dan mendeteksi *fingerprint* tanpa mengganggu kualitas visual gambar. Pendekatan ini berpotensi menjadi solusi teknis dalam perlindungan hak cipta pada pengembangan AI generatif.

Kata kunci: *Artificial Fingerprinting*, CNN, StyleGAN2, Atribusi Gambar, Hak Cipta.

ABSTRACT

Generative models such as StyleGAN2 are able to generate synthetic images that resemble real images, but pose a risk of copyright infringement due to unauthorized use of data. This research proposes a Convolutional Neural Network (CNN)-based artificial fingerprinting method to insert hidden digital traces in facial images to support the attribution of generated images. The system consists of an encoder with U-Net-like architecture for insertion and a decoder for fingerprint extraction. With the FFHQ dataset, evaluation using Binary Cross Entropy and Mean Squared Error shows that the method is effective in inserting and detecting fingerprints without compromising the visual quality of the image. This approach has the potential to be a technical solution for copyright protection in generative AI development.

Keywords: Artificial Fingerprinting, CNN, StyleGAN2, Image Attribution, Copyright.

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan pesat dalam bidang *Artificial Intelligence* (AI) telah mengubah berbagai aspek kehidupan modern, mulai dari otomasi industri, sistem rekomendasi, hingga teknologi pencitraan digital (Handoko dkk., 2024). Salah satu cabang *Artificial Intelligence* yang berkembang pesat adalah *Machine Learning* (ML), terutama *Deep Learning*, yang memungkinkan komputer untuk secara otomatis mempelajari dan memahami representasi data yang kompleks (Pratama dkk., 2025).

Salah satu kemajuan penting dalam *deep learning* adalah kemampuan untuk menghasilkan data sintetik yang menyerupai data nyata, khususnya melalui teknologi *Generative Adversarial Networks* (GAN) (Lu dkk., 2023). GAN yang diperkenalkan oleh Ian Goodfellow pada tahun 2014 menggabungkan dua jaringan saraf (*generator* dan *discriminator*) yang saling berkompetisi dalam proses pembelajaran, sehingga mampu menghasilkan citra sintetis yang sangat realistis (Goodfellow dkk., 2014).

Salah satu model GAN yang signifikan dan banyak digunakan adalah StyleGAN, yang mampu menghasilkan gambar berkualitas tinggi seperti wajah manusia, mobil, dan lanskap, yang tampak hampir nyata (Talib & Abed, 2023). Melalui teknik *Adaptive Discriminator Augmentation* (ADA), model ini mampu mengatasi tantangan keterbatasan data dan memperbaiki stabilitas pelatihan, menghasilkan citra sintetis yang semakin sulit dibedakan dari citra nyata (de Meira dkk., 2023).

Namun, kemajuan ini juga menimbulkan tantangan baru, terutama dalam hal penyalahgunaan teknologi. Model generatif seperti StyleGAN2 telah dimanfaatkan dalam pembuatan konten manipulatif seperti *deepfake*, yang dapat mengaburkan batas antara realitas dan rekayasa digital, serta berpotensi menimbulkan ancaman terhadap privasi, reputasi, dan keamanan informasi (Talib & Abed, 2023). Disisi lain, pelatihan model generatif sering

kali menggunakan *dataset* yang mengandung karya berhak cipta tanpa izin pemiliknya, dan model yang dilatih tersebut dapat menghasilkan citra baru yang menyerupai karya asli, sehingga menimbulkan potensi pelanggaran hak cipta dan plagiarisme visual (Franceschelli dan Musolesi, 2022).

Ketiadaan sistem atribusi atau penelusuran jejak digital yang melekat pada citra hasil model generatif menjadikan pelanggaran semacam ini sulit dibuktikan dan ditegakkan secara hukum (Franceschelli & Musolesi, 2022). Dalam praktik dunia nyata, banyak kreator digital seperti fotografer, seniman, dan desainer memiliki karya visual yang telah dilisensikan dengan hak royalti atas nama mereka. Karya-karya ini secara hukum berada dalam cakupan hak cipta dan setiap penggunaannya, terutama untuk tujuan komersial atau pengembangan teknologi seperti pelatihan model AI, semestinya melalui izin atau kompensasi sesuai kesepakatan lisensi.

Namun, penyalahgunaan gambar-gambar berhak cipta sebagai *dataset* pelatihan model generatif sering terjadi secara diam-diam dan sulit dibuktikan, karena model AI tidak menyimpan salinan gambar asli, dan hasil generate biasanya tidak identik secara visual dengan input aslinya. Untuk menjawab permasalahan ini, dibutuhkan pendekatan teknis yang mampu melindungi dan melacak penggunaan karya berhak cipta secara tersembunyi namun dapat diverifikasi.

Artificial fingerprinting hadir sebagai solusi untuk menyisipkan jejak digital (*artificial fingerprint*) ke dalam gambar orisinal sebelum didistribusikan. Jika kemudian suatu gambar hasil *generate* dari model AI ditemukan, *artificial fingerprint* yang tertanam dapat diekstraksi dan dibandingkan dengan *database fingerprint* milik pemilik asli. Dengan cara ini, kreator dapat membuktikan bahwa model AI tertentu telah dilatih menggunakan karyanya tanpa izin, sehingga dapat mendukung proses hukum dan klaim atas pelanggaran royalti dan hak cipta secara sah. Metode berbasis *Convolutional Neural Network (CNN)* memiliki potensi besar karena kemampuannya dalam mengenali pola spasial dan fitur tersembunyi dalam citra digital (Issn dkk., 2024). CNN dapat dimanfaatkan untuk menyisipkan

artificial fingerprint ke dalam data pelatihan serta mendeteksinya kembali dari gambar hasil generatif model.

Oleh karena itu, penelitian ini memfokuskan pada pengembangan dan evaluasi metode *artificial fingerprinting* berbasis CNN untuk melindungi *dataset* visual terhadap penyalahgunaan dalam pelatihan model StyleGAN2, tanpa mengubah kualitas visual gambar asli.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merancang metode penyisipan *artificial fingerprint* berbasis CNN pada dataset gambar wajah untuk mendukung atribusi gambar sintetis?
2. Bagaimana merancang metode deteksi *artificial fingerprint* berbasis CNN dari gambar hasil generate model StyleGAN2?

1.3 Pembatasan Masalah

Agar penelitian ini dapat terfokus dan diselesaikan sesuai waktu yang tersedia, maka penelitian ini dibatasi pada beberapa hal berikut:

1. Model generatif yang digunakan adalah StyleGAN2-ADA dengan konfigurasi tidak mengaktifkan fitur ADA saat *training*.
2. *Dataset* yang digunakan adalah FFHQ (*Flickr-Faces-HQ*) dalam versi thumbnail berukuran 128×128 piksel yang diperoleh dari *repository* resmi NVIDIA di github. *Dataset* ini tidak melalui proses *preprocessing* tambahan, karena ukuran dan formatnya telah sesuai untuk kebutuhan pelatihan model.
3. Penelitian ini menitikberatkan pada metode *artificial fingerprinting* berbasis CNN, bukan pada model generatif StyleGAN2 yang hanya digunakan sebagai penguji efektivitas metode.
4. Penelitian ini tidak membahas aspek keamanan terhadap serangan seperti *cropping,* kompresi gambar, atau penambahan noise.

5. Penelitian hanya mencakup gambar wajah statis.

1.4 Tujuan

Penelitian ini bertujuan untuk mengembangkan metode *artificial fingerprinting* berbasis *Convolutional Neural Network (CNN)* sebagai mekanisme perlindungan hak cipta dan royalti atas karya visual yang berisiko digunakan tanpa izin dalam pelatihan model generatif seperti StyleGAN2. Dengan menyisipkan *artificial fingerprint* ke dalam *dataset* secara tersembunyi namun detektabel, sistem yang dirancang diharapkan mampu mengidentifikasi penggunaan tidak sah dari suatu karya, bahkan setelah mengalami transformasi melalui proses pelatihan model AI.

Secara lebih spesifik, metode ini ditujukan untuk membantu pemilik karya yang telah melisensikan gambarnya dengan hak royalti, agar dapat melacak dan membuktikan jika karyanya digunakan secara ilegal dalam proses pelatihan AI. *Artificial fingerprint* yang tertanam dapat diekstraksi dari gambar hasil *generate* model generatif dan dicocokkan dengan *artificial fingerprint* asli milik kreator, sehingga memungkinkan sistem atribusi otomatis sekaligus pembuktian atas pelanggaran lisensi. Pendekatan ini memberikan solusi teknis terhadap ketidakjelasan perlindungan hukum atas *dataset* visual dalam era AI, serta mendukung transparansi dan keadilan dalam pengembangan teknologi generatif.

1.5 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat bagi kreator visual yang pemilik hak cipta. Dengan adanya sistem *artificial fingerprinting* berbasis CNN, kreator dapat melacak dan membuktikan penggunaan karya mereka dalam pelatihan model AI secara tidak sah, sehingga meningkatkan perlindungan hak cipta dan kesadaran terhadap penggunaan data secara etis.

1.6 Sistematika Penulisan

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

BAB I : PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang pemilihan judul tugas akhir "*Artificial Fingerprinting* Berbasis CNN untuk Atribusi Gambar Wajah Sintetik dari StyleGAN2". Rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI

Bab ini memuat penelitian-penelitian sebelumnya dan dasar teori yang berfungsi sebagai sumber atau alat dalam memahami permasalahan yang berkaitan dengan *Artificial Fingerprint*, *Convolutional Neural Network* (CNN) dan StyleGAN2.

BAB III : METODOLOGI PENELITIAN

Bab ini mengungkapkan proses tahapan - tahapan penelitian dimulai dari analisa kebutuhan sistem, kemudian perancangan sistem hingga *prototype* jadi dibuat.

BAB IV : HASIL DAN ANALISIS SISTEM

Bab ini menyajikan hasil penyisipan *artificial fingerprint* ke dalam *dataset* dan deteksi *fingerprint* pada gambar hasil generate StyleGAN2. Hasil ini menunjukkan potensi metode dalam melindungi hak cipta kreator dengan membuktikan bahwa karya mereka digunakan dalam pelatihan model generatif.

BAB V : KESIMPULAN DAN SARAN

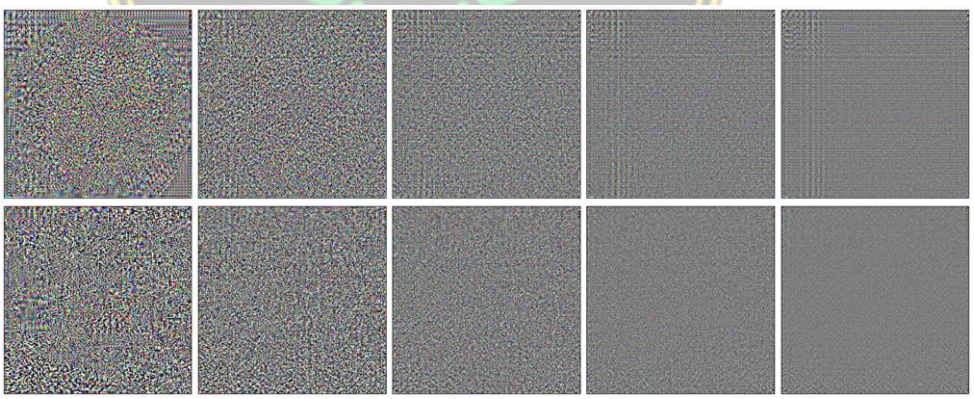
Bab terakhir memuat kesimpulan isi dari keseluruhan uraian bab-bab sebelumnya dan saran-saran dari hasil yang diperoleh dan diharapkan dapat bermanfaat dalam pengembangan selanjutnya

BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Upaya untuk mengidentifikasi asal-usul gambar yang dihasilkan oleh model generatif seperti *Generative Adversarial Networks* (GAN) telah menjadi salah satu fokus utama dalam bidang forensik digital. Salah satu pendekatan yang digunakan untuk mendukung proses ini adalah *fingerprinting*, yaitu teknik yang memungkinkan sistem mendeteksi dan mengatribusikan gambar sintetik berdasarkan ciri khas statistik yang dimilikinya (Yu dkk., 2021).

Penelitian awal dalam topik atribusi citra sintetis dilakukan oleh (Marra dkk., 2019) yang menunjukkan bahwa model generatif seperti ProGAN dan CycleGAN meninggalkan jejak statistik unik dalam gambar yang dihasilkan, mirip dengan pola noise sensor kamera digital PRNU (*Photo Response Non-Uniformity*). Melalui analisis terhadap noise residu (*residual noise patterns*), mereka menemukan bahwa setiap arsitektur GAN menghasilkan pola yang konsisten.



Gambar 2. 1 Cycle-GAN o2a (atas) dan Pro-GAN *kitchen* (bawah) perkiraan *fingerprints* dengan 2, 8, 32, 128, 512 residual (Marra dkk., 2019)

Pada gambar 2.1 menunjukkan bahwa semakin banyak residual yang digunakan, estimasi *fingerprint* menjadi lebih stabil dan menyerupai pola periodik yang khas dari masing-masing arsitektur GAN. Dengan menggunakan teknik korelasi silang (*cross-correlation*), mereka berhasil

mengidentifikasi model asal dari gambar baru dengan akurasi tinggi. Penelitian ini membuka peluang besar untuk melakukan atribusi, tanpa perlu memodifikasi proses pelatihan model secara eksplisit.

Menindaklanjuti penemuan tersebut, (Yu dkk., 2019) mengembangkan metode atribusi berbasis *fingerprint* yang lebih sistematis dan yang memperkenalkan konsep bahwa *fingerprint* GAN bukan hanya berupa artefak visual, melainkan distribusi statistik halus yang ditanamkan secara alami oleh parameter pelatihan dan konfigurasi model. Temuan penting mereka menunjukkan bahwa *fingerprint* ini tetap stabil meskipun model dilatih pada *dataset* yang berbeda atau menerima *noise input* yang bervariasi, mengindikasikan keberadaan karakteristik internal yang dapat diandalkan untuk proses atribusi.

Sebagai lanjutan dari pendekatan pasif tersebut, (Yu dkk., 2021) kemudian memperkenalkan pendekatan aktif melalui konsep *Artificial Fingerprinting*, yakni metode untuk menyisipkan tanda identitas ke dalam data pelatihan model generatif secara disengaja. Mereka mengusulkan metode *training-rooted fingerprinting*, yaitu proses penyisipan *fingerprint* secara disengaja ke dalam data pelatihan menggunakan *encoder steganografi*. Selama proses pelatihan, *fingerprint* ini diteruskan secara alami ke dalam model, sehingga hasil gambar dari model akan secara inheren memuat identitas *fingerprint* yang sudah disisipkan. *Fingerprint* ini kemudian dapat diekstrak menggunakan *decoder* yang telah dilatih, tanpa mengubah struktur internal model generatif.

Metode ini terdiri dari empat tahap utama, yaitu: (1) pelatihan *encoder–decoder steganografi*, (2) penyisipan *fingerprint* ke dalam data pelatihan, (3) pelatihan model generatif dengan data yang telah disisipkan *fingerprint*, dan (4) proses atribusi otomatis gambar sintetik melalui ekstraksi *fingerprint*. *Fingerprint* yang disisipkan terbukti tidak merusak kualitas visual gambar, bersifat tak terlihat, dapat ditransfer secara stabil ke model, serta tahan terhadap berbagai transformasi gambar maupun modifikasi model. Hal ini

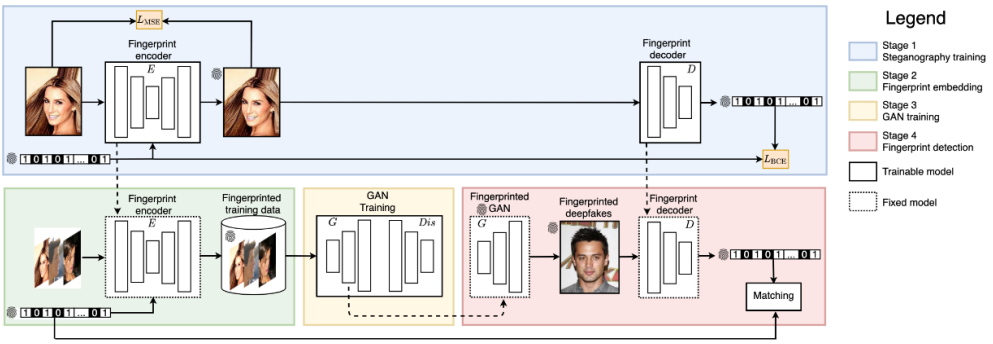
menjadikan *artificial fingerprinting* sebagai metode yang kuat, efisien, dan dapat diskalakan untuk sistem atribusi citra generative (Yu dkk., 2021).

Dalam pendekatan ini, arsitektur *Convolutional Neural Network (CNN)* memainkan peran sentral sebagai basis dari *encoder* dan *decoder*. CNN digunakan untuk menyisipkan *fingerprint* ke dalam data pelatihan secara tak terlihat dan sekaligus mendeteksi *fingerprint* tersebut dari gambar hasil generate. Kemampuan CNN dalam mengekstraksi fitur spasial dan pola statistik tersembunyi dari citra digital memungkinkan proses penyisipan dan pengenalan *fingerprint* dilakukan secara efektif dan presisi (Yu dkk., 2021).

2.2 Dasar Teori

2.2.1 *Artificial Fingerprint*

Artificial fingerprinting merupakan suatu teknik yang digunakan untuk menyematkan identitas atau jejak digital (*artificial fingerprint*) secara eksplisit ke dalam data pelatihan model *generative*, seperti *Generative Adversarial Networks* (GAN). Konsep ini pertama kali diperkenalkan oleh (Yu dkk., 2019) dalam penelitian mereka yang bertujuan untuk meningkatkan kemampuan atribusi terhadap citra sintetis, khususnya dalam konteks pelacakan asal-usul deepfake. Tidak seperti pendekatan atribusi pasif yang hanya mengandalkan artefak statistik alami dari model, *artificial fingerprint* bersifat aktif dan terencana, di mana sidik jari digital sengaja ditanamkan dalam proses pelatihan sehingga tercermin secara inheren dalam citra yang dihasilkan.



Gambar 2. 2 Alur Arteficial *Fingerprinting* untuk data pelatihan GAN (Yu dkk., 2021)

Teknik *artificial fingerprint* dilakukan dengan memodifikasi data pelatihan menggunakan *steganographic encoder*, yakni sebuah modul berbasis *Convolutional Neural Network* (CNN) yang bertugas menyisipkan *fingerprint* tersembunyi ke dalam citra pelatihan. dengan rancangan sistem *encoder* yang mampu menyisipkan serta *decoder* yang mampu mendeteksi kembali *fingerprint* dari dalam gambar.

Proses pelatihan sistem ini mengoptimalkan dua fungsi kerugian secara bersamaan, yaitu *Binary Cross Entropy (BCE) Loss* dan *Mean Squared Error (MSE) Loss*. *BCE Loss* digunakan untuk mengevaluasi akurasi *decoder* dalam memprediksi *artificial fingerprint* yang telah disisipkan, dengan cara membandingkan antara prediksi bit *ground truth* dengan *artificial fingerprint*. Sementara itu, *MSE Loss* digunakan untuk mengukur sejauh mana gambar hasil modifikasi *encoder* menyerupai gambar asli, sehingga memastikan bahwa penyisipan *artificial fingerprint* tidak menyebabkan distorsi visual yang signifikan. Tujuan dari kombinasi ini adalah agar *fingerprint* dapat dikenali kembali secara akurat (BCE), dengan tetap mempertahankan kualitas visual gambar asli (MSE). Dalam implementasinya, arsitektur *artificial fingerprint* terdiri dari dua komponen utama yang dilatih untuk menyisipkan *fingerprint* ke gambar dan untuk melakukan ekstraksi *fingerpirint* dari gambar, yaitu:

1. *Encoder*

   Encoder berfungsi menyisipkan vektor *fingerprint* ke dalam gambar. Arsitektur *encode*r dirancang menyerupai U-Net, yang terdiri dari:

   a. Jalur *downsampling (encoding)* dengan *convolutional layer* dengan *stride* untuk melakukan ekstraksi fitur spasial.
   b. Jalur *upsampling (decoding)* yang bertugas untuk membangun kembali fitur spasial dan menghasilkan residual untuk ditambahkan kedalam gambar asli melalui *skip connection.*
   c. *Skip connection* antar layer yang digunakan untuk menjaga informasi resolusi tetap tinggi selama decoding, seperti pada U-Net namun *output* dari *encoder* bukanlah segmentasi melainkan *residual image.*

2. *Decoder*

   Decoder berfungsi mendeteksi kembali *fingerprint* dari gambar hasil *generate* model GAN. Arsitektur *decoder* terdiri dari:

   a. Terdiri dari lapisan *convolutional* untuk mengekstrak fitur dari gambar.
   b. Dilanjutkan dengan *fully connected* untuk menghasilkan prediksi *fingerprint*.

Perbedaan utama dengan U-Net adalah pada tujuan dan outputnnya. U-Net digunakan untuk segmentasi citra (Hermawati dan Jaya, 2024). Sementa model ini ditujukan untuk menyisipkan informasi tersembunyi (*fingerprint*) dan untuk mendeteksinya Kembali.

Setelah model GAN dilatih menggunakan data yang telah disisipkan *fingerprint* tersebut, maka *fingerprint* akan terbawa dan muncul secara halus dalam citra sintetis hasil *generate*, meskipun tidak tampak secara visual. Pada tahap akhir, *fingerprint* dapat diekstraksi kembali menggunakan *decoder* yang telah dilatih sebelumnya untuk mengidentifikasi asal atau identitas dari model generatif.

Yu dkk., 2021 mendefinisikan empat tahapan utama dalam proses *artificial fingerprinting*, yaitu:

1. Pelatihan *Encoder* dan *Decoder* Steganografi

   Encoder menyisipkan *fingerprint* ke dalam citra, sementara *decoder* dilatih untuk mengekstraknya kembali.

2. Penyisipan *Fingerprint* pada Data Pelatihan

   Seluruh data pelatihan (*dataset*) yang digunakan untuk melatih model GAN dimodifikasi dengan *fingerprint* tertentu.

3. Pelatihan Model Generatif

   Model GAN dilatih menggunakan *dataset* yang telah dimodifikasi, menyebabkan *fingerprint* turut tertanam dalam parameter model.

4. Atribusi

   Citra hasil *generate* diuji menggunakan *decoder* untuk menentukan *fingerprint* dan mengaitkannya dengan identitas model pembuat.

Dengan demikian, *artificial fingerprint* menjadi solusi proaktif dalam mendukung upaya transparansi dan tanggung jawab dalam penggunaan model generatif, terutama dalam mendeteksi dan menanggulangi penyalahgunaan teknologi *deepfake* (Yu dkk., 2021).

2.2.2 Attribusi Gambar

Atribusi gambar sintetik merupakan proses untuk mengidentifikasi asal-usul gambar yang dihasilkan oleh model generatif, seperti *Generative Adversarial Networks* (GAN) atau model difusi teks-ke-gambar (Sha dkk., 2023). Dalam konteks teknologi *deep learning*, atribusi bertujuan untuk menentukan model atau sistem pembuat dari sebuah gambar digital (Yu dkk., 2021). Hal ini menjadi penting karena konten sintetik yang sangat realistis, seperti *deepfake*, dapat digunakan untuk menyesatkan publik, memalsukan identitas, atau merusak reputasi individu dan institusi (Marra dkk., 2019).

Atribusi dilakukan dengan menganalisis jejak digital tersembunyi (*fingerprint*) yang ditinggalkan oleh model generatif selama proses pelatihan atau inferensi. Berdasarkan pendekatannya, atribusi gambar dibagi menjadi dua kategori utama, yaitu atribusi pasif dan aktif (Yu dkk., 2021).

1. Atribusi Pasif, yang dilakukan dengan menganalisis pola statistik atau artefak halus yang secara alami tertanam dalam gambar hasil *generate*. (Marra dkk., 2019) menemukan bahwa gambar yang dihasilkan oleh GAN seperti ProGAN dan CycleGAN menyimpan pola residu yang dapat digunakan untuk mengenali arsitektur model pembuatnya.
2. Atribusi Aktif, yang dilakukan dengan menyisipkan *artificial fingerprint* secara sengaja ke dalam data pelatihan. Metode ini pertama kali diperkenalkan secara sistematis oleh (Yu dkk., 2021), yang menyarankan pelatihan *encoder* steganografi untuk menyematkan identitas ke dalam gambar pelatihan. *Fingerprint* ini kemudian terbawa ke dalam gambar hasil *generate* dan dapat dideteksi kembali menggunakan *decoder* khusus.

Pendekatan ini dikembangkan lebih lanjut dalam penelitian (Sha dkk., 2023), yang memperkenalkan *De-Fake*, sebuah sistem deteksi dan atribusi

untuk gambar yang dihasilkan oleh model difusi seperti DALL·E atau Stable Diffusion. Mereka menunjukkan bahwa *fingerprint* model dapat dikenali secara stabil meskipun gambar mengalami berbagai transformasi, seperti kompresi atau cropping.

2.2.3 Gambar Sintetik

Gambar sintetik merupakan citra digital yang dihasilkan secara buatan melalui proses komputasi, baik dengan cara memodifikasi gambar nyata maupun dengan menyintesis gambar sepenuhnya dari lingkungan virtual (Man dan Chahl, 2022). Gambar ini digunakan secara luas dalam pengembangan sistem *computer vision*, terutama untuk kebutuhan pelatihan model berbasis *deep learning* yang memerlukan data dalam jumlah besar (Man dan Chahl, 2022).

Gambar sintetik mencakup berbagai bentuk, mulai dari komposit gambar nyata dan objek buatan (*synthetic composite imagery*), hingga data visual yang sepenuhnya dibentuk dari lingkungan virtual tiga dimensi (*virtual synthetic imagery*) (Man dan Chahl, 2022). Keunggulan utama gambar sintetik adalah kemampuan untuk menghasilkan data dalam jumlah besar, dengan variasi dan label yang terkontrol secara otomatis, sehingga sangat menghemat waktu dan biaya dibandingkan pengumpulan data nyata (Man dan Chahl, 2022).

Sementara itu, Madhusudana dkk., 2022 menunjukkan bahwa gambar sintetik juga dapat digunakan secara efektif untuk pelatihan model tanpa memerlukan gambar nyata. Dalam studi mereka, model penilaian kualitas gambar (*Image Quality Assessment* atau IQA) dilatih secara eksklusif menggunakan gambar sintetik yang telah diberi berbagai jenis distorsi. Hasilnya, model mampu menilai kualitas gambar nyata secara akurat, membuktikan bahwa data sintetik dapat memberikan representasi visual yang cukup kuat untuk ditransfer ke domain dunia nyata atau dikenal dengan istilah *domain generalization*.

2.2.4 StyleGAN2

StyleGAN merupakan sebuah arsitektur *Generative Adversarial Network* (GAN) yang diperkenalkan oleh (Karras dkk., 2019) dan menawarkan pendekatan baru dalam proses sintesis citra dengan memperkenalkan pemisahan antara representasi laten dan output visual. Tidak seperti GAN konvensional yang mengubah vektor laten langsung melalui *generator*, StyleGAN menggunakan jaringan bernama *mapping network* untuk mengubah vektor laten z menjadi vektor baru w, yang kemudian digunakan untuk mengontrol berbagai tingkat gaya (*style*) dalam *generator* melalui mekanisme *adaptive instance normalization* (AdaIN). Mekanisme ini memberikan kontrol yang lebih halus terhadap fitur-fitur visual pada berbagai skala, seperti pose, bentuk wajah, dan detail tekstur.

Arsitektur ini memperkenalkan konsep *style mixing* yang memungkinkan interpolasi gaya dari dua vektor berbeda dalam berbagai lapisan, serta *stochastic variation* untuk menambahkan detail acak yang meningkatkan keragaman *output*. Dengan pendekatan ini, StyleGAN mampu menghasilkan citra sintetis dengan variasi dan realisme yang tinggi, menjadikannya terobosan penting dalam bidang sintesis citra berbasis pembelajaran mendalam (Karras dkk., 2019).

Meskipun StyleGAN mampu menghasilkan gambar sintetis berkualitas tinggi, model StyleGAN masih menghasilkan sejumlah artefak visual seperti *droplet-like artifacts* yang mengganggu realisme citra. Untuk mengatasi hal ini, (Karras dkk., 2020) mengembangkan StyleGAN2 sebagai penyempurnaan dari versi sebelumnya. Perubahan utama yang diperkenalkan dalam StyleGAN2 mencakup:

1. Penghapusan *adaptive instance normalization* (AdaIN)

   Mekanisme normalisasi adaptif pada StyleGAN sebelumnya diganti untuk mengurangi artefak dan meningkatkan stabilitas pelatihan.
2. Redesain *generator*

   StyleGAN2 memisahkan pengendalian gaya dari proses normalisasi, memungkinkan kontrol yang lebih alami terhadap fitur visual.
3. Peningkatan struktur *discriminator*

   Struktur jaringan *adversarial* ditingkatkan untuk mendukung citra beresolusi tinggi tanpa menyebabkan ketidakstabilan pelatihan.
4. *Path length regularization*

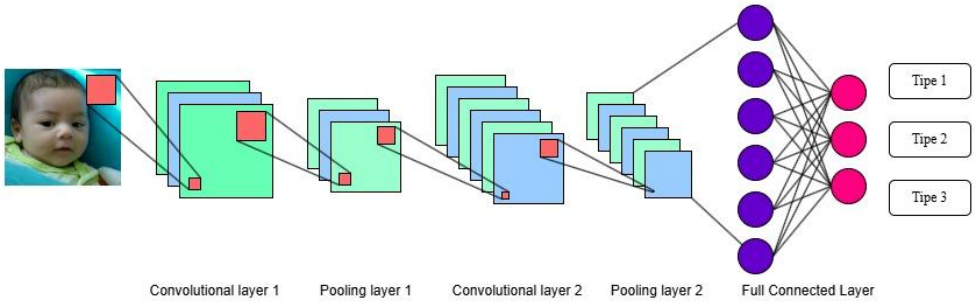
   Digunakan untuk menjaga konsistensi distribusi gaya terhadap perbedaan vektor laten, menghasilkan transisi visual yang lebih halus dan terkontrol.

Dengan peningkatan tersebut, StyleGAN2 mampu mengurangi artefak visual secara signifikan.

2.2.5 *Convolutional Neural Network* (CNN)

Convolutional Neural Network (CNN) adalah salah satu jenis arsitektur jaringan saraf tiruan yang dirancang secara khusus untuk mengolah data yang memiliki struktur *grid* seperti gambar dua dimensi. CNN diperkenalkan secara sistematis oleh (Lecun dkk., 1998) melalui penerapannya dalam sistem pengenalan tulisan tangan, terutama untuk membaca angka dan huruf dalam dokumen.

CNN bekerja berdasarkan prinsip bahwa fitur lokal dari data citra dapat diekstraksi melalui operasi konvolusi dan pooling. Arsitektur CNN umumnya terdiri atas tiga jenis lapisan utama, yaitu, *Convolutional Layer, Pooling Layer* dan *Fully Connected Layer*.

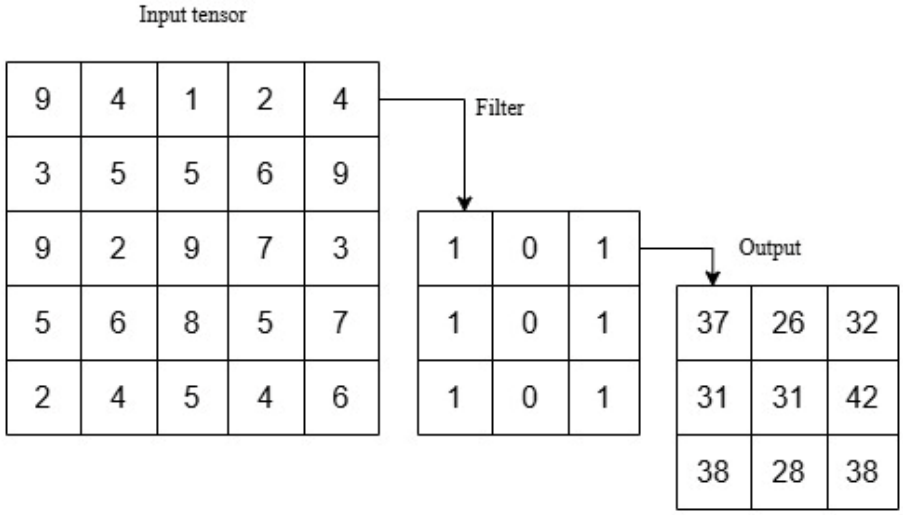


Gambar 2. 3 Arsitektur dasar CNN

Pada gambar 2.3 adalah tampilan arsitektur dasar CNN yang memiliki tiga jenis lapisan utama, yaitu: *Convolutional Layer*, *Pooling Layer*, dan *Fully Connected Layer* (*FC Layer*).

1. *Convolutional Layer*

   Convolutional layer merupakan salah satu komponen utama dalam arsitektur *Convolutional Neural Network* (CNN) yang berfungsi untuk mengekstraksi fitur dari *input* citra. Lapisan ini melakukan proses konvolusi terhadap hasil dari lapisan sebelumnya dengan menggunakan sejumlah *filter* atau kernel. *Filter-filter* tersebut memiliki bobot yang diinisialisasi secara acak dan akan dilatih selama proses pembelajaran. Tujuan utama dari operasi konvolusi ini adalah untuk membentuk representasi fitur yang relevan dengan struktur spasial pada data *input*. Proses konvolusi menghasilkan transformasi linier terhadap *input*, yang disesuaikan dengan pola-pola spasial tertentu dalam citra. Bobot pada *convolutional layer* menentukan bentuk kernel konvolusi, dan melalui proses pelatihan, kernel ini akan belajar menyesuaikan dirinya terhadap karakteristik data sehingga dapat mengekstrak informasi penting secara efektif (Alwanda dkk., 2020).



Gambar 2. 4 *Convolutional layer*

Gambar 2.4 menggambarkan konsep dasar *convolutional layer* yang bekerja dengan menggunakan kernel untuk melakukan proses konvolusi pada *tensor input*.

Rumus dimensi *output convolutional layer:*

$$Output\ size = \left(\frac{N + 2p - F}{S}\right) + 1 \quad (1)$$

N = Ukuran *input tensor* (h x w)

p = *Padding*

F = Ukuran *filter* (h x w)

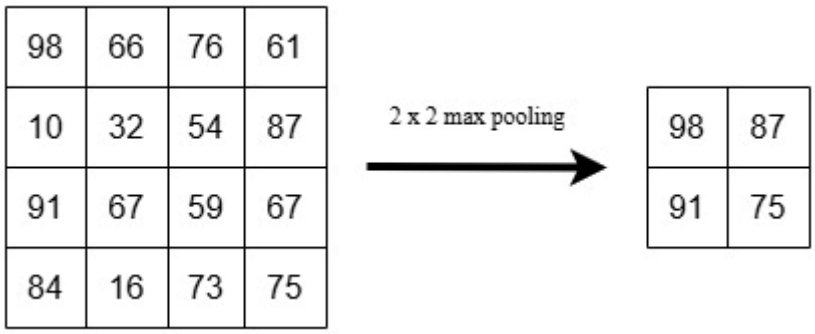
S = *Stride* adalah langkah pergeseran *filter* saat menggeser *filter* pada *input tensor*.

2. *Pooling layer*

Pooling layer merupakan lapisan yang digunakan untuk memproses *feature maps* dengan menerapkan operasi statistik terhadap nilai-nilai piksel dalam wilayah lokal tertentu. Dalam arsitektur CNN, lapisan ini biasanya ditempatkan secara berkala setelah satu atau beberapa lapisan konvolusi. Fungsinya adalah untuk mereduksi dimensi spasial dari *feature maps*, yang secara tidak langsung mengurangi jumlah parameter dan beban komputasi jaringan, serta membantu mencegah terjadinya *overfitting*. Pooling dilakukan pada setiap *feature map* secara terpisah

dengan menggeser sebuah *filter* berukuran tertentu, seperti 2x2, menggunakan langkah (*stride*) tertentu. Operasi ini menghasilkan versi *feature map* yang lebih ringkas (Peryanto dkk., 2020).

Terdapat dua jenis pooling yang umum digunakan, yaitu *Max Pooling* dan *Average Pooling*. Pada *Max Pooling*, nilai maksimum dari wilayah lokal akan dipilih sebagai *output* dari *filter* tersebut. Sebaliknya, *Average Pooling* akan menghitung dan menggunakan nilai rata-rata dari wilayah lokal tersebut. Dengan melakukan *downsampling* melalui *pooling layer*, jaringan menjadi lebih efisien secara komputasi dan lebih tangguh terhadap variasi kecil pada posisi fitur dalam citra (Peryanto dkk., 2020).



Gambar 2. 5 *Max pooling*

Gambar 2.5 menggambarkan konsep *max pooling* dengan ukuran kernel 2x2 dan *stride* 2.

Rumus *max pooling*, jika input *feature map* berukuran:

$$n_h \times n_w \times n_c$$

n_h = tinggi dari *feature map*

n_w = lebar dari *feature map*

n_c = jumlah channel

Maka dimensi output pooling layer adalah

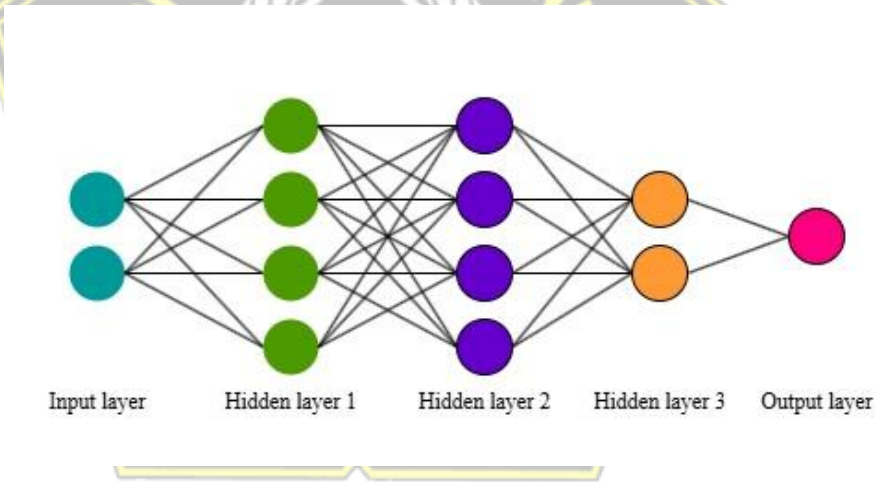
$$Output\ size = \left\lfloor \frac{n_h - f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n_w - f}{s} + 1 \right\rfloor \times n_c \qquad (2)$$

f = Ukuran *filter pooling*

S = *Stride* adalah langkah pergeseran *filter* saat menggeser *filter* pada *input tensor*.

3. *Fully connected layer*

Fully connected layer merupakan lapisan dalam jaringan saraf tiruan yang berfungsi untuk mengubah representasi fitur hasil ekstraksi dari lapisan sebelumnya ke dalam bentuk yang sesuai untuk proses klasifikasi. Tidak seperti lapisan konvolusi, lapisan ini tidak melakukan operasi konvolusi, melainkan menggunakan operasi perkalian matriks yang dilanjutkan dengan penambahan nilai bias. Setiap neuron pada lapisan ini terhubung secara penuh dengan seluruh unit aktivasi dari lapisan sebelumnya, sehingga disebut sebagai *fully connected*. Hubungan penuh ini memungkinkan jaringan untuk menggabungkan semua informasi lokal yang telah diekstraksi oleh lapisan sebelumnya menjadi representasi global yang dapat digunakan untuk menghasilkan keputusan klasifikasi akhir (Alwanda dkk., 2020).



Gambar 2. 6 *Fully connected layer*

Gambar 2.6 adalah tampilan *fully connected layer* yang terdiri dari *input layer* yang bertugas untuk menerima masukan, *hidden layer* sebagai pemrosesan data, dan *output layer* yang digunakan untuk menghasilkan prediksi

Rumus *fully connected layer*:

$$Output = W \times x + b \quad (3)$$

W = Bobot dari koneksi antar neuron

x = Jumlah kernel

b = Jumlah bias

Keunggulan utama CNN dibandingkan jaringan saraf tradisional terletak pada kemampuannya untuk secara otomatis mempelajari hierarki fitur dari *input* gambar tanpa memerlukan ekstraksi fitur manual. CNN juga dapat memanfaatkan parameter sharing pada *filter* konvolusi untuk mengurangi jumlah parameter, sehingga lebih efisien dalam proses pelatihan (Lecun dkk., 1998).

2.2.6 *Binary Cross Entropy (BCE)*

Dalam arsitektur *artificial fingerprinting*, salah satu tujuan utama adalah memastikan bahwa *artificial fingerprint* yang telah disisipkan ke dalam data pelatihan melalui *encoder* dapat dideteksi secara akurat oleh *decoder*. Untuk mengukur tingkat kesesuaian antara fingerprint asli dan hasil prediksi oleh *decode*r, digunakan fungsi kerugian *Binary Cross Entropy* (BCE) *Loss* (Yu dkk., 2021).

Binary Cross Entropy (BCE) *Loss* merupakan fungsi kerugian yang umum digunakan dalam berbagai sistem klasifikasi biner, termasuk dalam pengukuran akurasi prediksi *artificial fingerprint* yang direpresentasikan sebagai bit biner (0 dan 1). *BCE Loss* menghitung jarak antara distribusi target dan distribusi prediksi model dalam bentuk probabilitas, serta memberikan penalti yang lebih besar jika prediksi jauh dari nilai target (Terven dkk., 2025). *BCE Loss* dalam *artificial fingerptrint* dirumuskan sebagai berikut:

$$L_{BCE}(\tilde{x}, w; E, D) = \frac{1}{n}\sum_{k=1}^{n}(w_k \log \hat{w}_k + (1 - w_k)\log(1 - \hat{w}_k)) \quad (4)$$

D = *Decoder*

E = *Encoder*

$\tilde{x}, w$ = citra hasil *embed* dengan *artificial fingerprint*

n = jumlah bit *artificial fingerprint*

w_k = bit ke-k dari *artificial fingerprint ground truth*

$\hat{w}_k$ = bit ke-k hasil prediksi *decoder*

Nilai Binary Cross Entropy (BCE) Loss yang semakin kecil menunjukkan bahwa prediksi *artificial fingerprint* oleh *decoder* semakin akurat. Hal tersebut mengindikasikan bahwa model berhasil mengekstraksi kembali informasi *artificial fingerprint* yang tersembunyi dari gambar hasil generatif, meskipun *artificial fingerprint* tersebut awalnya hanya disisipkan pada tahap pelatihan. BCE menjadi pilihan yang tepat dalam konteks ini karena kemampuannya mengukur kesesuaian probabilistik pada tugas klasifikasi biner, serta sensitivitasnya terhadap kesalahan prediksi yang terlalu percaya diri (Terven dkk., 2025).

2.2.7 *Mean Squared Error (MSE)*

Dalam arsitektur *artificial fingerprinting*, selain memastikan *artificial fingerprint* dapat dideteksi dengan akurasi tinggi oleh *decoder*, aspek penting lainnya adalah menjaga kualitas visual dari citra yang telah dimodifikasi oleh *encoder*. Modifikasi ini terjadi karena proses penyisipan *artificial fingerprint* ke dalam citra asli. Untuk memastikan bahwa perubahan visual akibat penyisipan tidak menurunkan kualitas citra secara signifikan, digunakan fungsi kerugian *Mean Squared Error (MSE) Loss* (Yu dkk., 2021).

MSE Loss adalah salah satu fungsi kerugian yang paling umum digunakan pada tugas-tugas regresi dan rekonstruksi citra (Zhao dkk., 2021). Fungsi ini menghitung rata-rata kuadrat dari selisih nilai antara dua matriks (biasanya dua gambar), dan memberikan penalti proporsional terhadap besar kecilnya selisih tersebut (Terven dkk., 2025). Dalam konteks fingerprinting, MSE digunakan untuk membandingkan citra asli dengan citra hasil dari *encode*r yang telah disisipkan fingerprint.

MSE Loss dalam *artificial fingerptrint* dirumuskan sebagai berikut:

$$L_{MSE}(\tilde{x}, w; E) = \|E(\tilde{x}, w) - \tilde{x}\|_2^2 \quad (4)$$

E = *Encoder*

$\tilde{x}$ = citra asli

$\tilde{x}, w$ = citra hasil *embed artificial fingerprint*

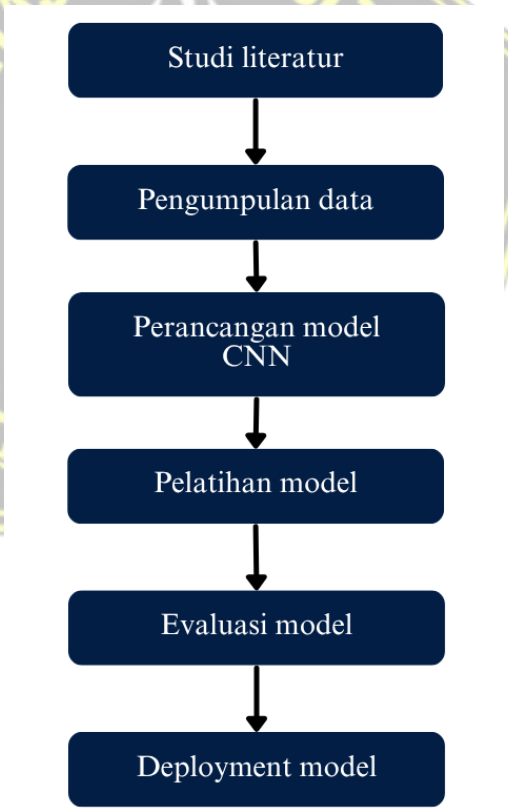
Nilai *MSE Loss* yang semakin kecil menunjukkan bahwa citra hasil modifikasi *encoder* sangat mirip dengan citra asli, yang berarti bahwa proses penyisipan *artificial fingerprint* tidak menyebabkan distorsi visual yang signifikan (Terven dkk., 2025). Hal ini sangat penting dalam konteks *invisible watermarking*, di mana *artificial fingerprint* seharusnya tidak mengganggu tampilan visual tetapi tetap dapat dikenali oleh sistem deteksi.

MSE Loss secara luas digunakan dalam berbagai aplikasi pembelajaran representasi dan pelestarian kualitas citra, karena sifatnya yang sensitif terhadap deviasi lokal dan mudah dihitung secara efisien (Terven dkk., 2025)terter. Dalam penelitian ini, penggunaan MSE membantu memastikan bahwa penyisipan *artificial fingerprint* tidak mengorbankan integritas visual gambar asli yang dipublikasikan.

BAB III
METODE PENELITIAN

3.1 Metode Penelitian

Dalam penelitian ini, metode yang digunakan adalah *Convolutional Neural Network* (CNN) dengan arsitektur yang menyerupai U-Net. Arsitektur ini terdiri atas dua komponen utama, yaitu *encoder* dan *decoder*. *Encoder* berfungsi untuk menyisipkan informasi *artificial fingerprint* ke dalam citra, sedangkan *decoder* bertugas mengekstraksi kembali informasi tersebut melalui proses konvolusi CNN guna mendeteksi *artificial fingerprint* yang telah disisipkan sebelumnya. Adapun tahapan yang harus dilakukan dalam penelitian ini ditunjukkan pada gambar *flowchart* 3.1.



Gambar 3. 1 Tahapan penelitian

1. Studi literatur

   Melakukan kajian teori terkait metode *Artificial Fingerprinting* untuk merumuskan dasar pendekatan yang digunakan.

2. Pengumpulan data

   Menggunakan *dataset* FFHQ yang tersedia di *repository* NVIDIA secara *open source* sebagai data *input*.

3. Perancangan model

   Membangun model *Artificial Fingerprint* yang berbasis CNN untuk menyematkan *fingerprint* ke dalam *dataset* FFHQ menggunakan Teknik stegastamp. Selama pelatihan kualitas gambar dan akurasi *fingerprint* akan dipertahankan menggunakan fungsi *Binary Cross Entropy Loss with Logits* (BCE) *loss* dan Mean *Squared Error Loss* (MSE) *loss.*

4. Pelatihan model

   Model dilatih selama 10 *epoch* menggunakan *dataset* FFHQ yang terdiri dari sekitar 70.000 citra wajah. Proses pelatihan bertujuan untuk mengoptimalkan *encoder* dan *decoder* agar mampu menyisipkan serta mengekstraksi *artificial fingerprint* secara akurat. Setiap pasangan gambar dan *artificial fingerprint* diproses oleh *encoder*, lalu hasilnya dievaluasi oleh *decoder*. Pelatihan menggunakan algoritma Adam dengan fungsi kerugian gabungan *Binary Cross Entropy* (BCE) untuk mengukur akurasi *artificial fingerprint* dan *Mean Squared Error* (MSE) untuk menjaga kualitas visual gambar.

5. Evaluasi model

   Evaluasi model dilakukan dengan menggabungkan *Binary Cross Entropy* (BCE) *Loss* untuk mengukur akurasi ekstraksi *artificial fingerprint* dalam bentuk *bit biner*, serta *Mean Squared Error* (MSE) *Loss* untuk menilai sejauh mana kualitas visual gambar tetap terjaga setelah penyisipan. Nilai BCE dan MSE yang rendah menunjukkan bahwa *artificial fingerprint* dapat disisipkan secara tersembunyi tanpa menimbulkan distorsi, dan tetap dapat dikenali secara akurat oleh sistem.

6. *Deployment*

   Model yang telah dilatih disimpan dalam format .pth dan dideploy melalui layanan API menggunakan FastAPI. *Backend* memuat model *encoder*, *decoder*, dan StyleGAN2 sebagai media uji, serta menyediakan *endpoint*

untuk proses penyisipan, ekstraksi, dan *generate* gambar. *Frontend* dikembangkan menggunakan ReactJS dengan Axios untuk komunikasi asinkron, dan Tailwind CSS untuk tampilan yang responsif. Sistem ini dirancang agar mudah digunakan bahkan oleh pengguna *non* teknis, dengan fitur unggah gambar, visualisasi proses, dan evaluasi model menggunakan metrik BCE, MSE, dan *Bitwise Accuracy*. Arsitektur ini menyederhanakan akses ke metode *artificial fingerprinting* secara praktis dan interaktif.

3.1.1 Studi Literatur

Pada tahap ini dilakukan peninjauan terhadap berbagai sumber literatur, termasuk *e-book*, artikel ilmiah, jurnal, skripsi terdahulu, serta referensi dari situs web terpercaya, dengan tujuan memperoleh pemahaman yang mendalam mengenai konsep *artificial fingerprinting*, arsitektur *Convolutional Neural Network* (CNN), dan model generatif StyleGAN2.

3.1.2 Pengumpulan Data

Data yang digunakan dalam penelitian ini berasal dari *dataset* publik Flickr-Faces-HQ (FFHQ), yakni Kumpulan 70.000 citra wajah beresolusi tinggi yang secara luas dimanfaatkan dalam pelatihan model generatif seperti StyleGAN. *Dataset* ini menyediakan beragam citra wajah manusia dengan variasi usia, ras, ekspresi, dan atribut lainnya, sehingga sangat sesuai untuk kebutuhan pelatihan model *Generative Adversarial Network* (GAN).

Dataset FFHQ bersifat sumber terbuka (*open source*) dan tersedia secara bebas untuk keperluan penelitian akademik melalui repositori resmi NVIDIA di GitHub. Dengan demikian, penelitian ini tidak memerlukan proses pengumpulan data primer melalui observasi langsung maupun eksperimen di lapangan.

3.1.3 Perancangan Model CNN

Model yang dikembangkan dalam penelitian ini terdiri dari dua komponen utama, yaitu *Encoder* dan *Decoder*, yang bekerja untuk

menyisipkan dan mengekstrak *fingerprint* dari gambar menggunakan konsep steganografi berbasis CNN:

a. *Encoder*

Encoder dirancang untuk menyisipkan *artificial fingerprint* ke dalam citra dengan memanfaatkan arsitektur *convolutional* yang kompleks, menggunakan struktur yang menyerupai U-Net, namun tanpa menggunakan *adversarial loss* seperti pada model generatif GAN. Struktur ini dipilih karena kemampuannya dalam mempertahankan informasi resolusi tinggi melalui mekanisme *skip connection*, yang menghubungkan fitur dari lapisan *encoder (downsampling)* awal ke lapisan *decoder (upsampling)* akhir, sehingga meminimalkan kehilangan informasi spasial selama proses *encoding*.

Proses *encoding* diawali dengan *artificial fingerprint* berdimensi tetap, berupa vektor biner yang kemudian diproyeksikan ke dalam representasi spasial melalui *fully-connected (dense) layer*. Hasil proyeksi ini kemudian dilakukan *reshape* menjadi *tensor* dua dimensi dan diperbesar melalui *upsampling* hingga mencapai dimensi spasial yang setara dengan citra *input*.

Setelah memiliki ukuran yang sesuai, representasi *artificial fingerprint* ini dikombinasikan secara langsung dengan citra asli melalui konkatenasi kanal (*channel-wise concatenation*), sehingga menghasilkan *tensor* gabungan yang kaya informasi. *Tensor* ini kemudian diproses oleh jaringan *convolutional* berlapis untuk menghasilkan citra baru yang tampak serupa dengan citra input namun telah mengandung *artificial fingerprint* tersembunyi di dalam struktur pikselnya. Tujuan dari desain ini adalah untuk memastikan bahwa citra hasil penyisipan tetap mempertahankan kualitas visual dan sekaligus menyimpan *artificial fingerprint* secara tersembunyi namun dapat diambil kembali secara akurat oleh *decoder*.

Encoder yang dikembangkan dalam penelitian ini memiliki dua jalur utama dalam arsitekturnya, yaitu jalur *downsampling* dan jalur

upsampling, yang secara berurutan berfungsi untuk mengekstraksi dan merekonstruksi fitur citra selama proses penyisipan *artificial fingerprint*. Arsitektur ini dirancang agar mampu menyisipkan informasi tersembunyi (*artificial fingerprint*) ke dalam citra dengan tetap mempertahankan tampilan visual yang sangat mirip dengan citra aslinya, sehingga tidak menimbulkan perbedaan visual yang signifikan bagi pengamat manusia. Secara umum, struktur *encoder* dapat dijelaskan sebagai berikut:

1. Blok *Downsampling* (ekstraksi fitur)

   Bagian awal *encoder* terdiri dari lima blok konvolusional bertingkat yang berfungsi untuk mengekstraksi fitur dari citra secara bertahap. Setiap blok terdiri dari lapisan konvolusional dengan kernel berukuran kecil, diikuti oleh fungsi aktivasi *non linear* (ReLU). Proses ini diiringi dengan pengurangan resolusi spasial melalui operasi *strided convolution*, sehingga fitur yang dihasilkan bersifat lebih kompak dan mengandung informasi semantik tinggi. Tahapan ini memungkinkan jaringan untuk memahami struktur global dari citra yang akan menjadi tempat penyisipan *artificial fingerprint*.

2. Blok *Upsampling* (rekonstruksi citra)

   Setelah representasi fitur terbentuk, citra diproses melalui empat blok konvolusional untuk *upsampling*. Setiap blok bertugas mengembalikan dimensi spasial citra secara bertahap melalui teknik *upsampling* diikuti konvolusi. Dalam setiap tahap upsampling, digunakan *skip connection* dari *layer-layer* yang bersesuaian di jalur *downsampling*. Mekanisme ini diadaptasi dari arsitektur U-Net, dan berfungsi untuk membawa kembali informasi spasial beresolusi tinggi yang telah dipelajari pada tahap awal, sehingga hasil rekonstruksi tetap mempertahankan detail visual penting.

3. Blok Rekonstruksi dan *Output* Residual

   Setelah proses *upsampling* selesai, dilakukan tahap rekonstruksi akhir menggunakan blok konvolusional tambahan. Tujuannya adalah untuk memurnikan hasil akhir dan menghasilkan *output* residual, yaitu

perbedaan halus antara citra asli dan citra yang mengandung *artificial fingerprint*. *Output* residual ini kemudian dijumlahkan secara elemen-per-elemen (element-wise addition) dengan citra asli untuk membentuk citra tersisip (stego image). Teknik residual ini memastikan bahwa perubahan pada citra tetap minimal, sehingga kualitas visual tetap terjaga, namun *artificial fingerprint* tetap tertanam secara tersembunyi dalam struktur piksel citra.

Arsitektur ini secara keseluruhan dirancang agar proses penyisipan *artificial fingerprint* tidak hanya bersifat imperseptibel (tidak terlihat secara kasat mata), tetapi juga tetap dapat diekstraksi kembali secara akurat oleh *decoder*, meskipun citra tersisip mengalami transformasi tertentu. Hal ini menjadikan sistem yang dibangun memiliki potensi tinggi dalam mendukung atribusi digital dan perlindungan hak cipta pada citra sintetik.

b. *Decoder*

Decoder berfungsi sebagai komponen utama dalam sistem untuk mengekstraksi kembali *artificial fingerprint* yang telah disisipkan secara tersembunyi ke dalam citra oleh *encoder*. Tujuan utama dari model ini adalah untuk merekonstruksi kembali *artificial fingerprint* dengan tingkat akurasi setinggi mungkin.

Arsitektur *decoder* bersifat konvolusional murni (*purely convolutional*), dirancang agar mampu menangkap dan mengolah pola-pola spasial yang menyimpan informasi tersembunyi dalam gambar. Struktur jaringan ini terdiri dari dua tahap utama, yaitu:

1. Lapisan-lapisan konvolusional bertingkat untuk downsampling

   Decoder diawali dengan beberapa blok konvolusional berukuran kernel kecil, masing-masing diikuti dengan aktivasi ReLU dan normalisasi batch. Lapisan-lapisan ini tidak hanya mengekstraksi fitur visual dari citra tersisip, tetapi juga secara bertahap mengecilkan ukuran spasial (*downsampling*) agar fokus jaringan mengarah pada informasi global dalam citra. Reduksi resolusi ini memungkinkan

model untuk menyaring sinyal *artificial fingerprint* yang mungkin tersembunyi dan tersebar secara halus di seluruh citra.

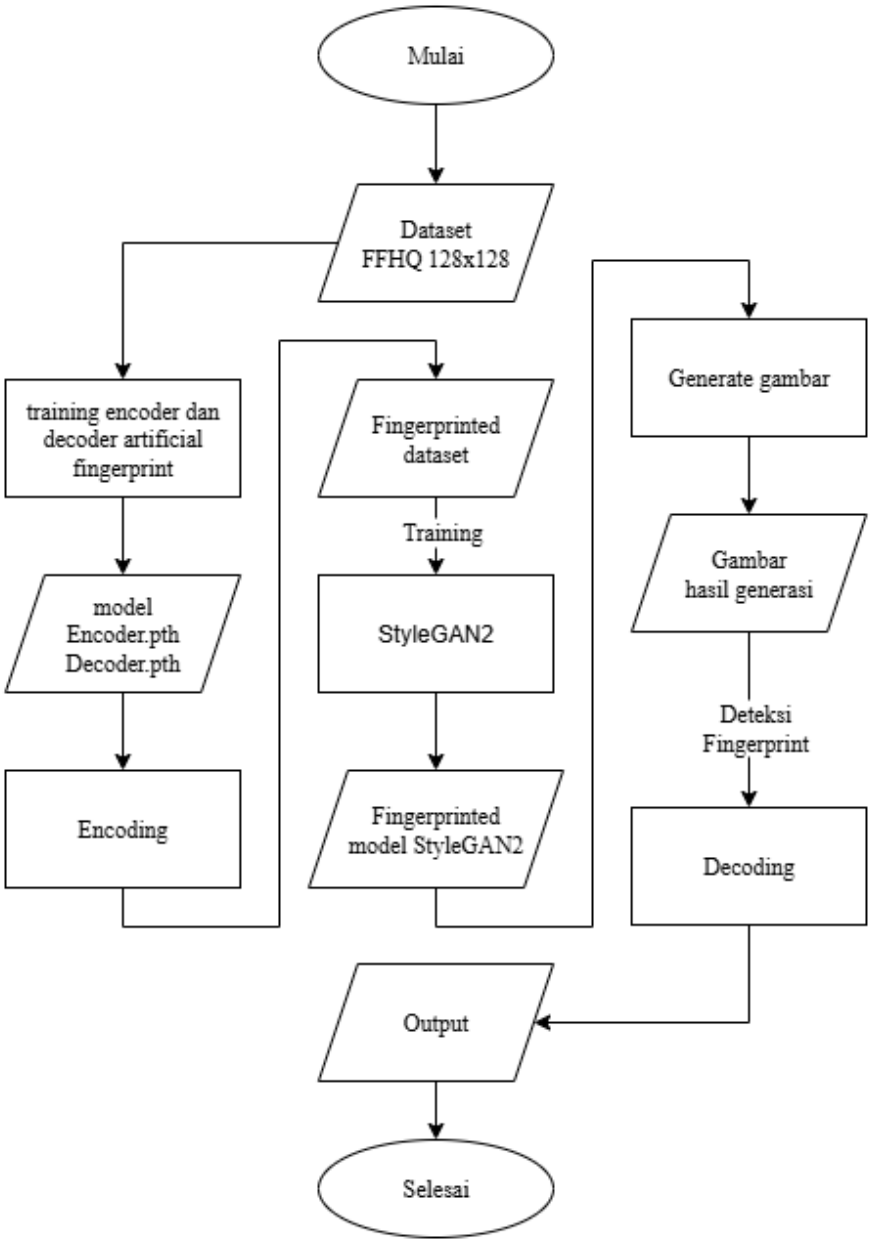
2. Lapisan fully connected (dense layer) untuk pemetaan akhir

   Setelah representasi fitur diperoleh dari tahap konvolusional, fitur-fitur tersebut kemudian diratakan (*flatten*) dan dimasukkan ke dalam *fully connected layer*, yang bertugas untuk memetakan fitur-fitur spasial tersebut ke dalam *artificial fingerprint* berdimensi tetap. *Artificial fingerprint* yang dihasilkan berada dalam format vektor biner, yang nantinya dibandingkan dengan *artificial fingerprint* asli untuk menghitung loss selama pelatihan.

Decoder ini dilatih secara *end-to-end* bersamaan dengan *encoder*, dengan fungsi kehilangan utama berupa *Binary Cross Entropy* (BCE) antara *artificial fingerprint* asli dan hasil prediksi, serta dapat ditambahkan regularisasi atau metrik evaluasi tambahan seperti *bitwise accuracy*. Tujuan dari desain ini adalah agar sistem mampu melakukan rekonstruksi *artificial fingerprint* secara presisi, bahkan dalam kondisi yang tidak ideal atau penuh gangguan visual. Dengan demikian, *decoder* memainkan peran penting dalam menjaga keandalan sistem *artificial fingerprinting* terhadap manipulasi citra di dunia nyata.

3.1.4 Pelatihan Model

Setelah tahap perancangan arsitektur model selesai, proses selanjutnya adalah melakukan pelatihan terhadap model encoder dan decoder dengan menggunakan dataset citra wajah beresolusi 128×128 piksel yang telah disiapkan sebelumnya. Secara umum, alur proses pelatihan ini dapat dilihat pada diagram flowchart 3.2



Gambar 3. 2 *flowchart* pelatihan model

Model yang telah dirancang dilatih selama 10 *epoch* menggunakan *dataset* FFHQ, yang berisi sekitar 70.000 citra wajah beresolusi tinggi. Proses pelatihan ini merupakan tahap penting untuk mengoptimalkan parameter pada *encoder* dan *decoder*, agar sistem mampu menyisipkan serta mendeteksi *artificial fingerprint* secara akurat.

Selama pelatihan, setiap *input* berupa pasangan gambar dan *artificial fingerprint* diproses melalui *encoder* untuk menghasilkan citra yang telah disisipkan *artificial fingerprint*, kemudian *decoder* bertugas mengekstraksi kembali *artificial fingerprint* dari hasil tersebut. Pelatihan dilakukan menggunakan optimisasi berbasis algoritma Adam, dengan fungsi kerugian yang menggabungkan *Binary Cross Entropy* (BCE) dan *Mean Squared Error* (MSE). BCE digunakan untuk mengukur akurasi prediksi *artificial fingerprint*, sementara MSE digunakan untuk menjaga agar kualitas visual gambar tetap mendekati gambar asli.

3.1.5 Evaluasi Model

Evaluasi model dilakukan melalui dua pendekatan utama yang saling melengkapi, dengan fokus pada penggunaan dua fungsi kerugian sebagai metrik utama, yaitu *Binary Cross Entropy* (BCE) *Loss* dan *Mean Squared Error* (MSE) *Loss*. Fungsi BCE *Loss* digunakan untuk mengevaluasi tingkat akurasi ekstraksi *artificial fingerprint* yang telah disisipkan oleh *encoder* ke dalam gambar wajah, kemudian diekstraksi kembali oleh *decoder* dalam bentuk *bit biner* (0 dan 1). Karena *artificial fingerprint* direpresentasikan sebagai vektor *biner*, BCE *Loss* dipilih karena secara khusus dirancang untuk mengukur kesesuaian distribusi probabilistik antara target *biner* dan hasil prediksi. Nilai BCE yang rendah mengindikasikan bahwa bit-bit hasil prediksi sangat mendekati nilai *ground truth*, yang berarti bahwa sistem mampu mengenali kembali *artificial fingerprint* secara akurat meskipun telah mengalami transformasi melalui proses konvolusi saat *encoding*.

Sementara itu, MSE *Loss* digunakan untuk mengukur kemiripan visual antara gambar asli dan gambar hasil modifikasi oleh *encoder*. Setelah *artificial fingerprint* disisipkan, idealnya gambar tidak mengalami perubahan visual yang signifikan, oleh karena itu MSE digunakan untuk mengkuantifikasi selisih piksel demi piksel antara dua citra tersebut. Nilai MSE yang rendah menunjukkan bahwa penyisipan jejak digital berlangsung secara tersembunyi (*invisible*) tanpa menyebabkan distorsi visual yang dapat terdeteksi oleh mata manusia. Evaluasi menggunakan MSE menjadi sangat

penting dalam konteks *invisible watermarking*, di mana kualitas gambar harus tetap terjaga agar tidak menimbulkan kecurigaan atau degradasi estetika.

Kombinasi kedua metrik ini, BCE untuk aspek akurasi ekstraksi *artificial fingerprint*, dan MSE untuk aspek preservasi kualitas citra, menjadi acuan dalam menilai efektivitas metode *artificial fingerprinting* yang diusulkan. Hanya jika kedua nilai *loss* berada pada tingkat yang rendah, maka metode dapat dianggap berhasil menyisipkan *artificial fingerprint* secara tersembunyi sekaligus tetap dapat dikenali secara akurat melalui proses *decoding*.

3.1.6 Deployment model

Model yang telah dilatih kemudian disimpan dalam format berkas .pth, yaitu format standar yang digunakan oleh pustaka PyTorch untuk menyimpan bobot dan parameter model. Model ini selanjutnya dideploy melalui sebuah layanan *Application Programming Interface* (API) yang dibangun menggunakan *framework* FastAPI. *Backend* sistem ini dirancang untuk memuat model *encoder, decoder*, serta model StyleGAN2 sebagai media uji. FastAPI menyediakan sejumlah *endpoint* yang memungkinkan pengguna untuk menyisipkan *artificial fingerprint* ke dalam citra, mengekstraksi *artificial fingerprint* dari citra, serta menghasilkan gambar wajah sintetik menggunakan StyleGAN2. Fungsi utama dari *backend* adalah menyediakan antarmuka bagi ketiga model tersebut agar dapat diakses dan digunakan secara efisien oleh sistem *frontend* melalui protokol HTTP.

Pada sisi *frontend*, antarmuka pengguna dikembangkan menggunakan ReactJS, dengan tujuan mempermudah proses pengoperasian sistem, bahkan bagi pengguna tanpa latar belakang teknis. Komunikasi antara *frontend* dan *backend* dilakukan menggunakan library Axios, yang memungkinkan pengiriman dan penerimaan data secara asinkron melalui *endpoint* yang tersedia. Untuk tampilan dan desain antarmuka, digunakan *framework* Tailwind CSS yang bersifat *utility first*, guna menghasilkan desain yang responsif, ringan, dan konsisten. *Frontend* mencakup fitur-fitur utama seperti unggah gambar, visualisasi proses *encode* dan *decode*, serta tampilan hasil

evaluasi model, termasuk nilai *Binary Cross Entropy* (BCE), *Mean Squared Error* (MSE), dan *Bitwise Accuracy*. Dengan arsitektur ini, sistem mampu menjembatani kompleksitas teknis metode *artificial fingerprinting* dengan kemudahan penggunaan, sehingga dapat diakses secara luas oleh berbagai kalangan, mulai dari peneliti hingga kreator konten digital.

3.2 Analisa Kebutuhan

Pada tahap ini penulis menganalisa apa saja kebutuhan selama pembangunan sistem, seperti *software, tools*, bahasa pemrograman, dan library yang digunakan. Berikut adalah apa saja yang digunakan dalam pembangunan sistem :

1. Bahasa Pemrograman
   a. Python

      Python merupakan bahasa pemrograman utama yang digunakan dalam pengembangan sistem ini. Python dipilih karena memiliki sintaks yang sederhana dan didukung oleh berbagai pustaka *Deep Learning* yang sangat powerful. Dengan fitur-fitur seperti kemudahan dalam pemrograman berorientasi objek, dukungan untuk pengolahan data, serta kemampuan untuk bekerja dengan data besar, Python adalah bahasa yang sangat ideal untuk pengembangan sistem berbasis *Deep Learning*.
2. Perangkat Lunak (*Software*)
   a. *Kaggle Notebook*

      Kaggle Notebook digunakan sebagai lingkungan pengembangan untuk menulis dan menjalankan kode secara interaktif. Platform ini memudahkan penulis untuk menguji dan memodifikasi kode, serta mendokumentasikan eksperimen.
   b. *Visual Studio Code*

      Visual Studio Code dipilih sebagai *text editor* pada pengembangan aplikasi dalam penelitian ini, *Visual Studio Code* dipilih dikarenakan mendukung banyak bahasa pemrograman dan

framework, *multi platform*, performa yang sangat cepat, mempunyai banyak *extensions* yang dapat mempermudah proses pengembangan website.

3. *Library* dan *framework*
   a. *Pytorch*

      Pytorch adalah *framework* yang digunakan untuk membangun, melatih, dan menerapkan model *Deep Learning*. *Pytorch* dipilih karena efisiensinya dalam menangani perhitungan matriks dan operasi *tensor*, serta kemampuan untuk melakukan pelatihan model menggunakan CPU/GPU.

   b. *Numpy*

      NumPy digunakan untuk manipulasi *array* dan matriks numerik. Pustaka ini sangat penting untuk pengolahan data numerik, yang sering digunakan dalam model *Deep Learning* untuk pemrosesan *batch* data.

   c. *Matplotlib*

      Matplotlib digunakan untuk visualisasi hasil eksperimen dan model. Visualisasi ini sangat penting untuk menganalisis proses pelatihan, seperti menggambar grafik loss dan akurasi selama pelatihan model, serta untuk memverifikasi performa model.

   d. TQDM

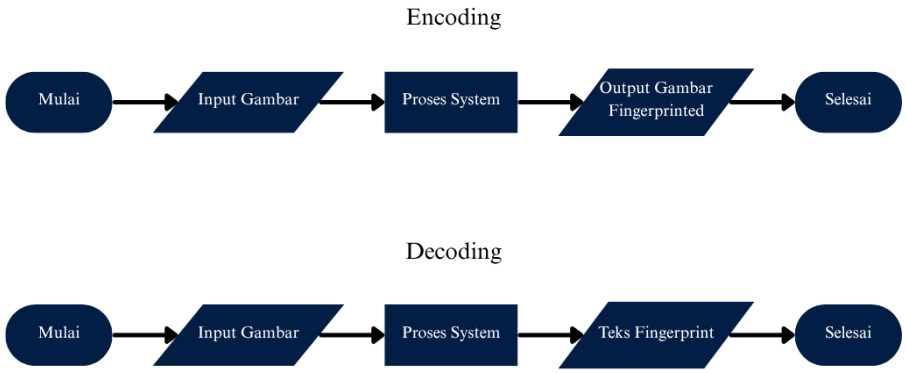
      TQDM digunakan untuk menampilkan progress bar pada iterasi pelatihan model. Ini memudahkan pengembang untuk memonitor kemajuan pelatihan, terutama ketika bekerja dengan *dataset* yang besar.

   e. *Tensorboard*

      Tensorboard Digunakan untuk visualisasi pelatihan model dalam hal grafik akurasi, *loss*, dan metrik lainnya. *TensorBoard* membantu dalam pemantauan pelatihan model secara *realtime*.

3.3 Penggunaan Sistem

Pada tahap ini dilakukan analisa untuk menentukan alur kerja penggunaan sistem penyisipan *artificial fingerprint* ke gambar dan ekstraksi *artificial fingerprint* dari gambar yang akan dilakukan oleh *user* dalam bentuk *flowchart*.



Gambar 3. 3 *Flowchart* alur kerja system

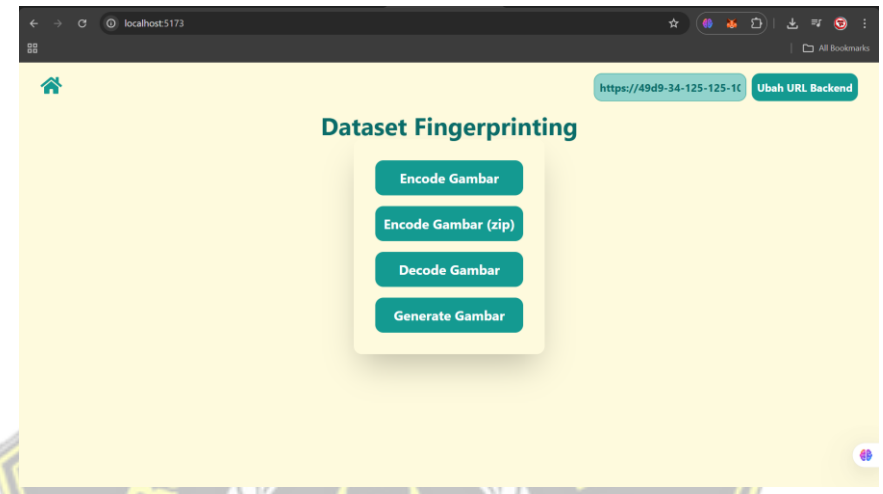
Gambar 3. 3 adalah *flowchart Encoding* dan *Decoding* yang menggambarkan proses sistem saat digunakan oleh *user*. Pada gambar 3. 1 memiliki beberapa tahapan, sebagai berikut :

1. *User* membuka aplikasi kemudian *user* akan melihat tampilan awal website attribusi gambar.
2. Pengguna memilih menu *encoding*, kemudian mengunggah file gambar, menentukan *seed* untuk *artificial fingerprint*, dan menekan tombol untuk memulai proses *encoding* gambar.
3. Kemudian sistem akan memproses gambar tersebut dengan cara menambahkan *tensor artificial fingerprint* dan mengekstraksi gambar tersebut.
4. *User* akan mendapatkan *output* dari sistem berupa hasil gambar yang sudah disisipi *tensor artificial fingerprint* serta *list artificial fingerprint* yang disispkan kedalam gambar.
5. Ketika *user* memilih menu *decoding*, maka *user* akan mengunggah file gambar lalu menekan tombol *encode* gambar untuk mulai proses *decoding* gambar.

6. *User* akan mendapatkan *output* dari sistem berupa hasil *artificial fingerprint* yang ada pada gambar.

3.4 Perancangan *User Interface*

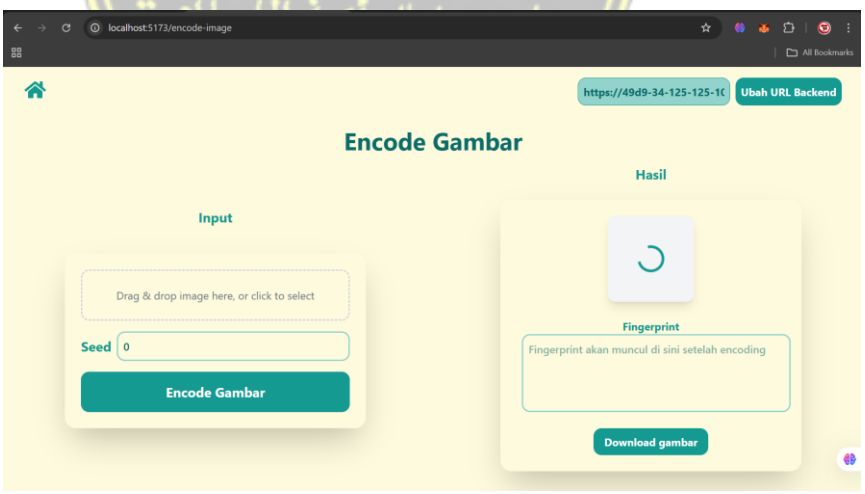
3.4.1 Halaman awal sistem



Gambar 3. 4 Tampilan halaman awal sistem

Gambar 3. 4 merupakan tampilan rancangan antar muka pada bagian halaman awal sistem. Pada halaman ini, pengguna bisa memilih untuk menyisipkan artificial fingerprint pada gambar (*encoding*) atau mengekstraksi artificial fingerprint dari gambar (*decoding*).

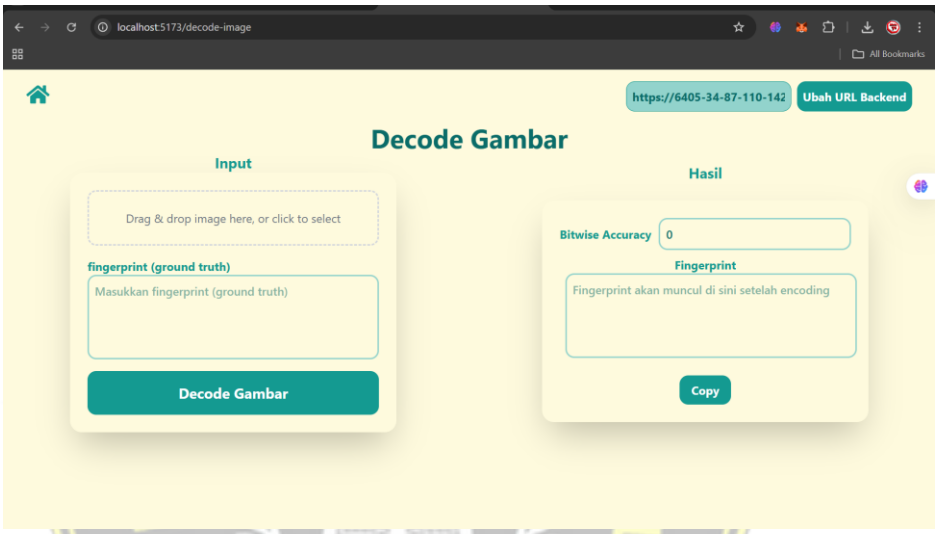
3.4.2 Halaman *Encode* gambar



Gambar 3. 5 Halaman *encode* gambar pada sistem

Gambar 3. 5 merupakan tampilan rancangan antar muka pada bagian halaman *encoding* gambar. Pada halaman ini, pengguna akan diminta untuk mengunggah gambar yang ingin di sisipi *artificial fingerprint*. Setelah gambar diunggah, sistem akan memproses gambar tersebut dan menghasilkan gambar yang sudah di sisipi *artificial fingerprint* beserta *artificial fingerprint* nya.

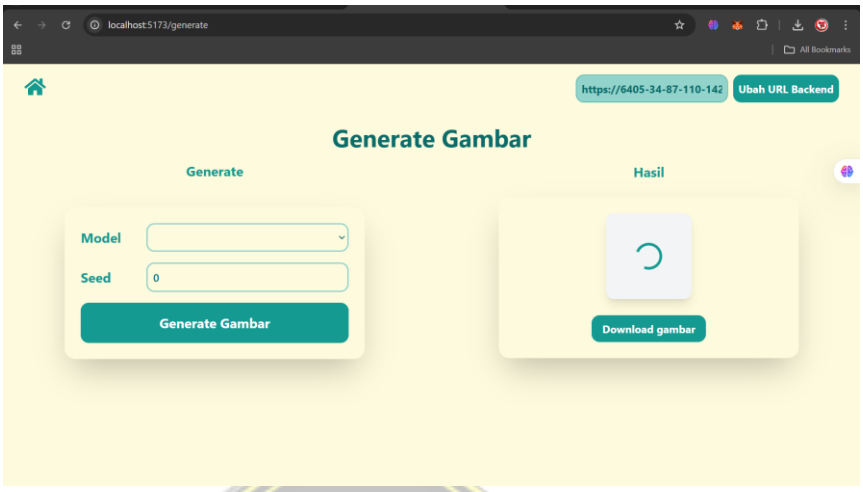
3.4.3 Halaman *Decode* gambar



Gambar 3. 6 Halaman *decode* gambar pada sistem

Gambar 3. 6 merupakan tampilan rancangan antar muka pada bagian halaman *decoding* gambar. Pada halaman ini, pengguna akan diminta untuk mengunggah gambar yang ingin di ekstrak untuk mendapatkan *artificial fingerprint* didalamnya dan juga *artificial fingerprint* yang ingin dicocokan sebagai *ground truth*. Setelah gambar diunggah, sistem akan memproses gambar tersebut dan menghasilkan *bitwise accuracy* dan *artificial fingerprint*.

3.4.4 Halaman Generate gambar

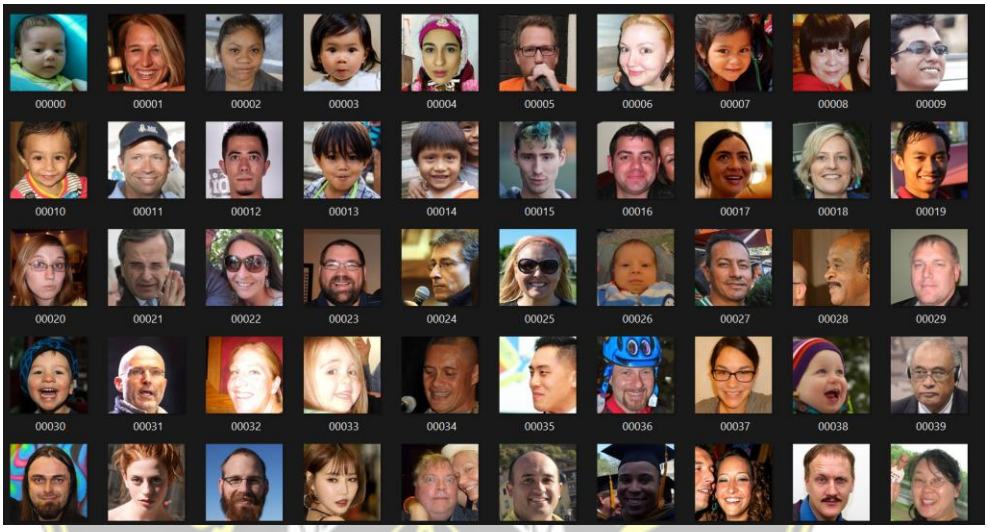


Gambar 3. 7 Halaman *generate* gambar

Gambar 3. 7 merupakan tampilan rancangan antar muka pada bagian halaman *generate* gambar. Pada halaman ini, pengguna akan diminta untuk memilih model styleGAN2 dan memasukkan *seed* yang akan di proses oleh sistem yang akan menghasilkan gambar sintetik.

BAB IV
HASIL DAN ANALISIS PENELITIAN

4.1 Hasil Pengumpulan Data



Gambar 4. 1 *Dataset* wajah FFHQ

Pada gambar 4. 1 merupakan kumpulan *dataset* gambar wajah FFHQ. Karena data adalah data yang disediakan dengan tujuan penelitian dan pengembangan model *generative* seperti GAN, maka data sudah sesuai dengan format yang diperlukan model, sehingga tidak perlu untuk melakukan *preprocessing* data terlebih dahulu.

4.2 Hasil Modeling

4.3.1 *Encoder*

Model pada *encoder* memiliki struktur yang menyerupai U-Net dengan beberapa modifikasi untuk menyesuaikan kebutuhan penyisipan *artificial fingerprint* pada gambar. Penyisipan fingerprint dilakukan dengan merubah inputan *artificial fingerprint* yang sebelumnya masih berupa *tensor* satu dimensi menjadi *tensor* 2 dimensi yang berukuran 16x16. *Artificial fingerprint* yang telah dikonversi menjadi *tensor* 2 dimensi di lakukan upsampling menyesuaikan dengan dimensi ukuran gambar *input* yaitu 128x128, kemudian proses dilanjutkan dengan menggabungkan kedua *tensor*

tersebut sebelum *tensor* hasil penggabungan kedua *tensor* tersebut masuk kedalam dua jalur utama model, yaitu:

1. *Encoder path* (*downsampling*)

   Encoder pada model ini berfungsi untuk mengekstrak fitur penting dari *tensor* gabungan antara gambar dengan *artificial fingerprint* melalui proses konvolusi yang bertahap.

2. *Decoder path* (*upsampling*)

   *Decode*r pada model ini berfungsi untuk mengembalikan *tensor* gabungan antara gambar dan *artificial fingerprint* yang sudah di konvolusi sebelumnya di jalur *encoder*.

Output dari *decode*r di hubungkan dengan *skip connection* dengan data *input* awal digunakan sebagai *ground truth* untuk menghitung kerugiannya, sehingga model mampu menggunakannya untuk belajar dan memperbaiki parameternya di *epoch* berikutnya. Rincian struktur arsitektur *encoder* ini ditunjukkan pada tabel 4.1.

Tabel 4. 1 Ringkasan model *Encoder*

No	Tipe Layer	input Shape	Output Shape
1	Input	(B, 2, 128, 128)	(B, 2, 128, 128)
2	Conv 1	(B, 2, 128, 128)	(B, 32, 128, 128)
3	Conv 2	(B, 32, 128, 128)	(B, 32, 64, 64)
4	Conv 3	(B, 32, 64, 64)	(B, 64, 32, 32)
5	Conv 4	(B, 64, 32, 32)	(B, 128, 16, 16)
6	Conv 5	(B, 128, 16, 16)	(B, 256, 8, 8)
7	Up6 + skip 4	(B, 256, 8, 8)	(B, 128, 16, 16)
8	Up7 + skip 3	(B, 128, 16, 16)	(B, 64, 32, 32)
9	Up8 + skip 2	(B, 64, 32, 32)	(B, 32, 64, 164)
10	Up9 + skip 1	(B, 32, 128, 128)	(B, 32, 128, 128)
11	Conv 10	(B, 32, 128, 128)	(B, 32, 128, 128)
12	Final output	(B, 32, 128, 128)	(B, 1, 128, 128)

4.3.2 *Decoder*

Model *decoder* memiliki arsitektur CNN sederhana yang terdiri dari beberapa lapisan konvolusi bertingkat dan diakhiri dengan dua lapisan *fully connected*. Tujuan utama dari *decoder* ini adalah untuk mendekode atau mengekstraksi *artificial fingerprint* yang telah disisipkan ke dalam gambar oleh *encoder* sebelumnya. Tidak seperti *encoder* yang memiliki struktur mirip U-Net dengan *skip connection*, arsitektur *decoder* hanya melakukan ekstraksi fitur bertingkat dari gambar hasil penyisipan menggunakan konvolusi dan aktivasi ReLU. *Output* dari jaringan konvolusional ini kemudian diratakan (*flattened*) dan diproses oleh dua lapisan *dense* untuk menghasilkan kembali *fingerprint* berdimensi satu, sesuai ukuran *fingerprint* yang disisipkan. Rincian struktur arsitektur *decoder* ini ditunjukkan pada tabel 4.2.

Tabel 4. 2 Ringkasan model *Decoder*

No	Tipe Layer	input Shape	Output Shape
1	Conv2d + ReLU	(1, 1, 128, 128)	(1, 32, 64, 64)
2	Conv2d + ReLU	(1, 32, 64, 64)	(1, 32, 64, 64)
3	Conv2d + ReLU	(1, 32, 64, 64)	(1, 64, 32, 32)
4	Conv2d + ReLU	(1, 64, 32, 32)	(1, 64, 32, 32)
5	Conv2d + ReLU	(1, 64, 32, 32)	(1, 64, 16, 16)
6	Conv2d + ReLU	(1, 64, 16, 16)	(1, 128, 8, 8)
7	Conv2d + ReLU	(1, 128, 8, 8)	(1, 128, 4, 4)
8	FC1 + ReLU	(1, 2048)	(1, 512)
9	FC2 (*Output*)	(1, 512)	(1, *fingerprint_size*)
10	Up9 + skip 1	(B, 32, 128, 128)	(B, 32, 128, 128)

4.3 Hasil Pelatihan Model

Setelah tahap pemodelan selesai, proses selanjutnya adalah melakukan pelatihan terhadap model *Encoder* dan *Decoder* dengan menggunakan *dataset* citra wajah beresolusi 128×128 piksel. Pelatihan dilakukan selama 10 *epoch* dengan dua fungsi kerugian (*loss functions*) yang berbeda, sesuai dengan dua tujuan utama sistem. Fungsi *Binary Crossentropy Error* digunakan untuk

mengukur tingkat kesalahan dalam ekstraksi *artificial fingerprint*, sedangkan *Mean Squared Error (MSE)* digunakan untuk mengevaluasi sejauh mana kualitas visual citra hasil penyisipan berubah dibandingkan dengan citra asli. Penggunaan kombinasi kedua fungsi ini bertujuan untuk memastikan bahwa fingerprint dapat ditanam dan diekstraksi secara akurat tanpa menyebabkan distorsi visual yang signifikan pada citra.

Selain menggunakan *Binary Crossentropy* untuk mengukur kesalahan prediksi pada *artificial fingerprint*, evaluasi performa sistem juga dilengkapi dengan penggunaan *bitwise accuracy*. Metrik ini menghitung persentase bit *artificial fingerprint* yang berhasil diprediksi dengan benar terhadap jumlah total bit. *Bitwise accuracy* memberikan gambaran yang lebih spesifik terhadap keberhasilan ekstraksi fingerprint pada tingkat biner, yang penting dalam konteks atribusi digital yang membutuhkan presisi bit level. Nilai akurasi ini dihitung berdasarkan jumlah bit yang identik antara *artificial fingerprint* asli dan hasil prediksi oleh *decoder*.

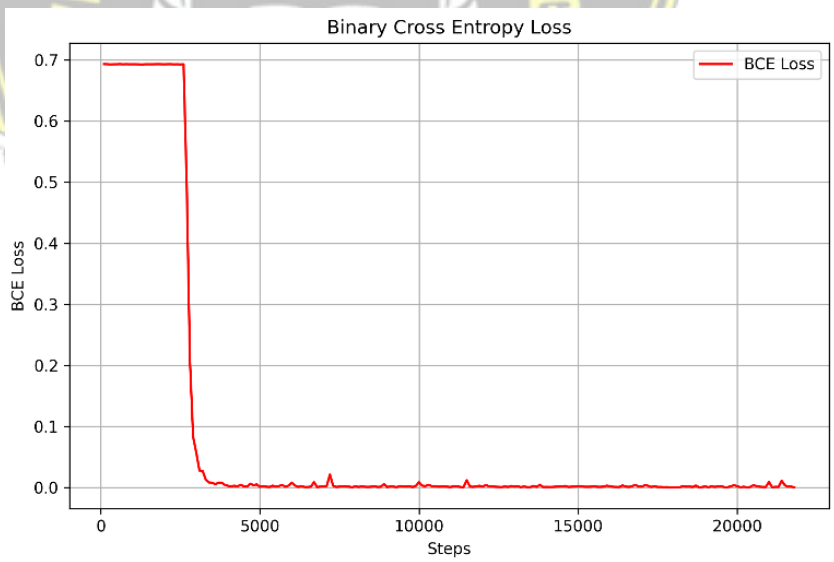
4.4 Hasil Evaluasi Model

Setelah tahap pemodelan arsitektur sistem selesai, proses selanjutnya adalah melakukan pelatihan terhadap model *encoder* dan *decoder* dengan menggunakan *dataset* wajah FFHQ beresolusi 128×128 piksel. *Dataset* ini dipilih karena menyediakan gambar wajah berkualitas tinggi dengan keragaman ekspresi, usia, dan pencahayaan, sehingga mampu merepresentasikan kondisi realistis dalam proses pelatihan. Pelatihan model dilakukan selama beberapa *epoch* dengan tujuan mengoptimalkan parameter pada *encoder* dan *decoder* agar sistem dapat secara efektif menyisipkan serta mengekstraksi *artificial fingerprint* secara tersembunyi.

Dalam proses pelatihan ini, digunakan dua fungsi kerugian (*loss functions*) utama, yaitu *Binary Cross Entropy* (BCE) *Loss* dan *Mean Squared Error* (MSE) *Loss*. Fungsi BCE digunakan untuk mengukur akurasi prediksi *artificial fingerprint* oleh *decoder*, yang direpresentasikan dalam bentuk bit *biner* (0 dan 1). BCE secara khusus mengukur perbedaan distribusi

probabilistik antara *artificial fingerprint* asli dan hasil prediksi, sehingga nilai loss yang rendah menunjukkan bahwa *artificial fingerprint* dapat dikenali kembali dengan tingkat presisi yang tinggi. Sementara itu, MSE digunakan untuk mengevaluasi sejauh mana kualitas visual gambar berubah setelah dilakukan proses penyisipan *artificial fingerprint*. Fungsi ini menghitung selisih kuadrat antar piksel antara gambar asli dan gambar hasil modifikasi, sehingga dapat menilai apakah terjadi distorsi visual yang signifikan.

Kombinasi dari kedua fungsi kerugian ini bertujuan untuk mencapai keseimbangan antara ketepatan deteksi *artificial fingerprint* dan preservasi kualitas citra visual. Dengan meminimalkan nilai BCE dan MSE secara bersamaan, diharapkan sistem mampu menyisipkan *artificial fingerrint* secara akurat dan tersembunyi (*invisible*) tanpa menurunkan kualitas estetika gambar asli secara nyata. Hal ini penting dalam konteks sistem atribusi digital, di mana keberhasilan deteksi *artificial fingerprint* dan keterjagaan citra visual merupakan dua komponen utama dari efektivitas metode yang diusulkan.



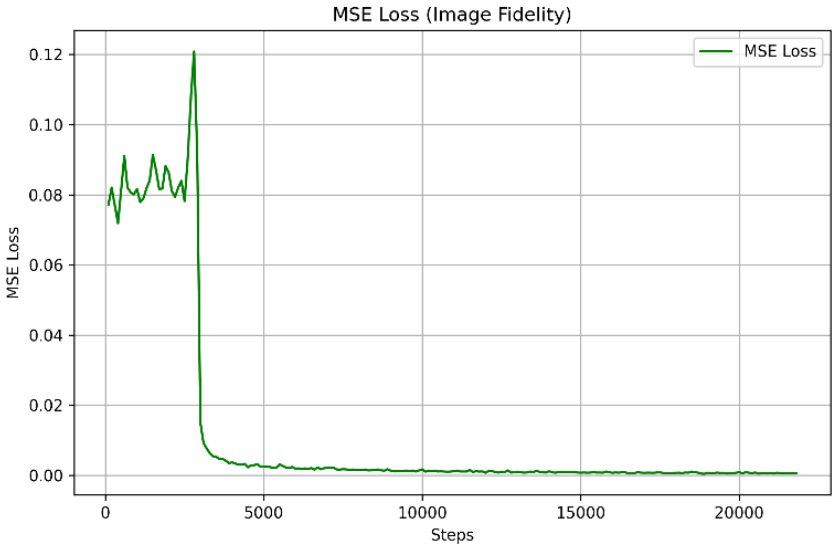
Gambar 4. 2 Grafik *BCE Loss*

Gambar 4.2 menyajikan grafik nilai *Binary Cross Entropy* (BCE) *Loss* selama proses pelatihan model, yang merepresentasikan tingkat kesalahan model dalam mempertahankan informasi *artificial fingerprint* yang disisipkan ke dalam citra. Pada fase awal pelatihan, nilai BCE berada di kisaran 0,7, menandakan bahwa model masih berada dalam tahap eksplorasi parameter

dan belum mampu mengenali pola *artificial fingerprint* dengan baik. Nilai loss yang relatif tinggi pada tahap ini merupakan hal yang wajar, karena parameter model masih acak dan belum tersesuaikan dengan distribusi data pelatihan.

Seiring dengan bertambahnya jumlah *epoch*, model secara bertahap mulai belajar menyesuaikan bobot-bobotnya untuk merepresentasikan hubungan antara gambar yang telah disisipkan *artificial fingerprint* dan *artificial fingerprint* aslinya. Penurunan signifikan pada nilai BCE mulai terlihat sekitar langkah ke-3000, yang menjadi indikasi bahwa model berhasil mengidentifikasi pola struktural dari *artificial fingerprint* buatan secara lebih konsisten. Penurunan ini menunjukkan bahwa *encoder* dan *decoder* mulai bekerja selaras dalam menyisipkan dan mengekstraksi *artificial fingerprint* dengan lebih presisi.

Stabilisasi nilai BCE pada level yang lebih rendah setelah titik tersebut menunjukkan bahwa model telah memasuki fase konvergensi, yaitu kondisi di mana pembaruan parameter menghasilkan peningkatan performa yang semakin kecil. Hal ini mengindikasikan bahwa model telah mencapai kondisi optimal dalam mempelajari distribusi fingerprint dan siap untuk digunakan dalam proses evaluasi lebih lanjut, termasuk pengujian ketahanan fingerprint terhadap transformasi yang dihasilkan oleh model generatif seperti StyleGAN2.



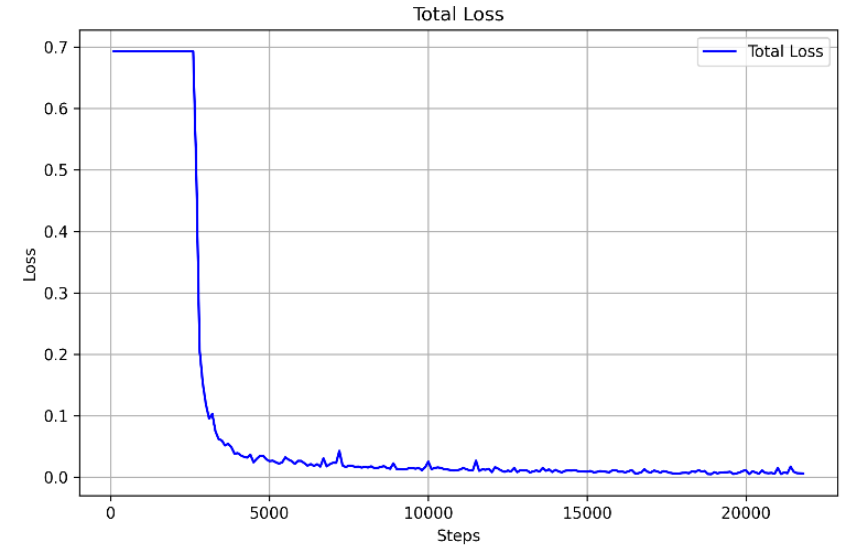
Gambar 4. 3 Grafik *MSE Loss*

Gambar 4.3 menyajikan grafik *Mean Squared Error* (MSE) *Loss* selama proses pelatihan model, yang digunakan untuk mengevaluasi kemampuan sistem dalam mempertahankan kualitas visual citra setelah proses penyisipan *artificial fingerprint*. Pada tahap awal pelatihan, nilai MSE berada di kisaran 0,07 hingga 0,09 dan menunjukkan fluktuasi yang cukup tinggi. Fluktuasi ini mencerminkan bahwa model masih berada dalam fase eksplorasi parameter, di mana bobot-bobot awal belum optimal dan penyisipan *artificial fingerprint* masih menghasilkan perubahan visual yang cukup signifikan terhadap citra asli.

Puncak nilai MSE terjadi pada sekitar langkah ke-2800, yang menandakan momen ketika modifikasi visual akibat proses penyisipan masih belum terkontrol sepenuhnya oleh encoder. Namun setelah melewati titik tersebut, grafik menunjukkan tren penurunan yang konsisten dan signifikan. Penurunan drastis pada nilai MSE ini menunjukkan bahwa model mulai berhasil belajar menanam *artificial fingerprint* dengan cara yang lebih halus dan tersembunyi, sehingga dampaknya terhadap citra menjadi semakin minimal.

Stabilisasi nilai MSE pada level yang sangat rendah mendekati nol mengindikasikan bahwa model telah berhasil meminimalkan distorsi visual dan menjaga integritas gambar asli. Hal ini penting dalam konteks *invisible*

watermarking, di mana tujuan utama adalah menyisipkan jejak digital tanpa menimbulkan perbedaan visual yang dapat dikenali oleh mata manusia. Dengan demikian, hasil ini memperkuat efektivitas model encoder dalam menyisipkan *artificial fingerprint* secara tersembunyi, tanpa mengorbankan aspek estetika maupun kualitas visual gambar.



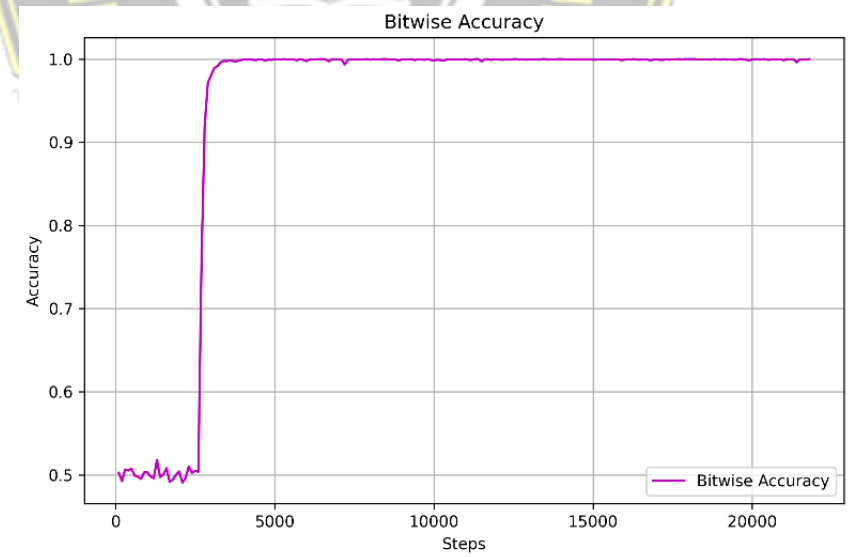
Gambar 4. 4 Total *Loss* (*BCE* + *MSE*)

Gambar 4.4 memperlihatkan gabungan total fungsi kerugian dari dua metrik utama yang digunakan dalam pelatihan model, yaitu *Binary Cross Entropy* (BCE) dan *Mean Squared Error* (MSE). Total *loss* ini merepresentasikan performa keseluruhan model dalam dua aspek penting: kemampuan menyisipkan serta mengenali *artificial fingerprint*, dan kemampuan menjaga kualitas visual citra yang telah dimodifikasi. Pada fase awal pelatihan, nilai total *loss* relatif tinggi, yaitu berada di kisaran 0,7. Nilai tersebut menunjukkan bahwa model masih belum mampu menjalankan kedua fungsi utamanya secara optimal, baik dari segi akurasi *artificial fingerprint* maupun dari sisi preservasi kualitas gambar. Kondisi ini merupakan hal yang umum pada tahap inisialisasi pelatihan, di mana bobot dan parameter masih berada dalam kondisi acak.

Memasuki sekitar langkah ke-3000, grafik menunjukkan penurunan yang sangat signifikan terhadap total loss. Penurunan ini menandakan bahwa model mulai menemukan kombinasi parameter yang mampu menyelaraskan

dua tujuan pelatihan yang berbeda namun saling bergantung mempertahankan struktur *artificial fingerprint* secara akurat, sekaligus memastikan perubahan visual yang dihasilkan tetap minimal. Dengan adanya sinergi antara fungsi BCE dan MSE dalam *loss* total, penurunan pada grafik ini menunjukkan bahwa proses pembelajaran yang terjadi pada *encoder* dan *decoder* berlangsung efektif dan berimbang.

Setelah fase penurunan tersebut, nilai total *loss* cenderung mengalami stabilisasi pada angka yang rendah, mendekati nol. Stabilitas ini mengindikasikan bahwa model telah mencapai fase konvergensi, di mana pembaruan parameter hanya menghasilkan peningkatan performa yang sangat kecil atau bahkan tidak signifikan. Dengan kata lain, model telah berhasil belajar untuk menyisipkan jejak digital secara tersembunyi tanpa mengganggu kualitas gambar, serta mampu mengenali kembali *artificial fingerprint* tersebut secara akurat dari citra hasil *encode*. Tren ini sekaligus memperkuat efektivitas keseluruhan arsitektur model dalam menjalankan tugas *artificial fingerprinting* secara komprehensif.



Gambar 4. 5 *Bitwise Accuracy*

Gambar 4.5 menyajikan grafik *bitwise accuracy,* yang digunakan sebagai metrik evaluasi untuk mengukur tingkat keberhasilan model dalam menyisipkan dan merekonstruksi *artificial fingerprint* dalam bentuk vektor biner secara akurat. Metrik ini menghitung persentase bit yang berhasil

diprediksi dengan benar oleh *decoder* dibandingkan dengan bit-bit asli yang ditanamkan oleh *encoder*. Pada tahap awal pelatihan, nilai *bitwise accuracy* berkisar antara 0,49 hingga 0,51, yang secara statistik hampir setara dengan hasil tebakan acak (*random guess*). Nilai ini menunjukkan bahwa model belum mampu mengenali pola *artificial fingerprint* secara efektif karena parameter masih belum terlatih dengan baik, dan proses pembelajaran belum membentuk representasi yang bermakna terhadap data input.

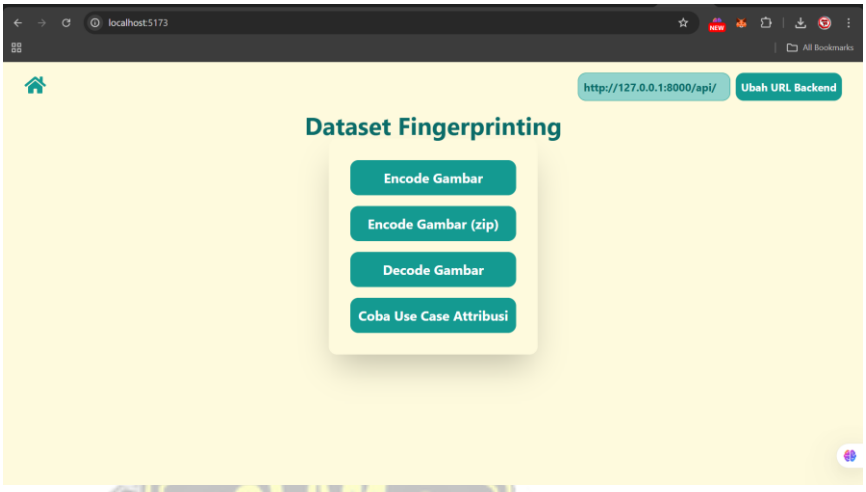
Namun, seiring berjalannya pelatihan, terutama setelah melewati sekitar langkah ke-3000, grafik menunjukkan lonjakan tajam pada nilai *bitwise accuracy*. Peningkatan ini mencerminkan bahwa model telah mulai mengidentifikasi dan mempelajari pola *artificial fingerprint* buatan dengan lebih akurat. Nilai akurasi mendekati 1,0 (atau 100%) menunjukkan bahwa hampir seluruh bit dalam *artificial fingerprint* berhasil diprediksi dengan benar oleh *decoder*, bahkan setelah *artificial fingerprint* tersebut melalui proses *encoding* dan penyisipan ke dalam gambar wajah.

Kenaikan *bitwise accuracy* ini memperkuat hasil evaluasi dari fungsi BCE *Loss* yang sebelumnya digunakan, karena menunjukkan secara eksplisit bahwa model tidak hanya mempelajari distribusi probabilistik, tetapi juga menghasilkan prediksi yang tepat pada tingkat granular bit. Hal ini sangat penting dalam konteks atribusi digital, di mana kesesuaian absolut pada representasi biner menjadi syarat utama untuk membuktikan kepemilikan atau pelanggaran terhadap hak cipta. Dengan demikian, nilai *bitwise accuracy* yang tinggi menjadi bukti kuat bahwa sistem *artificial fingerprinting* yang dikembangkan telah mencapai performa optimal dalam menyisipkan jejak digital tersembunyi secara akurat dan dapat dikenali kembali dengan presisi tinggi.

4.5 Hasil Implementasi

Setelah tahap pemodelan selesai, selanjutnya implementasi sistem *artificial fingerprinting* gambar ke dalam *platform* website dengan

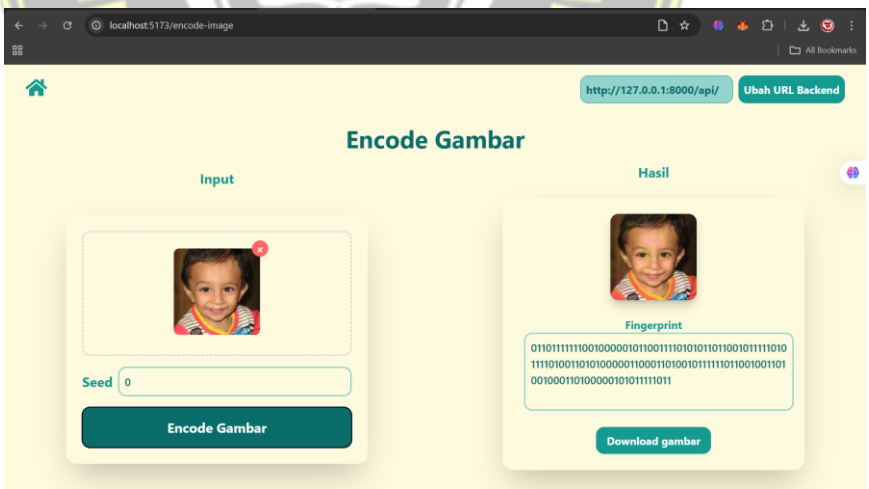
menggunakan *framework* FastAPI untuk backend dan ReactJS untuk frontend. Berikut ini merupakan hasil implementasi sistem :



Gambar 4. 6 Tampilah halaman awal sistem

Pada Gambar 4. 6 merupakan tampilan halaman utama yang akan ditampilkan pengguna saat pertama kali menjalankan sistem. Pada halaman ini terdapat 4 menu utama yang bisa dipilih oleh *user*, yaitu:

1. Menu *encode* gambar

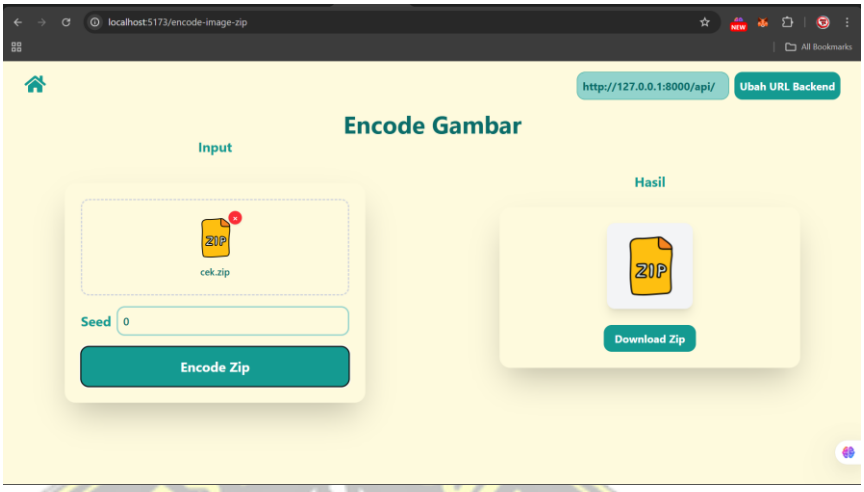


Gambar 4. 7 Tampilan halaman *encode* gambar

Pada gambar 4.7 menunjukkan tampilan pada halaman encode gambar. Dimana pada halaman ini user bisa menyisipkan gambarnya dengan sebuah fingerprint dengan cara mengunggah sebuah gambar lalu setelah proses *encoding* oleh sistem maka dihasilkan lah output berupa

gambar yang sudah disisipi yang bisa di download dan menampilkan informasi *artificial fingerprint* yang disisipkan.

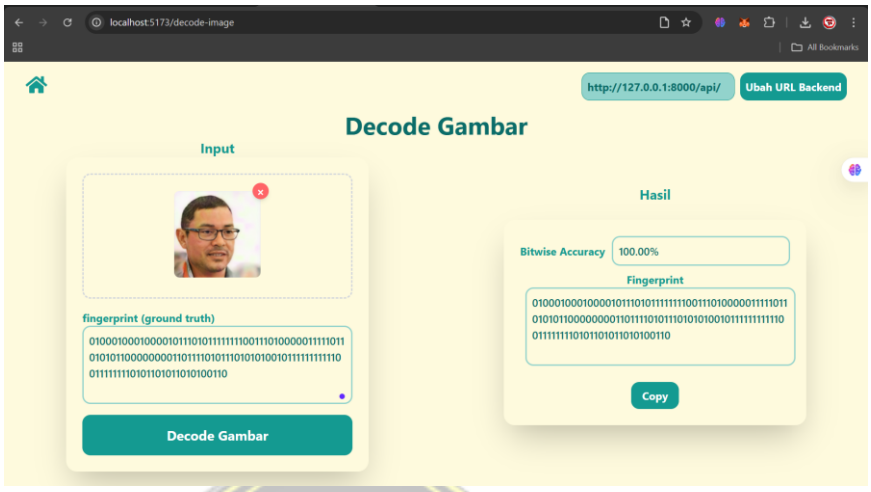
2. Menu *encode* gambar (zip)



Gambar 4. 8 Tampilan halaman *encode* (zip)

Gambar 4.8 menampilkan halaman *encode* (zip), yaitu antarmuka di mana pengguna dapat menyisipkan *artificial fingerprint* ke dalam gambar secara *batch*. Pada menu ini, pengguna diminta untuk mengunggah sejumlah gambar yang telah dikompresi dalam format ZIP melalui form yang tersedia. Setelah proses *encoding* selesai dilakukan oleh sistem, pengguna akan menerima kembali file ZIP yang berisi gambar-gambar yang telah disisipkan *artificial fingerprint*, beserta sebuah berkas berformat .json yang memuat informasi terkait *artificial fingerprint* yang ditanamkan pada gambar.

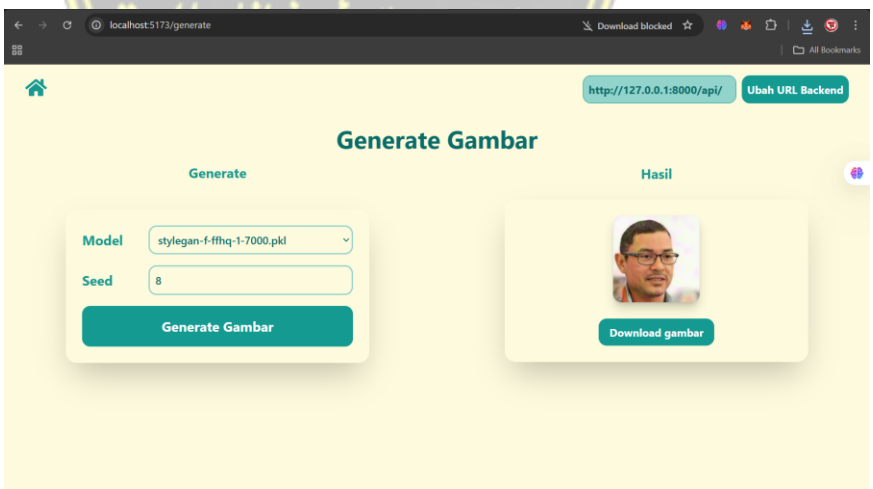
3. Menu *decode* gambar



Gambar 4. 9 Tampilan halaman *decode* gambar

Gambar 4.9 menunjukkan halaman *decode* gambar, yaitu antarmuka yang memungkinkan pengguna melakukan proses ekstraksi *artificial fingerprint* dari sebuah gambar. Pada menu ini, pengguna dapat mengunggah gambar yang diduga telah disisipkan *artificial fingerprint* melalui formulir yang tersedia. Setelah proses *decoding* dilakukan oleh sistem, hasil ekstraksi *artificial fingerprint* akan ditampilkan pada halaman, dan dapat disalin oleh pengguna untuk keperluan verifikasi atau pencocokan dengan *fingerprint* referensi.

4. Menu *generate* gambar



Gambar 4. 10 Tampilan halaman *generate* gambar

Pada menu ini, pengguna dapat menghasilkan gambar wajah sintetik menggunakan model generatif StyleGAN2 yang telah dilatih dengan *dataset* yang telah disisipi *artificial fingerprint*. Melalui gambar tersebut, pengguna dapat mengevaluasi apakah *artificial fingerprint* yang disisipkan tetap terbawa setelah proses pelatihan dan *generate* gambar oleh model. Pengguna juga dapat memilih varian model StyleGAN2 yang tersedia untuk melakukan proses *generate* gambar sintetik. Setelah gambar berhasil dihasilkan oleh sistem, pengguna dapat mengunduhnya dan melanjutkan proses dekripsi pada halaman *decode* guna memverifikasi keberadaan *fingerprint* yang telah disisipkan sebelumnya.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian ini, telah berhasil dikembangkan sebuah sistem *artificial fingerprinting* untuk atribusi gambar wajah sintetik yang dihasilkan oleh StyleGAN2, dengan menggunakan *convolutional neural network* (CNN) sebagai arsitektur utama pada bagian *encoder* dan *decoder*.

Model dilatih selama 10 *epoch* dan menunjukkan performa optimal setelah mengeksplorasi sekitar 3.000 gambar, dengan nilai *Binary Cross Entropy* (BCE) dan *Mean Squared Error* (MSE) *loss* sebesar 0,0071 dan *bitwise accuracy* sebesar 0,9996 pada *epoch* ke-10.

Hasil ini menunjukkan bahwa model mampu mengekstrasi *fingerprint* secara akurat dari gambar yang telah disisipkan. Sistem ini berpotensi menjadi solusi dalam mendukung perlindungan hak kekayaan intelektual para kreator visual, serta meningkatkan kesadaran terhadap penggunaan data secara etis dalam era kecerdasan buatan.

5.2 Saran

Penelitian selanjutnya disarankan untuk menguji sistem *artificial fingerprinting* pada model generatif lain seperti StyleGAN3, DALL-E, atau *Stable Diffusion* guna mengevaluasi kemampuan generalisasi sistem. Pengujian terhadap ketahanan model terhadap manipulasi gambar seperti kompresi, pemotongan, dan penambahan *noise,* juga penting untuk meningkatkan *robustness*.

Pengembangan lanjutan dapat mencakup integrasi mekanisme verifikasi digital, penggunaan *dataset* yang lebih besar dan beragam, eksplorasi arsitektur model yang lebih kompleks seperti *transformer*, serta peningkatan antarmuka pengguna agar lebih interaktif dan terintegrasi dengan layanan pelaporan otomatis.

DAFTAR PUSTAKA

Alwanda, M. R., Ramadhan, R. P. K., & Alamsyah, D. (2020). Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle. *Jurnal Algoritme*, *1*(1), 45–56. https://doi.org/10.35957/algoritme.v1i1.434

de Meira, N. F. C., Silva, M. C., Bianchi, A. G. C., & Oliveira, R. A. R. (2023). Generating Synthetic Faces for Data Augmentation with StyleGAN2-ADA. *International Conference on Enterprise Information Systems, ICEIS - Proceedings*, *1*(Iceis), 649–655. https://doi.org/10.5220/0011994600003467

Franceschelli, G., & Musolesi, M. (2022). Copyright in generative deep learning. *Data and Policy*, *4*(3), 1–18. https://doi.org/10.1017/dap.2022.10

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). *Generative Adversarial Nets*. 3063–3071. https://doi.org/10.1109/ICCVW.2019.00369

Handoko, D., Nizamiyati, Andi, S., Aghata, F., Wulandari, Fahrullah, Yunita, F., Puspasari, I., Atho'illah, I., Asnur, P., Rahmah, sabrina aulia, Jaya, I., Siregar, amril mutoi, Oktarino, A., Rizal, A., & Farizy, S. (2024). *Artificial Intelligent Revolusi Kecerdasan Buatan*.

Hermawati, F. A., & Jaya, V. A. (2024). *Komputika : Jurnal Sistem Komputer Segmentasi Kepala Janin pada Citra Ultrasound Menggunakan Arsitektur Jaringan U-Net Fetal Head Segmentation in Ultrasound Images Using U-Net Network Architecture*. *13*, 193–199. https://doi.org/10.5281/zenodo.1322001

Issn, I. P. E.-, Setiawan, R. F., Zuhdi, M. R., & Harjo, B. I. (2024). *IDENTIFIKASI KESEGARAN DAGING AYAM MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK S*. *02*, 7–15.

Karras, T., Laine, S., & Aila, T. (2019). A Style-Based Generator Architecture for Generative Adversarial Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(12), 4217–4228. https://doi.org/10.1109/TPAMI.2020.2970919

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020).

Analyzing and improving the image quality of stylegan. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 8107–8116. https://doi.org/10.1109/CVPR42600.2020.00813

Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition To cite this version : HAL Id : hal-03926082 Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791

Lu, Y., Shen, M., Wang, H., Wang, X., van Rechem, C., Fu, T., & Wei, W. (2023). *Machine Learning for Synthetic Data Generation: A Review*. *14*(8), 1–18. http://arxiv.org/abs/2302.04062

Madhusudana, P. C., Birkbeck, N., Wang, Y., Adsumilli, B., & Bovik, A. C. (2022). Image Quality Assessment using Synthetic Images. *Proceedings - 2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops, WACVW 2022*, 93–102. https://doi.org/10.1109/WACVW54805.2022.00015

Man, K., & Chahl, J. (2022). A Review of Synthetic Image Data and Its Use in Computer Vision. *Journal of Imaging*, *8*(11). https://doi.org/10.3390/jimaging8110310

Marra, F., Gragnaniello, D., Verdoliva, L., & Poggi, G. (2019). Do GANs Leave Artificial Fingerprints? *Proceedings - 2nd International Conference on Multimedia Information Processing and Retrieval, MIPR 2019*, 506–511. https://doi.org/10.1109/MIPR.2019.00103

Peryanto, A., Yudhana, A., & Umar, R. (2020). Rancang Bangun Klasifikasi Citra Dengan Teknologi Deep Learning Berbasis Metode Convolutional Neural Network. *Format : Jurnal Ilmiah Teknik Informatika*, *8*(2), 138. https://doi.org/10.22441/format.2019.v8.i2.007

Pratama, A. R., Wabula, F., Ilmandry, H., Isabela, M. L., & Sianipar, R. (2025). *Literature Review The Impact of Machine Learning in Modern Industries*. *2021*.

Sha, Z., Li, Z., Yu, N., & Zhang, Y. (2023). DE-FAKE: Detection and Attribution

of Fake Images Generated by Text-to-Image Generation Models. *CCS 2023 - Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 3418–3432. https://doi.org/10.1145/3576915.3616588

Talib, D. A., & Abed, A. A. (2023). Real-Time Deepfake Image Generation Based on Stylegan2-ADA. *Revue d'Intelligence Artificielle*, *37*(2), 397–405. https://doi.org/10.18280/ria.370216

Terven, J., Cordova-Esparza, D. M., Ramirez-Pedraza, A., Chavez-Urbiola, E. A., & Romero-Gonzalez, J. A. (2025). *Loss Functions and Metrics in Deep Learning*. http://arxiv.org/abs/2307.02694

Yu, N., Davis, L., & Fritz, M. (2019). Attributing fake images to GANs: Learning and analyzing GAN fingerprints. *Proceedings of the IEEE International Conference on Computer Vision*, *2019-Octob*, 7555–7565. https://doi.org/10.1109/ICCV.2019.00765

Yu, N., Skripniuk, V., Abdelnabi, S., & Fritz, M. (2021). Artificial Fingerprinting for Generative Models: Rooting Deepfake Attribution in Training Data. *Proceedings of the IEEE International Conference on Computer Vision*, 14428–14437. https://doi.org/10.1109/ICCV48922.2021.01418

Zhao, X., Liu, H., Fan, W., Liu, H., Tang, J., & Wang, C. (2021). AutoLoss: Automated Loss Function Search in Recommendations. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 3959–3967. https://doi.org/10.1145/3447548.3467208