SISTEM DETEKSI PEMALSUAN CITRA MENGGUNAKAN CNN DENSENET-121 DAN *WATERMARK LEAST SIGNIFICANT BIT* (LSB) UNTUK VALIDASI CITRA PALSU

LAPORAN TUGAS AKHIR

Laporan ini Disusun Guna Memenuhi Salah Satu Syarat Memperoleh Gelar Sarjana Strata (S1) pada Program Studi Teknik Informatika

Fakultas Teknologi Industri

Universitas Islam Sultan Agung Semarang

Disusun Oleh :

Nama	: Fatwa Akbar Jiwani
NIM	: 32602100045
Program Studi	: Teknik Informatika

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM SULTAN AGUNG

SEMARANG

2025

DEEPFAKE IMAGE DETECTION SYSTEM USING CNN DENSENET-121 LSB WATERMARKING FOR VALIDATION DEEPFAKE

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 :

Name : Fatwa Akbar Jiwani
NIM : 32602100045

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

LEMBAR PENGESAHAN
TUGAS AKHIR

SISTEM DETEKSI PEMALSUAN CITRA DIGITAL MENGGUNAKAN CNN DENSENET-121 DAN *WATERMARK* LSB UNTUK VALIDASI CITRA PALSU

FATWA AKBAR JIWANI
NIM 32602100045

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

TIM PENGUJI UJIAN SARJANA:

Dedy Kurniadi ST,M.Kom NIK. 210615048 (Ketua Penguji)	:	.....
Badie'ah, M.Kom NIK. 210615044 (Anggota Penguji)	:	10 - 3 - 2025 .....
Bagus SWP,S.Kom,M.Cs NIK. 210616051 (Pembimbing)	:	10 - 3 - 2025 .....

Semarang, ...
Mengetahui,
Kaprodi Teknik Informatika
Universitas Islam Sultan Agung

Moch. Taufik, ST., MIT
NIDN.0622037502

SURAT PERYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Fatwa Akbar Jiwani

Nim : 32602100045

Judul Tugas Akhir : **SISTEM DETEKSI PEMALSUAN CITRA DIGITAL MENGGUNAKAN CNN DENSENET-121 DAN *WATERMARK LEAST SIGNIFICANT BIT* (LSB) UNTUK VALIDASI CITRA PALSU**

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

Semarang

Yang Menyatakan

Fatwa Akbar Jiwani

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Fatwa Akbar Jiwani

NIM : 32602100045

Program Studi : Teknik Informatika

Fakultas : Teknologi Industri

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul :

SISTEM DETEKSI PEMALSUAN CITRA DIGITAL MENGGUNAKAN CNN DENSENET-121 DAN *WATERMARK LEAST SIGNIFICANT BIT* (LSB) UNTUK VALIDASI CITRA PALSU

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

Semarang

Yang Menyatakan

Fatwa Akbar Jiwani

KATA PENGANTAR

Dengan mengucap rasa syukur alhamdulillah atas kehadirat Allah SWT yang telah memberikan rahmat dan karunianya kepada penulis, sehingga dapat menyelesaikan Tugas Akhir dengan judul "Sistem Deteksi Pemalsuan Citra Menggunakan Cnn Densenet-121 Dan Watermark Lsb Untuk Validasi Citra Palsu" ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana (S-1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang. Tugas Akhir ini disusun dan dibuat dengan adanya bantuan dari berbagai pihak, materi maupun teknis, oleh karena itu saya selaku penulis mengucapkan terima kasih kepada :

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

Dengan segala kerendahan hati, penulis menyadari bahwa laporan tugas akhir ini masih memiliki banyak kekurangan dalam hal kualitas, kuantitas, dan ilmu pengetahuan. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun untuk membantu laporan ini menjadi lebih baik di masa mendatang.

Semarang, ……

Fatwa Akbar Jiwani

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Deteksi *deepfake* menjadi tantangan besar dalam era digital karena dampaknya terhadap keamanan informasi. Penelitian ini mengusulkan sistem deteksi *deepfake* menggunakan CNN DenseNet-121 untuk mengklasifikasikan gambar sebagai asli atau palsu, dengan LSB *Watermarking* sebagai validasi keaslian citra. Dataset terdiri dari gambar asli dan *deepfake* yang diproses dengan augmentasi untuk meningkatkan generalisasi model. Model dilatih dengan optimasi Adam dan binary cross-entropy, lalu dievaluasi berdasarkan akurasi, precision, recall, dan F1-score. Hasilnya, kombinasi DenseNet-121 dan LSB *Watermarking* mencapai akurasi 91%, dengan precision 0.92, recall 0.91, dan F1-score 0.91. Pendekatan ini tidak hanya mendeteksi *deepfake* dengan presisi tinggi, tetapi juga menyematkan bukti forensik digital yang dapat diverifikasi.

Kata kunci : *Deepfake*, DenseNet-121, CNN, LSB *Watermarking*, Deteksi Pemalsuan Citra

ABSTRACK

Deepfake detection has become a major challenge in the digital era due to its impact on information security. This study proposes a deepfake detection system using CNN DenseNet-121 to classify images as real or fake, with LSB Watermarking for image authenticity validation. The dataset consists of real and deepfake images processed with augmentation to enhance model generalization. The model is trained using the Adam optimizer and binary cross-entropy, then evaluated based on accuracy, precision, recall, and F1-score. The results show that the combination of DenseNet-121 and LSB Watermarking achieves 91% accuracy, with a precision of 0.92, recall of 0.91, and F1-score of 0.91. This approach not only detects deepfake images with high precision but also embeds verifiable digital forensic evidence.

Keyword : Deepfake, DenseNet-121, CNN, LSB Watermarking, Deteksi Pemalsuan Citra

BAB I
PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi dalam pengolahan citra dan kecerdasan buatan telah membawa dampak signifikan di berbagai bidang seperti kesehatan, keamanan, hiburan, dan forensik digital. Inovasi ini memberikan banyak manfaat, seperti peningkatan kualitas gambar medis untuk diagnosis, sistem pengenalan wajah untuk keamanan, serta efek visual realistis dalam industri hiburan. Namun, di balik manfaat tersebut, muncul tantangan serius terkait manipulasi dan pemalsuan citra digital yang dapat digunakan untuk tujuan yang tidak etis.

Salah satu bentuk manipulasi citra yang semakin mendapat perhatian adalah *deepfake*, yaitu teknik yang menggunakan kecerdasan buatan untuk mengganti atau mengubah wajah, ekspresi, atau adegan dalam gambar dan video dengan tingkat ketelitian yang tinggi. Teknologi ini semakin canggih dan sulit dibedakan dari citra asli, sehingga menimbulkan berbagai ancaman, seperti penyebaran informasi palsu, pencemaran nama baik, hingga ancaman terhadap keamanan dan privasi individu. Penyalahgunaan *deepfake* dalam konteks politik, sosial, dan ekonomi juga dapat menimbulkan dampak yang luas bagi masyarakat.

Seiring dengan pesatnya perkembangan teknik pemalsuan citra, kebutuhan akan metode deteksi yang efektif menjadi semakin mendesak. Salah satu pendekatan yang menjanjikan dalam mendeteksi pemalsuan citra adalah pembelajaran mendalam (*Deep Learning*), khususnya dengan model *Convolutional Neural Network* (CNN). CNN memiliki kemampuan dalam mengenali pola kompleks dan mendeteksi perbedaan halus antara citra asli dan yang telah dimanipulasi. Dengan mengekstraksi fitur-fitur dari citra, model ini dapat mengidentifikasi tanda-tanda manipulasi yang sulit dideteksi oleh mata manusia.

Selain pembelajaran mendalam, teknik *watermarking* berbasis kriptografi dapat menjadi solusi tambahan dalam menjaga keaslian citra. *Watermarking* memungkinkan penanaman informasi tersembunyi dalam citra yang dapat digunakan untuk verifikasi keasliannya. Dengan adanya watermark yang sulit

dihapus atau dimanipulasi, setiap perubahan atau pemalsuan citra dapat terdeteksi dengan lebih mudah. Metode ini dapat memberikan perlindungan terhadap citra sejak awal distribusi dan memastikan integritasnya dalam jangka panjang.

Berdasarkan permasalahan di atas, penelitian ini dilakukan untuk mengembangkan sistem deteksi pemalsuan citra *deepfake* yang menggabungkan metode pembelajaran mendalam dengan teknik *watermarking*. Pendekatan ini diharapkan dapat memberikan solusi yang lebih efektif dalam mendeteksi dan mencegah penyalahgunaan *deepfake*, sekaligus memastikan integritas citra digital. Dengan adanya sistem ini, diharapkan dapat tercipta lingkungan digital yang lebih aman dan terpercaya dalam menghadapi ancaman pemalsuan citra yang semakin berkembang.

1.2 Perumusan Masalah

Berdasarkan latar belakang permasalahan, maka dapat diindetifikasikan permasalahan yang akan dibahas sebagai berikut :

1. Bagaimana penerapan metode pembelajaran mendalam, khususnya *Convolutional Neural Network* (CNN) dengan arsitektur DenseNet 121, dapat digunakan untuk deteksi manipulasi pada citra digital ?
2. Bagaimana *watermarking* dengan metode *Least Significant Bit* (LSB) dapat diterapkan untuk memberikan validasi dalam verifikasi keaslian citra tanpa mengurangi kualitas dari cira itu sendiri ?

1.3 Pembatasan Masalah

Adapun pembatasan masalah dari penulisan proposal ini yaitu sebagai berikut:

1. Format Citra yang Diteliti

   Penelitian ini hanya mencakup citra digital dengan format tertentu seperti JPEG atau PNG, dan tidak mencakup jenis media lain seperti video, audio, atau teks.

2. Pendekatan Pembelajaran Mendalam

   Penelitian ini terbatas pada penggunaan *Convolutional Neural Network* (CNN) sebagai algoritma pembelajaran mendalam untuk deteksi pemalsuan

citra, sementara algoritma lain seperti RNN atau GAN tidak termasuk dalam cakupan penelitian.

3. Teknik Steganografi

   Penerapan steganografi dalam penelitian ini dibatasi pada penggunaan *watermarking* digital berbasis algoritma *Least significant bit* (LSB) untuk menyisipkan informasi pada citra digital. Metode ini digunakan untuk memastikan watermark dapat disematkan dengan perubahan minimal pada kualitas visual citra

4. Jenis Manipulasi Citra

   Penelitian ini berfokus pada deteksi manipulasi citra khususnya seperti *deepfake* image, dan tidak mencakup manipulasi yang lebih kompleks atau perubahan pada video.

1.4 Tujuan

Adapun tujuan tugas akhir dari penulisan proposal ini yaitu sebagai berikut :

1. Mengembangkan model deeplearning berbasis *Convolutional Neural Network* (CNN) menggunakan arsitektur DenseNet121 untuk mendeteksi pemalsuan citra (*deepfake* image).
2. Menerapkan teknik *watermarking* sebagai metode untuk menyisipkan informasi integritas citra tanpa mengurangi kualitas citra yang berguna untuk validasi sehingga bisa membuat system lebih cepat dalam mendeteksi .
3. Menggabungkan metode *deeplearning* dan *watermarking* untuk menciptakan sistem deteksi pemalsuan citra yang lebih aman dan efektif.

1.5 Manfaat

Adapun manfaat yang diharapkan dari penyusunan tugas akhir ini adalah sebagai berikut:

1. Kontribusi terhadap Pengembangan Teknologi Deteksi *Deepfake*

   Laporan ini diharapkan dapat memberikan kontribusi dalam pengembangan teknologi deteksi pemalsuan citra dengan memanfaatkan model *Deep Learning* berbasis *Convolutional Neural Network* (CNN) menggunakan arsitektur DenseNet121. Hal ini dapat membantu meningkatkan akurasi dalam mengidentifikasi citra yang telah dimanipulasi.

2. Validasi Hasil Deteksi Melalui Teknik *Watermarking*

   Dengan menerapkan teknik *watermarking* sebagai validasi hasil deteksi, sistem akan secara otomatis menyisipkan teks pada citra sesuai dengan hasil deteksi. Jika gambar terdeteksi sebagai palsu, akan ditambahkan watermark berupa teks *"Gambar Palsu"*. Sebaliknya, jika gambar terdeteksi asli, akan disisipkan watermark *"Gambar Asli"*. Teknik ini membantu memastikan integritas citra sekaligus memberikan informasi visual yang jelas terkait keaslian gambar.

3. Integrasi Metode Deteksi dan Perlindungan Citra yang Lebih Efektif

   Penggabungan metode *Deep Learning* dengan teknik *watermarking* berbasis hasil deteksi diharapkan mampu menciptakan sistem deteksi pemalsuan citra yang lebih aman, efisien, dan efektif. Sistem ini dapat menjadi solusi inovatif yang mampu memberikan perlindungan sekaligus validasi langsung pada citra yang terdeteksi.

4. Sumbangan Ilmiah bagi Penelitian Selanjutnya

   Laporan ini dapat menjadi referensi akademis bagi peneliti atau mahasiswa yang tertarik mengembangkan teknologi deteksi citra lebih lanjut. Hasil penelitian ini diharapkan dapat membuka peluang untuk eksplorasi lanjutan dalam bidang kecerdasan buatan, keamanan data, dan steganografi digital.

1.6 Sistematika Penulisan

Sistematika penulisan yang akan digunakan oleh penulis dalam sebuah pembuatan laporan tugas akhir adalah sebagai berikut :

BAB 1 : **PENDAHULUAN**

Pada bab ini penulisan pengutamakan latar belakang pemilihan judul, rumusan masalah, Batasan masalah, tujuan penelitian, metode penelitian, serta sistematika penulisan

BAB 2 : **TINJAUAN PUSTAKA DAN DASAR TEORI**

Bab ini memuat penelitian-penenlitian sebelumnya dan dasar teori untuk membantu penulisan memahami bagaimana teori yang berhubungan dengan metode algoritma *Convolutional Neural*

Network dan proses *Watermarking* untuk penelitian ini.

BAB 3 : **METODE PENELITIAN**

Bab ini mengungkapkan proses tahapan-tahapan penelitian dimulai dari mendapatkan data hingga proses perancangan model dan implementasi

BAB 4 : **HASIL DAN ANALISA PENELITIAN**

Pada bab ini penulisan mengungkapkan hasil penelitian yaitu penggunaan CNN dengan algoritma DenseNet-121 sebagi model untuk mendeteksi pemalsuan gambar dan *watermarking* LSB untuk validasi keaslian gambar.

BAB 5 : **KESIMPULAN DAN SARAN**

Bab ini penulisan memaparkan kesimpulan proses penelitian dari awal hingga akhir.

BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Pemalsuan citra digital, termasuk teknologi *deepfake*, telah berkembang pesat seiring kemajuan kecerdasan buatan (AI) dan pembelajaran mendalam. *Deepfake* memungkinkan manipulasi citra dan video dengan tingkat ketelitian yang sangat tinggi, sehingga sering kali sulit dideteksi secara visual oleh manusia. *Deepfake* dapat mengubah wajah, ekspresi, atau keseluruhan adegan dalam citra dan video, yang berdampak pada privasi, keamanan informasi, dan kepercayaan publik terhadap media digital (Almars, 2021)·(Karnouskos, 2020), Seiring dengan semakin canggihnya teknik manipulasi ini, muncul kebutuhan mendesak untuk mengembangkan metode pendeteksian yang lebih andal, akurat, dan efisien dalam membedakan citra asli dari citra hasil manipulasi.

Menurut penelitian yang dilakukan oleh Tolosana (Tolosana dkk., 2020) dan Karnouskos (Karnouskos, 2020), sudah banyak metode yang dikembangkan untuk mendeteksi *deepfake*, baik yang berbasis analisis piksel maupun teknik pembelajaran mesin. Meskipun demikian, tantangan utama dalam pendeteksian *deepfake* adalah kemampuan teknologi ini yang terus meningkat dalam meniru pola wajah, pencahayaan, dan tekstur secara realistis (Masood dkk., 2023).

Salah satu metode deeplearning yang cocok dan efektif untuk mengenali pola dalam citra adalah CNN yang bekerja dengan mengekstraksi fitur spasial dari citra, sehingga mampu membedakan citra asli dari citra yang telah dimanipulasi. Penelitian yang dilakukan oleh (License & Rizvee, 2023) menggunakan delapan arsitektur CNN untuk mendeteksi gambar *deepfake* dari dataset berukuran besar, dengan hasil yang terbukti andal dan akurat. Dari penelitian yang dilakukan oleh (Patel dkk., 2023) menunjukkan bahwa CNN sangat efektif dalam mendeteksi gambar *deepfake*.

DenseNet-121 adalah arsitektur jaringan saraf konvolusional yang menghubungkan setiap lapisan dengan seluruh lapisan yang lebih dalam, contohnya, lapisan pertama terhubung langsung dengan lapisan kedua, ketiga, keempat, dan seterusnya. Pendekatan ini bertujuan untuk mengoptimalkan aliran data di antara lapisan-lapisan dalam jaringan (Satrio dkk., n.d.). Studi yang dilakukan oleh Alzahrani, Ahmed (Alzahrani, 2024) melaporkan bahwa model DenseNet121 mencapai akurasi hingga 92.32% dalam mengidentifikasi *deepfake*. Yang mana akan dilakukan evaluasi dengan menggunakan metrik seperti akurasi, presisi, recall, F1-score, dan area under the ROC curve (AUC),

Namun, salah satu tantangan utama dalam sistem ini adalah bagaimana mengamankan hasil deteksi model agar informasi mengenai keaslian suatu citra dapat dipertahankan dan diverifikasi oleh pihak lain. Meskipun CNN telah terbukti efektif dalam mendeteksi *deepfake*, hasil deteksi ini masih dapat dimanipulasi atau dipalsukan kembali tanpa mekanisme perlindungan tambahan. Selain itu, metode konvensional dalam pendeteksian *deepfake* umumnya hanya menghasilkan output dalam bentuk nilai prediksi atau label oleh karna itu di samping pendeteksian *deepfake*, *watermarking* steganografi adalah metode yang digunakan untuk melindungi keaslian citra digital dengan menambahkan tanda air yang dapat diverifikasi. Teknik ini menggunakan algoritma steganografi untuk menyisipkan tanda air atau hash ke dalam citra digital, yang kemudian dapat diverifikasi untuk memastikan integritas citra (Sanivarapu dkk., 2022),(Swaminathan dkk., 2006). Menurut penelitian (Memon & Wong, 1998), *watermarking* memberikan perlindungan tambahan pada citra, yang membantu mengidentifikasi setiap perubahan atau manipulasi yang terjadi pada citra sejak pertama kali dibuat. sedangkan studi oleh Boato dkk. (Boato dkk., 2009) mengevaluasi kekuatan dan ketahanan *watermarking* kriptografi terhadap berbagai bentuk manipulasi.

Dengan kemajuan teknologi dan kemudahan akses ke media digital, isu manipulasi serta pemalsuan gambar digital semakin mengemuka (A C & M T, 2023). Metode *watermarking* berfungsi melindungi keaslian gambar dengan menyisipkan informasi yang dapat mengidentifikasi sumber atau pemilik gambar, sehingga memungkinkan deteksi terhadap perubahan atau penggunaan yang tidak

sah (Begum & Uddin, 2020). Bose dan Sabramanyam (Bose dkk., 2014),(Subramanyam dkk., 2012) telah mengeksplorasi dampak kompresi terhadap efektivitas *watermarking*. salah satu metode *watermarking* adalah menggunakan metode LSB (*Least significant bit*), metode LSB mampu menghasilkan kualitas citra yang baik dengan rata-rata PSNR mencapai 65 dB, serta memungkinkan penyisipan watermark secara tak terlihat tanpa mengubah citra asli secara signifikan (Chopra, 2012). Studi lebih lanjut oleh Parthasarathy. (Parthasarathy & Nagar, 2009),(Wan dkk., n.d.) menyoroti potensi peningkatan ketahanan teknik LSB terhadap gangguan eksternal seperti kompresi dan manipulasi, sehingga dapat meningkatkan keandalannya untuk aplikasi keamanan data digital. Perkembangan teknologi ini mengindikasikan bahwa solusi integratif antara pendeteksian *deepfake* dan *watermarking* digital merupakan arah masa depan yang menjanjikan, terutama dalam menghadapi tantangan manipulasi media yang semakin kompleks.

2.2 Dasar Teori

2.2.1 Deeplearning

Deep Learning, atau pembelajaran mendalam, adalah cabang dari Machine Learning yang bertujuan untuk menyelesaikan berbagai permasalahan kompleks dalam pengolahan data digital, termasuk deteksi pemalsuan citra atau *deepfake*. Pembelajaran mendalam memiliki keunggulan dalam mengekstraksi fitur dari data mentah secara otomatis, sehingga menjadi alat yang sangat efisien dalam mendeteksi perubahan detail yang umumnya sulit dikenali oleh mata manusia atau teknik pengolahan gambar tradisional (Lecun dkk., 2023).

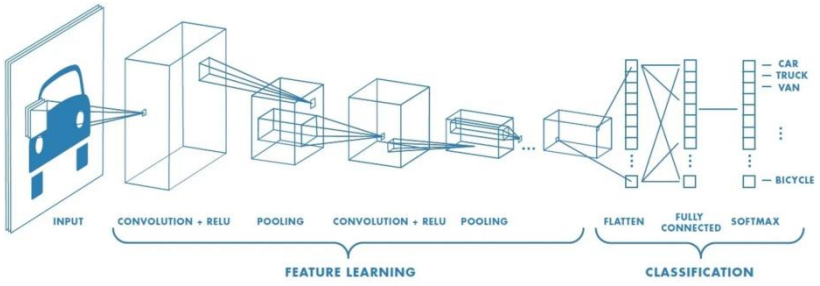
Berdasarkan jenis pembelajarannya, *Deep Learning* dibagi menjadi dua kategori utama: *Supervised Learning*, seperti klasifikasi gambar yang terpantau; dan Unsupervised Learning, seperti analisis pola tanpa panduan label. Dalam konteks pendeteksian *deepfake*, pembelajaran terstruktur ini berguna untuk melatih model agar dapat membedakan antara gambar asli dan gambar yang dimanipulasi dengan akurasi yang tinggi (Sharma dkk., 2020).

Salah satu metode *Deep Learning* yang relevan untuk pendeteksian citra adalah *Convolutional Neural Network* (CNN), yang terkenal dalam pemrosesan citra karena kemampuannya menangkap pola visual dalam data gambar. Selain itu,

beberapa arsitektur lain seperti *Recurrent Neural Network* (RNN) dan *Long Short-Term Memory* (LSTM) juga digunakan dalam mengolah data yang memerlukan analisis berurutan, yang kadang diperlukan dalam pendeteksian video *deepfake* (Shrestha & Mahmood, 2019)

2.2.2 Convolutional Neural Network (CNN)

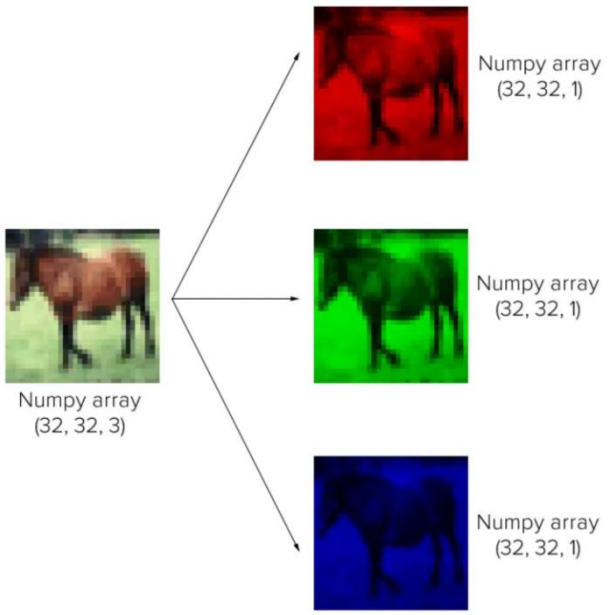
Convolutional Neural Network (CNN) merupakan sebuah arsitektur jaringan saraf tiruan yang dibuat khusus untuk memproses dan menganalisis data dalam bentuk grid, seperti citra digital. *Convolutional Neural Network* (CNN) memiliki keunggulan dalam mendeteksi pola dan fitur penting dari data melalui lapisan konvolusi, sehingga memungkinkan model mempelajari struktur dan detail visual dengan lebih efisien.(Satrio et al., n.d.). Dalam konteks sistem deteksi pemalsuan citra (*deepfake*) yang memanfaatkan *Deep Learning* dengan perlindungan *watermarking* kriptografi, *Convolutional Neural Network* (CNN) memiliki peran yang sangat penting. CNN adalah jenis jaringan saraf tiruan yang dioptimalkan untuk pemrosesan data dalam bentuk grid atau gambar. Dalam aplikasi deteksi *deepfake*, CNN memungkinkan untuk mengidentifikasi pola dan fitur dalam gambar yang mungkin tidak terlihat oleh mata manusia, seperti tekstur, ketajaman, dan perubahan warna halus yang sering kali menjadi tanda adanya manipulasi digital (Praveen Chakravarthy et al., 2022).



Gambar 2. 1. rancangan arsitektur CNN (Hussain et al., 2019)

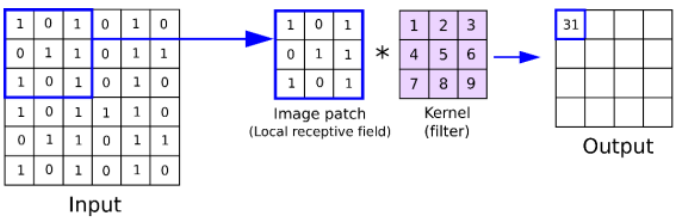
Arsitektur CNN dirancang dengan beberapa lapisan utama, *convolutional layer* (terdiri dari *pooling layer*, activation function dan hyperparameter) dan fully conncected layer.

a) *Convolutional* Layer (Conv. Layer)



Gambar 2. 2. RGB (Red, Green, Blue) image berukuran 32x32

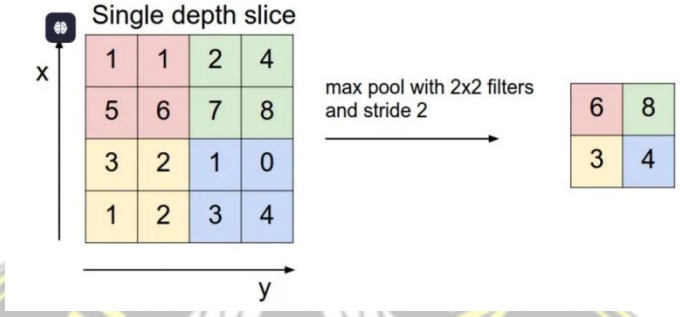
B Gambar diatas adalah RGB (Red, Green, Blue) image berukuran 32x32 pixels yang sebenarnya adalah multidimensional array dengan ukuran 32x32x3 (3 adalah jumlah channel). *Convolutional* layer terdiri dari filter 5x5x3 yang digeser ke seluruh gambar. Setiap pergeseran melakukan operasi "dot product" antara filter dan input, menghasilkan activation map atau feature map.



Gambar 2. 3 Ilustrasi Feature Map

b) *Pooling Layer*

Pooling layer umumnya terletak setelah convolutional layer. Pooling layer menggunakan filter dengan ukuran dan stride tertentu yang bergerak di seluruh area feature map. Dua jenis pooling yang sering digunakan adalah Max Pooling dan Average Pooling. Misalnya, dengan *Max Pooling* 2x2 dan stride 2, filter akan memilih nilai maksimum dari area 2x2 pixel yang dilaluinya, sementara Average Pooling akan memilih nilai rata-rata dari area tersebut.



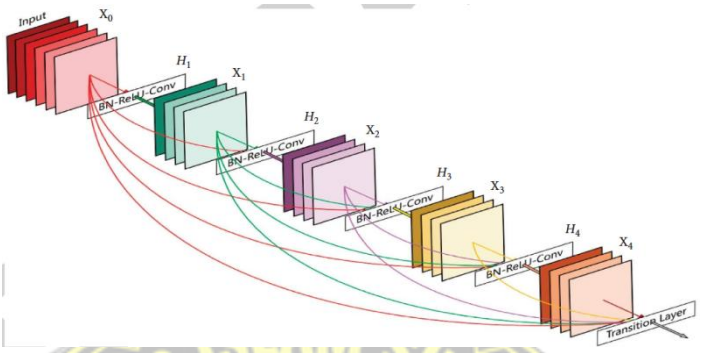
Gambar 2. 4 Contoh Operasi MaxPooling

Tujuan dari penggunaan pooling layer adalah mengurangi dimensi dari feature map (downsampling), sehingga mempercepat komputasi karena parameter yang harus diupdate semakin sedikit dan mengatasi overfitting.

2.2.3 DenseNet

Dense Convolutional Network (DenseNet) adalah arsitektur jaringan saraf konvolusional yang menghubungkan setiap lapisan atau blok secara langsung dengan semua lapisan atau blok sebelumnya melalui koneksi umpan maju. Berbeda dengan jaringan konvolusional tradisional yang memiliki koneksi antara lapisan yang berurutan, DenseNet memiliki koneksi langsung antara setiap lapisan, memungkinkan informasi dari lapisan sebelumnya digunakan sebagai input untuk lapisan berikutnya. Keunggulan utama DenseNet meliputi kemampuan mengatasi masalah gradien yang menghilang, memperkuat penyebaran fitur, mendorong penggunaan ulang fitur, dan secara signifikan mengurangi jumlah parameter. Pada tahun 2017, model DenseNet menerima penghargaan sebagai makalah terbaik di konferensi CVPR (Computer Vision and Pattern Recognition) (Huang, Liu, Van Der Maaten, & Weinberger, 2017).

DenseNet menggunakan tiga blok utama yang masing-masing terdiri dari lapisan-lapisan dengan batch normalization, fungsi aktivasi ReLU, dan konvolusi dengan filter 3×3. Lapisan yang menghubungkan dua blok disebut lapisan transisi, yang berfungsi untuk mengubah ukuran fitur melalui proses konvolusi dan pooling. Desain ini dapat dilihat pada Gambar dibawah (License & Rizvee, 2023)



Gambar 2. 5 architektur DenseNet

(License & Rizvee, 2023)

DenseNet dengan arsitektur yang terdiri dari empat blok menggunakan kombinasi lapisan batch normalization, aktivasi ReLU, dan konvolusi dengan filter 3×3 di dalam setiap blok padatnya (dense block). Antara dua blok yang berdekatan, terdapat transition layer yang berfungsi untuk mengubah ukuran fitur menggunakan konvolusi dan average pooling. Pada tahap klasifikasi, DenseNet menerapkan global average pooling diikuti oleh aktivasi softmax untuk menghasilkan output akhir.

Proses CNN dalam DenseNet-121 memiliki beberapa tahap sebelum menghasilkan output berupa hasil akhir, Tahapan pertama dimulai dari Convolutional Layer yang bertujuan untuk Ekstraksi fitur lokal dari gambar yang dilakukan dengan cara enggeser kernel/filter W pada gambar input X dan menghasilkan fitur baru Y. persamaannya dapat dilihat pada persamaan dibawah :

$$Y[i,j] = \sum_{m=1}^{k} \sum_{n=1}^{k} W[m,n] * \mathrm{X}[i+m, j+n] \quad (1)$$

$Y[i,j]$ = Nilai piksel hasil konvolusi di posisi (i,j)

$\mathrm{X}[i+m, j+n]$ = Nilai piksel dari gambar asli (input) di sekitar posisi (i,j)

$W[m,n]$ = Nilai bobot dari filter atau kernel konvolusi berukuran $k * k$

k = Ukuran kernel atau filter (misalnya $3*3$, $5*5$)

$\sum =$ Menunjukkan operasi penjumlahan elemen-elemen hasil perkalian antara filter dan citra

Setelah proses Convolutional Layer selesai maka proses selanjutnya adalah Activation Function yang bertujuan untuk Meningkatkan non-linearitas, sehingga model dapat mempelajari hubungan kompleks dengan menggunakan fungsi umum ReLU (Rectified Linear Unit) (Albelwi, 2022).

$$f(x) = max(0, x) \quad (2)$$

$f(x)$ = Nilai keluaran setelah fungsi aktivasi diterapkan.

x = Nilai masukan (hasil operasi sebelumnya, misalnya dari lapisan konvolusi).

Tahap selanjutnya pada bagian Dense Block Setiap layer akan menerima input dari semua layer sebelumnya, menghasilkan aliran fitur yang padat dimana Output dari layer lll dalam Dense Block (Albelwi, 2022).

$$x_l = H_l([x0, x1, \ldots, xl - 1]) \quad (3)$$

Tahap selanjutnya dibagian Transition Layer, Mengontrol ukuran data dengan down-sampling menggunakan pooling dan convolution. Yang mana dilakukan Batch Normalization untuk stabilisasi

$$\hat{x} = \frac{x-\mu}{\sigma} \quad (4)$$

x = Nilai asli (data mentah).

μ = Rata-rata (mean) dari seluruh data.

σ/sigma = Standar deviasi dari data.

$\hat{x}$ = Nilai yang telah dinormalisasi .

Tahap selanjutnya adalah menentukan Softmax, Softmax digunakan untuk menghasilkan probabilitas dari setiap kelas dengan Menggabungkan semua fitur dan membuat prediksi akhir seperti persamaan dibawah

$$y = Softmax\ (Wx + b) \quad (5)$$

y = Output setelah diterapkan fungsi Softmax (probabilitas untuk setiap kelas).

W= Bobot (weight) yang dikalikan dengan input.

x = Input (misalnya fitur yang diekstrak dari CNN).

b = Bias (nilai tambahan untuk menggeser output).

$Softmax$ = Fungsi aktivasi yang mengubah nilai menjadi probabilitas antara 0 dan 1, serta totalnya menjadi 1 (untuk klasifikasi multi-kelas).

Dan langkah terahir adalah menghitung Loss Function menggunakan metode Cross-Entropy Loss (untuk klasifikasi) yang mana fungsinya adalah untuk Mengukur kesalahan prediksi untuk mengoptimalkan model untuk persamaan nya sendiri seperti dibawah

$$L = -\sum_i y_i log(\hat{y}_i) \quad (6)$$

L = Nilai loss (kerugian), yaitu seberapa jauh prediksi dari target sebenarnya.

y_i = Nilai target asli (label sebenarnya), biasanya bernilai 1 untuk kelas yang benar dan 0 untuk kelas lainnya (one-hot encoding).

$\hat{y}_i$ = Probabilitas prediksi untuk kelas iii setelah Softmax (output model).

$log(\hat{y}_i))$ = Logaritma dari probabilitas prediksi untuk mengukur seberapa jauh dari target sebenarnya.

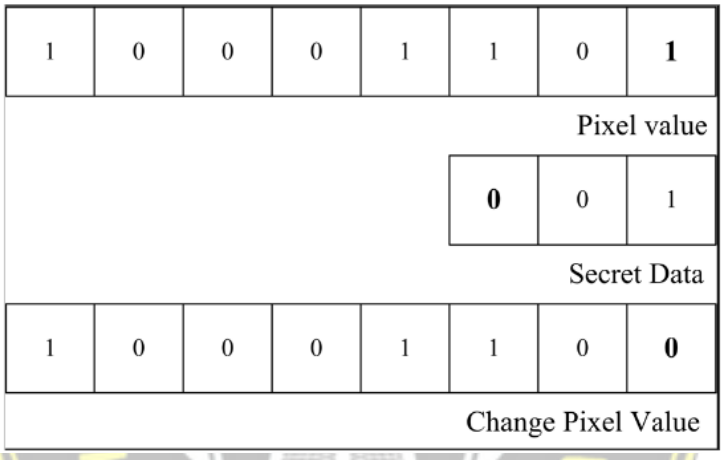
$\sum_i$ = Penjumlahan dilakukan untuk semua kelas

2.2.4 Least significant bit (LSB)

Metode *Least significant bit* (LSB) adalah salah satu teknik *watermarking* pada domain spasial yang paling sederhana dan populer. Teknik ini bekerja dengan memanfaatkan bit paling tidak signifikan (*least significant bit*) dari piksel citra digital. Dalam metode ini, watermark disisipkan dengan mengganti LSB piksel citra host dengan data watermark (Faheem dkk., 2022).

Dalam konteks representasi biner dari sebuah data, bit yang paling signifikan (Most Significant Bit atau MSB) adalah bit yang terletak paling kiri dalam suatu angka biner dan memiliki nilai paling besar. Sebaliknya, bit yang paling tidak signifikan (*Least significant bit* atau LSB) adalah bit yang terletak paling kanan dan memiliki kontribusi nilai terkecil terhadap nilai keseluruhan data.

Sebagai contoh, jika sebuah byte (8 bit) memiliki nilai biner 10110011, bit pertama (1) adalah MSB dan bit terakhir (1) adalah LSB.



Gambar 2. 6 contoh dari 1 bit LSB

BAB III

METODE PENELITIAN

3.1 Metode Penelitian

Penelitian ini menggunakan CNN dengan arsitektur DenseNet-121 untuk membuat model yang bisa digunakan untuk mendteksi apakah gambar *deepfake* atau tidak serta dengan *watermarking* LSB sebagai validasi. sehingga system tidak hanya memberikan hasil klasifikasi namun tanda digital yang disematkan kedalam gambar.

3.1.1 Pengumpulan Data

Dataset yang digunakan dalam penelitian ini terdiri dari dua jenis citra, yaitu citra asli (*original images*) yang tidak mengalami modifikasi atau manipulasi, serta citra *deepfake* (*deepfake images*) yang telah dimanipulasi menggunakan algoritma *deepfake* dengan tanda-tanda modifikasi yang halus dan sulit dikenali. Dataset ini diperoleh dari sumber terbuka (*opensource*) Kaggle, yang mencakup face forensic images dan *deepfake* images. Seluruh dataset dibagi menjadi tiga subset, yaitu: Data pelatihan (*training* data), Data validasi (*validation* data) dan, Data pengujian (*testing* data).

Pembagian ini dilakukan untuk memastikan proses pengembangan model yang terstruktur dan komprehensif. Dataset disimpan dalam struktur folder terpisah dan diunggah ke Google Drive guna memudahkan integrasi dengan Google Colab dalam proses pelatihan serta evaluasi model. Pembagian dataset dilakukan dengan proporsi yang seimbang untuk memastikan akurasi dan generalisasi model.

Tabel 3. 1 Pembagian Dataset

No.	Dataset	Deepake Image	Real Image
1.	Dataset Pelatihan	8000	8000
2.	Dataset validasi	2000	2000
3.	Dataset Pengujian	1191	1191
	Total	**11191**	**11191**

3.1.2 *Preprocessing* Dataset

Tahapan ini mencakup beberapa langkah penting dalam pengolahan citra sebelum digunakan untuk pelatihan model:

A. ***Resize* :** Menyesuaikan ukuran gambar menjadi 128x128 piksel agar lebih optimal dalam proses pelatihan dan mengurangi beban komputasi.

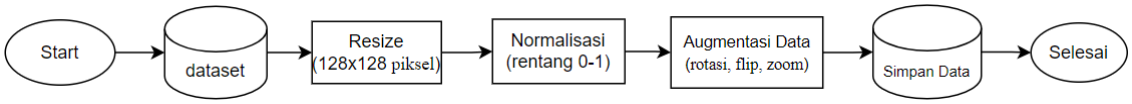
B. **Konversi ke *Grayscale*** : Setiap gambar dikonversi menjadi skala abu-abu (grayscale) untuk mengurangi dimensi fitur dan meningkatkan efisiensi pemrosesan.

C. **Normalisasi :** Nilai piksel dipetakan ke dalam rentang 0-1 dengan membagi setiap piksel dengan 255 (rescale=1./255) guna mempercepat konvergensi model.

D. **Augmentasi Data (Data Augmentation)** : Dilakukan hanya pada data pelatihan untuk menambah variasi data secara virtual melalui transformasi berikut:

1) Rotasi hingga 20°
2) Perubahan ukuran lebar & tinggi ±20%
3) Shear transformation hingga 20%
4) Zoom in/out hingga 20%
5) Flipping horizontal
6) Variasi kecerahan (0.8 – 1.2)
7) Perubahan intensitas channel hingga 10 unit

Gambar yang telah diproses disimpan dalam bentuk array NumPy, kemudian diberikan dimensi tambahan untuk menyesuaikan format input model (1 channel untuk grayscale). Dataset dibagi menjadi data pelatihan, validasi, dan pengujian/testing, dengan proses normalisasi yang sama untuk memastikan distribusi data tetap konsisten.



Gambar 3. 1 Flowchart Preprocessing data

3.1.3 Pelatihan Model

A. Model CNN DenseNet-121

Penelitian ini menggunakan model *Convolutional Neural Network* (CNN) dengan arsitektur DenseNet-121 untuk mengklasifikasikan gambar sebagai asli atau telah dimanipulasi (*deepfake*). Dense Convolutional Network (DenseNet) merupakan arsitektur jaringan saraf konvolusional yang menghubungkan setiap lapisan (layer) secara langsung dengan semua lapisan sebelumnya melalui koneksi umpan maju (feed-forward).

Tidak seperti CNN tradisional yang hanya menghubungkan lapisan secara berurutan, DenseNet memanfaatkan fitur dari semua lapisan sebelumnya sebagai input. Hal ini memungkinkan model untuk mengurangi masalah *vanishing gradient*, mempercepat proses pelatihan, dan meningkatkan efisiensi parameter karena adanya koneksi langsung antar lapisan.

Struktur DenseNet-121 terdiri dari beberapa blok utama, masing-masingnya :

1. Batch Normalization: Menormalkan output dari layer sebelumnya untuk mempercepat pelatihan.
2. Fungsi Aktivasi ReLU: Memperkenalkan non-linearitas pada jaringan. Lapisan-lapisan ini dihubungkan oleh transition layers, yang berfungsi untuk mengurangi ukuran fitur map melalui pooling dan konvolusi, sehingga mengurangi kompleksitas komputasi.

Proses Pelatihan:

1. Dataset yang telah diproses sebelumnya digunakan untuk melatih model.
2. Model dikompilasi dengan:
    a) Optimizer: Adam, yang dipilih karena kemampuannya menyesuaikan learning rate secara dinamis.
    b) Loss Function: Binary Crossentropy, yang cocok untuk tugas klasifikasi biner.
    c) Metrik Acuracy: untuk melihat accuracy dari model saat ditrainning.
3. Parameter Pelatihan yang digunakan:

d) Epoch: 10 (jumlah iterasi pelatihan model)
e) Batch Size: 32 (jumlah sampel yang diproses dalam satu kali iterasi)
f) Learning Rate: 0.001 (tingkat pembelajaran awal)

Setelah pelatihan, model dievaluasi menggunakan **data validasi** untuk mengukur performa dan menghindari *overfitting*. Proses ini memastikan model dapat menggeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya.

B. *Watermarking Least significant bit* (LSB)

Teknik *Least significant bit* (LSB) adalah salah satu metode *watermarking* sederhana namun efektif yang bekerja di domain spasial. Teknik ini memanfaatkan bit terkecil (*least significant bit*) dari piksel citra digital untuk menyisipkan tanda air (watermark).

Proses Penyisipan Watermark:

1. Setiap piksel citra direpresentasikan dalam format biner.
2. Bit terakhir dari representasi biner piksel diubah sesuai dengan data watermark. Misalnya, jika citra memiliki representasi biner `10110011`, bit terakhir (`1`) dapat diubah menjadi `0` atau `1` sesuai watermark yang ingin disisipkan.

Penerapan dalam Sistem Deteksi *Deepfake*:

1. Setelah model melakukan klasifikasi, watermark akan disisipkan ke dalam gambar sesuai hasil deteksi.
   a) Jika gambar terdeteksi sebagai **palsu**, watermark berupa teks *"Gambar Palsu"* akan disisipkan.
   b) Jika gambar terdeteksi sebagai **asli**, watermark berupa teks *"Gambar Asli"* akan disisipkan.
2. Watermark ini bersifat imperceptible (tidak terlihat oleh mata manusia) namun tetap dapat diekstraksi untuk memverifikasi keaslian gambar.

Keunggulan LSB:

1. Tidak memengaruhi kualitas visual gambar secara signifikan.
2. Proses penyisipan dan ekstraksi yang cepat dan efisien.

Dengan integrasi antara model deteksi berbasis DenseNet-121 dan teknik *watermarking* LSB, sistem ini dapat memberikan validasi otomatis terhadap gambar digital, meningkatkan keamanan, dan menjaga integritas data.

3.1.4 Evaluasi Model CNN

Mengukur kinerja model dalam mendeteksi citra *deepfake* pada dataset yang belum pernah dilihat sebelumnya (*validation dan testing set*). Validation set digunakan selama pelatihan untuk memantau performa model. Testing set digunakan untuk mengevaluasi model setelah pelatihan selesai dengan prosedur model dilatih menggunakan dataset *training*. Dataset *validation* digunakan untuk memilih hyperparameter terbaik, seperti *learning rate* dan jumlah epoch. Dataset *testing* digunakan untuk mengevaluasi performa akhir model.

a) Metrik Evaluasi

Model dievaluasi menggunakan beberapa matrik evaluasi yakni *Accuracy* (Akurasi), *precision* (presisi), *recall* dan *F1 score* dengan rumus seperti berikut :

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (7)$$

Akurasi digunakan untuk menunjukkan persentase prediksi yang benar dari seluruh prediksi

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

Presisi digunakan untuk mengukur seberapa baik model menghindari kesalahan dalam mendeteksi citra asli sebagai *deepfake*. Semakin tinggi nilai presisi, semakin sedikit jumlah citra asli yang secara keliru diklasifikasikan sebagai *deepfake*, sehingga mengurangi false positives. Hal ini menunjukkan bahwa model tersebut secara konsisten menghasilkan deteksi yang benar, yang merupakan hal penting dalam

menjaga keutuhan dan kepercayaan dalam sistem verifikasi gambar digital.

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

Recall digunakan untuk mengukur kemampuan model untuk mendeteksi semua citra *deepfake* dengan benar. Semakin tinggi nilai recall, semakin sedikit citra *deepfake* yang tidak terdeteksi (false negatives), sehingga memastikan bahwa hampir semua citra yang dimanipulasi berhasil diidentifikasi oleh sistem.

$$F1\ SCORE = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (10)$$

F1-Score berfungsi untuk menyeimbangkan Precision dan Recall dalam evaluasi model klasifikasi, terutama saat terdapat ketidakseimbangan kelas. Metrik ini membantu mengukur seberapa baik model dalam mendeteksi *deepfake* tanpa terlalu banyak kesalahan False Positives atau False Negatives.

3.1.5 Evaluasi Watermark

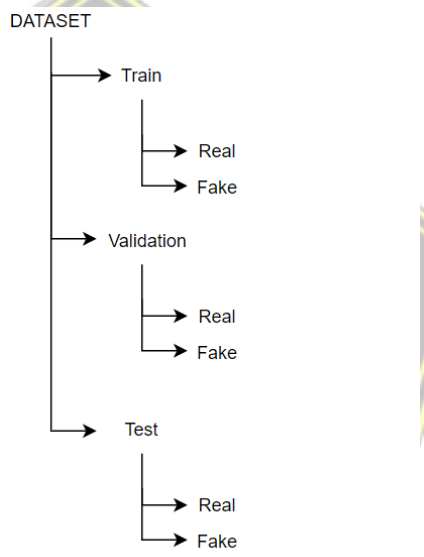
Mengukur kualitas visual gambar setelah watermark disisipkan serta ketahanan watermark terhadap manipulasi. Gambar asli tanpa watermark dibandingkan dengan gambar yang telah diberi watermark. Untuk mengevaluasi watermark, seberapa baik watermark disisipkan kedalam gambar dan tidak mengurangi kualitas dari gambar maka *Peak Signal-to-Noise Ratio* (PSNR) digunakan untuk mengukur kualitas gambar setelah watermark disisipkan. Nilai *Peak Signal-to-Noise Ratio* (PSNR) yang tinggi menunjukkan bahwa watermark tidak memengaruhi kualitas visual gambar. Rumus *Peak Signal-to-Noise Ratio* (PSNR) :

$$PSNR = 10 * \log_{10} \left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}}\right) \quad (11)$$

BAB IV
HASIL DAN PEMBAHASAN

4.1 Pengumpulan Dataset

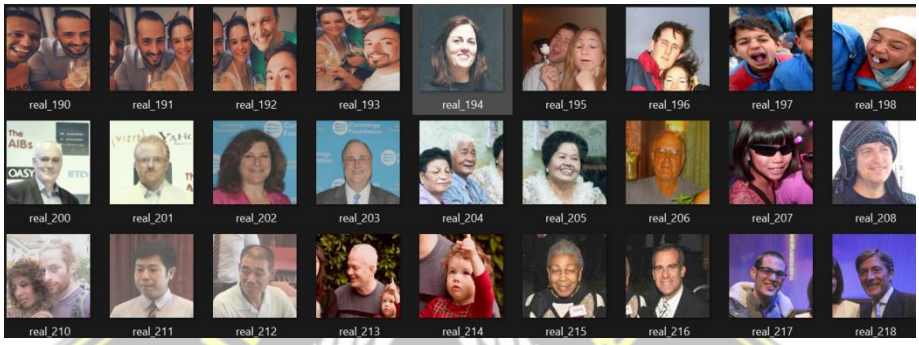
Pada tahap pengumpulan dataset, data diperoleh dari beberapa website open source seperti Kaggle dan hugging face untuk mendapatkan dataset *deepfake* image yang terdiri dari gambar asli dan palsu yang mana dataset dibagi menjadi data training, data testing dan data validation



Gambar 4. 1 Pemetaan dataset

Pembagian ini dilakukan untuk memastikan proses pengembangan model yang terstruktur dan komprehensif. Dataset disimpan dalam struktur folder terpisah dan diunggah ke Google Drive guna memudahkan integrasi dengan Google Colab dalam proses pelatihan serta evaluasi model. Pembagian dataset dilakukan dengan proporsi yang seimbang untuk memastikan akurasi dan generalisasi model.

Gambar 4. 2 data gambar Fake



Gambar 4. 3 Contoh gambar asli

Gambar diatas adalah contoh data gambar asli dan palsu yang digunakan untuk melakukan proses pelatihan data yang nantinya akan di lakukan tahapan preprocessing sebelum digunakan untuk melatih model agar bisa mendeteksi gambar asli atau palsu

4.2 Hasil *Preprocessing* Dataset

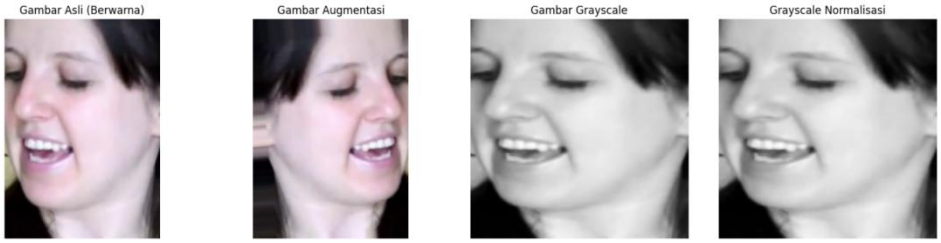
Pada tahap ini dataset dari data data yang sudah dikumpuilkan akan melewati beberapa langkah penting sebelum data digunakan untuk melatih model yakni resize,konversi ke grayscale,augmentasi dan normalisasi

Tabel 4. 1 Augmentasi data

Rescale	1./255
Rotation range	20
Width shift range	0.2
Hight shift range	0.2
Shear range	0.2
Zoom range	0.2
Horizontal range	True
Brightness range	[0.8 , 0.2]

Channel shift range	10

Pada table 4.1, beberapa transformasi diterapkan untuk meningkatkan keragaman data pelatihan dan membantu model belajar lebih baik dari variasi gambar yang lebih luas. Pertama, parameter rescale=1./255 digunakan untuk melakukan normalisasi nilai piksel gambar dari rentang 0-255 menjadi 0-1, yang membantu mempercepat proses pelatihan dan membuat model lebih stabil. Selanjutnya, rotation_range=20 memungkinkan gambar diputar secara acak hingga 20 derajat, sehingga model menjadi lebih tangguh terhadap rotasi gambar di dunia nyata. Parameter width_shift_range=0.2 dan height_shift_range=0.2 secara acak menggeser gambar secara horizontal dan vertikal hingga 20% dari dimensi gambar, meningkatkan kemampuan model untuk mengenali objek meskipun posisinya sedikit bergeser. Transformasi shear_range=0.2 memperkenalkan efek kemiringan pada gambar, sementara zoom_range=0.2 memperbesar gambar secara acak hingga 20%, membuat model lebih tahan terhadap variasi ukuran objek. Terakhir, horizontal_flip=True membalik gambar secara horizontal secara acak, yang bermanfaat untuk data dengan simetri horizontal, seperti wajah manusia. Semua transformasi ini diterapkan secara acak selama pelatihan, menciptakan dataset yang lebih beragam dan membantu mengurangi overfitting. Data testing akan dilakukan augmentasi yang didalamnya terdapat beberapa tahapan yang dimulai dari normalisasi dan lain sebaganya namun untuk data validasi dan testing hanya akan dinormalisasi



Gambar 4. 4 Hasil preprocessing

Gambar 4.4 menunjukkan tahapan pemrosesan citra yang dilakukan sebelum digunakan untuk melatih model. Gambar pertama adalah gambar asli dalam format berwarna (RGB). Ini adalah citra yang diambil langsung dari dataset sebelum mengalami perubahan atau pemrosesan. Gambar kedua adalah hasil augmentasi

data, di mana berbagai transformasi seperti rotasi, pergeseran, shear, zoom, dan flipping diterapkan untuk meningkatkan variasi data. Augmentasi ini bertujuan untuk membuat model lebih robust terhadap variasi input, sehingga tidak hanya mengandalkan pola statis dalam mendeteksi *deepfake*. Gambar ketiga menunjukkan hasil konversi ke grayscale, di mana citra berwarna diubah menjadi hitam-putih. Konversi ini menghilangkan informasi warna dan hanya mempertahankan intensitas cahaya dari setiap piksel. Dengan cara ini, model lebih fokus pada pola tekstur dan bentuk tanpa terganggu oleh variasi warna. Gambar keempat adalah hasil normalisasi dari gambar grayscale. Proses normalisasi dilakukan dengan membagi nilai piksel dengan 255, sehingga semua nilai berada dalam rentang [0,1]. Ini membantu model dalam konvergensi lebih cepat saat pelatihan karena menghindari skala nilai yang terlalu besar atau kecil.

Urutan pemrosesan ini memastikan bahwa gambar yang dimasukkan ke dalam model sudah dalam format yang optimal, dengan menghilangkan informasi yang tidak relevan daCn meningkatkan variasi data agar model dapat mendeteksi *deepfake* dengan lebih akurat.

4.3 Hasil Pelatihan Model

4.3.1 Model *Deepfake*

Pada proses ini model dilatih menggunakan dataset yang sudah melewati tahap preprocessing dimana tahapan selanjutnya adalah membangun sebuah model deteksi gambar *deepfake* menggunakan TensorFlow dan Keras dengan arsitektur berbasis *transfer learning*. Model dimulai dengan input gambar grayscale berukuran 128x128 piksel dengan 1 saluran (channel). Karena model *DenseNet121* pralatih dari ImageNet membutuhkan input RGB, digunakan layer kustom GrayscaleToRGB untuk mengonversi input grayscale menjadi tiga saluran. Setelah konversi, *DenseNet121* digunakan sebagai *feature extractor* tanpa menyertakan layer klasifikasi atasnya (*include_top=False*). Output dari DenseNet diproses melalui layer GlobalAveragePooling2D untuk meratakan hasil ekstraksi fitur, kemudian diterapkan Dropout dengan rasio 0.5 untuk mencegah overfitting seperti tabel 4.2. Akhirnya, sebuah layer dense dengan fungsi aktivasi sigmoid digunakan

untuk mengklasifikasikan gambar dalam dua kelas, yaitu asli atau palsu (*deepfake*). Model dikompilasi menggunakan optimizer *Adam* dengan fungsi kerugian *binary_crossentropy* dan metrik akurasi, yang cocok untuk tugas klasifikasi biner.

Tabel 4. 2 Arsitektur model dan layer

Arsitektur Model Dan Layer	Keterangan
Base model	DenseNet 121
Input layer	(128,128,1)
Global average pooling	true
Dropout rate	0.5
Output activation	sigmoid

Tabel 4. 3 Optimizer & Loss Function

Optimizer dan *Loss Function*	Keterangan
Optimizer	Adam
Loss function	Binary crossentropy
Matriks	acuracy

Tabel 4. 4 Trainning parameter

Trainning Parameter	Keterangan
Epoch	10
Batch size	32
Matriks	acuracy
Patience	3
Restore best weight	True

Dari table 4.4, melatih model dengan menggunakan data pelatihan (train_data dan train_labels) serta data validasi (val_data dan val_labels) selama maksimal 10 epoch dengan ukuran batch 32. Dua callback digunakan untuk mengoptimalkan pelatihan: EarlyStopping memonitor *validation loss* dan secara otomatis menghentikan pelatihan jika tidak ada perbaikan selama 5 epoch berturut-turut, sambil mengembalikan bobot model terbaik yang tercapai selama pelatihan; *ModelCheckpoint* menyimpan model terbaik berdasarkan *validation loss* ke file

'best_model.keras'. *Callback* ini membantu mencegah overfitting dan memastikan model yang disimpan memiliki performa validasi terbaik.

Gambar 4. 5 Gambar hasil prediksi

Gambar yang ditampilkan merupakan hasil prediksi model deteksi *deepfake* terhadap lima gambar wajah dengan menggunakan teknik pemrosesan gambar, di mana setiap gambar memiliki label prediksi berupa "*Fake*" atau " " di bagian atasnya yang menunjukkan apakah gambar tersebut diklasifikasikan sebagai *deepfake* atau asli oleh model. Warna hijau kebiruan dengan pola khas menunjukkan bahwa gambar telah melalui teknik pemrosesan.

4.3.2 *Watermarking* LSB

Pada tahapan ini *watermarking* menggunakan metode *Least significant bit (LSB)* dilakukan dengan menyisipkan dan mengekstrak bit pesan secara langsung dari bit paling akhir (LSB) pada setiap channel warna di setiap piksel gambar. Pada tahap penyisipan (fungsi *encode_image*), program mengambil pesan yang akan disisipkan, mengubahnya menjadi representasi biner (bit per bit), kemudian secara berurutan menggantikan LSB pada masing-masing channel (R, G, dan B) dengan bit pesan tersebut, setelah semua bit pesan ditanam, proses penyisipan dihentikan, dan sebagai penanda akhir pesan ditambahkan karakter null ('\0' atau biner 00000000) agar proses pembacaan tahu kapan harus berhenti.

Pada tahap pembacaan (fungsi *decode_image*), kode mengambil nilai setiap piksel, mengekstrak LSB dari tiap channel, menyusun kembali bit-bit tersebut menjadi byte (8 bit), dan mengonversinya menjadi karakter ASCII proses ini berlanjut hingga terdeteksinya byte null yang menandai akhir pesan. Dengan cara ini, watermark (pesan *rahasia*) dapat disisipkan dan dibaca tanpa terlihat perubahan mencolok pada gambar, karena modifikasi hanya terjadi pada bit paling rendah setiap channel.

Hasil dari proses *watermarking* LSB menunjukkan bahwa watermark dapat disisipkan dan diekstraksi dengan tingkat keberhasilan tinggi tanpa mengubah kualitas visual gambar secara signifikan. Gambar di bawah ini menampilkan perbandingan antara gambar asli dan gambar yang sudah disisipi watermark

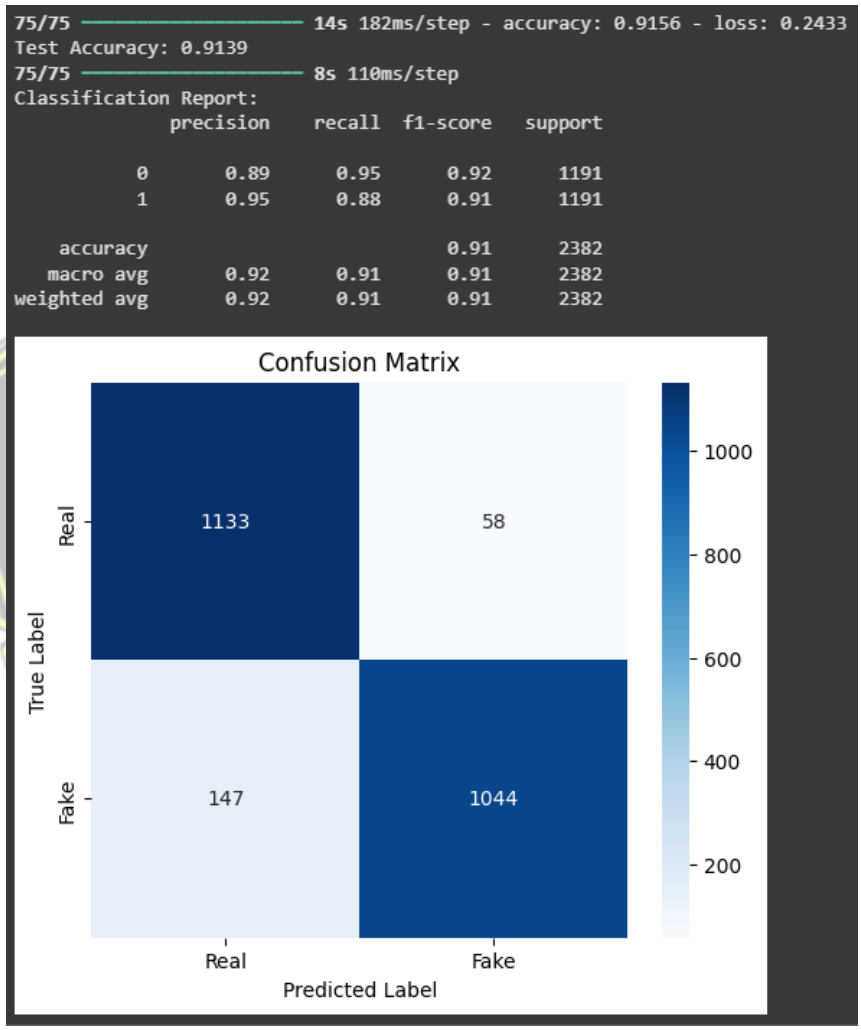
Gambar 4. 6 Gambar asli dan watermark

Gambar yang ditampilkan menunjukkan hasil dari proses *watermarking* digital pada sebuah citra. Pada bagian atas, terdapat tiga gambar yang merepresentasikan tahapan dalam proses ini. Gambar pertama adalah gambar asli sebelum diberikan watermark, yang berfungsi sebagai referensi awal. Gambar kedua adalah gambar yang telah disisipkan watermark tersembunyi. Secara visual, gambar ini tampak hampir identik dengan gambar asli, menandakan bahwa watermark disisipkan dengan metode yang tidak mengganggu tampilan utama. Gambar ketiga menunjukkan hasil ekstraksi watermark dari gambar yang telah dimodifikasi. Teks "Watermark Tersembunyi" mengindikasikan bahwa watermark berhasil disisipkan dan dapat diambil kembali dari gambar watermarked.

4.4 Hasil Evaluasi

4.4.1 Evaluasi Model CNN

Model CNN DenseNet121 diterapkan untuk mendeteksi apakah sebuah gambar merupakan *deepfake* atau asli. Model ini diuji menggunakan dataset yang terdiri dari 2.382 gambar, dengan komposisi 50% gambar asli dan 50% gambar *deepfake*.



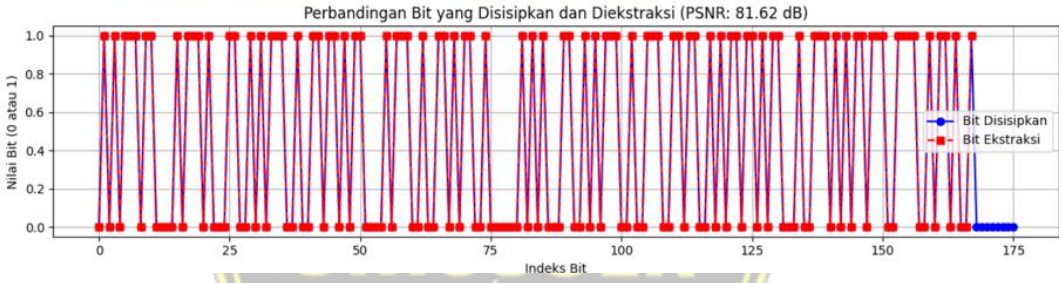
Gambar 4. 7 Hasil accuracy,precission recall dan f1-score

Gambar 7 menunjukkan ringkasan metrik evaluasi model. Hasil pelatihan model menunjukkan akurasi sebesar 91.56% dengan loss 0.2433 pada dataset uji. Dari hasil ini, dapat disimpulkan bahwa model mampu membedakan *deepfake* dengan baik.

Tabel 4. 5 Tabel evaluasi

Matrix	Real	Fake
Akurasi	$\frac{1044+1133}{1044+1133+58+147} = \frac{2177}{2382} = 0.91$	
Presisi	$\frac{1133}{1133+58} = \frac{1133}{1191} = 0.89$	$\frac{1044}{1044+147} = \frac{1044}{1191} = 0.95$
Recall	$\frac{1133}{1133+147} = \frac{1133}{1280} = 0.95$	$\frac{1044}{1044+58} = \frac{1044}{1102} = 0.88$
F1-Score	$2 * \frac{0.89 * 0.95}{0.89+0.95} = 2 * \frac{0.8455}{1.84} = 0.92$	$2 * \frac{0.95 * 0.88}{0.95+0.88} = 2 * \frac{0.836}{1.83} = 0.91$

4.4.2 Evaluasi *Watermarking*



Gambar 4. 8 Gambar asli dan *watermark*

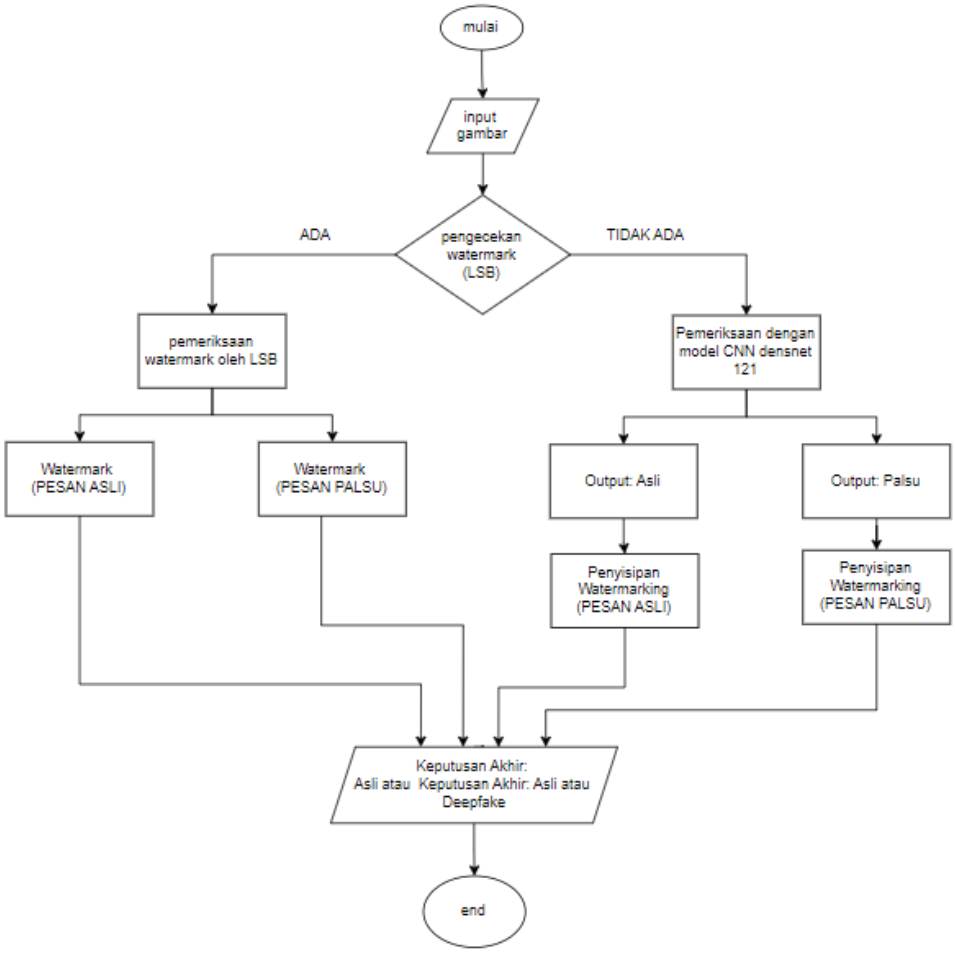
Di bagian bawah, terdapat grafik yang membandingkan bit-bit watermark yang disisipkan dengan bit-bit yang berhasil diekstraksi. Grafik ini memiliki sumbu X yang menunjukkan indeks bit dalam watermark dan sumbu Y yang merepresentasikan nilai biner dari masing-masing bit (0 atau 1). Dua jenis data ditampilkan dalam grafik, yaitu bit yang disisipkan (ditandai dengan garis biru dan titik bulat) serta bit yang diekstraksi (ditandai dengan garis merah dan kotak merah). Dari pola grafik, terlihat bahwa sebagian besar bit yang diekstraksi sesuai dengan bit yang disisipkan, menunjukkan bahwa proses *watermarking* berhasil.

Secara keseluruhan, hasil ini menunjukkan bahwa metode *watermarking* yang digunakan cukup efektif dalam menyisipkan informasi tanpa mengubah

tampilan gambar secara signifikan. Selain itu, watermark dapat diekstraksi kembali dengan tingkat keberhasilan yang tinggi. Sedikit perbedaan dalam hasil ekstraksi bisa menjadi indikasi adanya faktor eksternal yang mempengaruhi proses pengambilan watermark, tetapi secara umum, metode ini bekerja dengan baik dalam menjaga keutuhan informasi yang disisipkan.

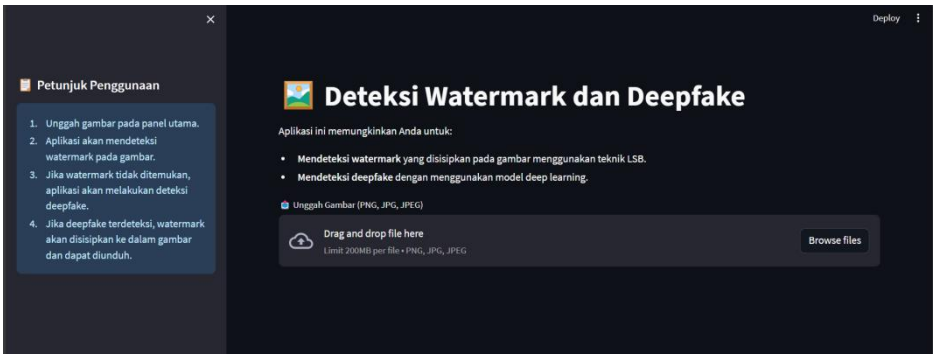
4.5 Implementasi Model Deteksi *Deepfake* dan *Watermarking*

Pada tahap ini, model yang telah dilatih akan dikombinasikan dengan teknik *watermarking Least significant bit* (LSB). Metode ini digunakan untuk menyisipkan informasi ke dalam citra dengan cara menggantikan bit paling tidak signifikan, sehingga tidak mengubah struktur utama gambar secara signifikan. Integrasi model *Deep Learning* dengan *watermarking* LSB bertujuan untuk meningkatkan keamanan dan keandalan sistem dalam mendeteksi citra *deepfake*



Gambar 4. 9 Flowchart Alur Sistem

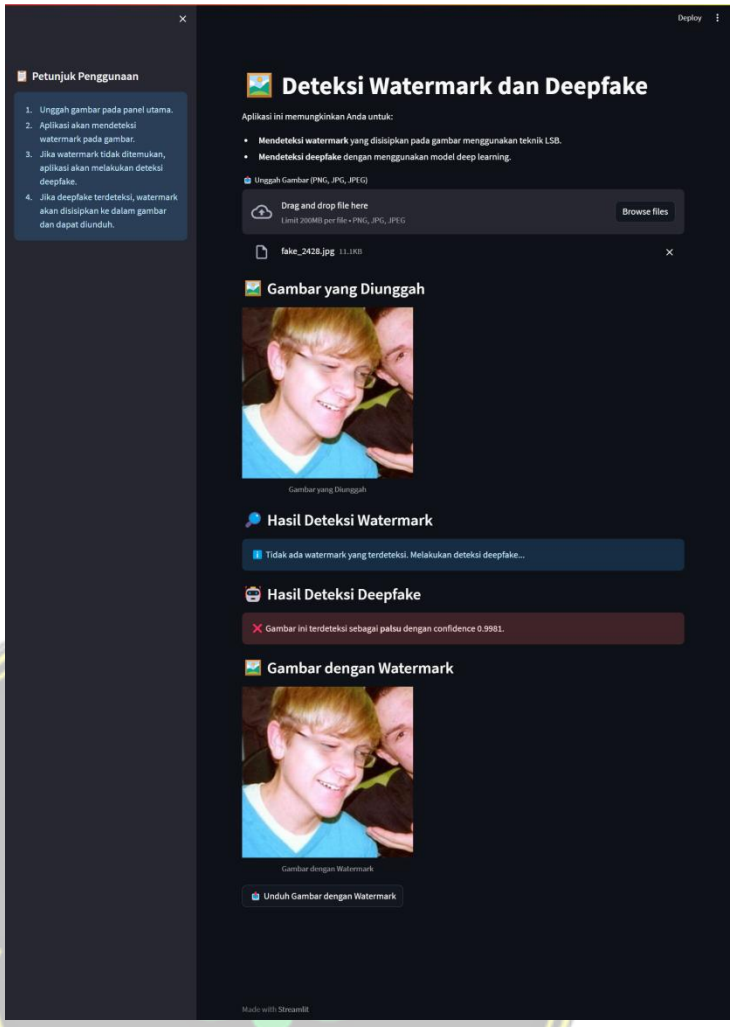
Dari alur system sesuai flowchart tersebut maka akan di implementasikan dalam bentuk tampilan website yang nantinya user bisa mengecek gambar apakah gambar yang diperiksa termasuk kedalam *deepfake* atau asli,berikut adalah tampilan halaman utama dari system tersebut



Gambar 4. 10 Tampilan website

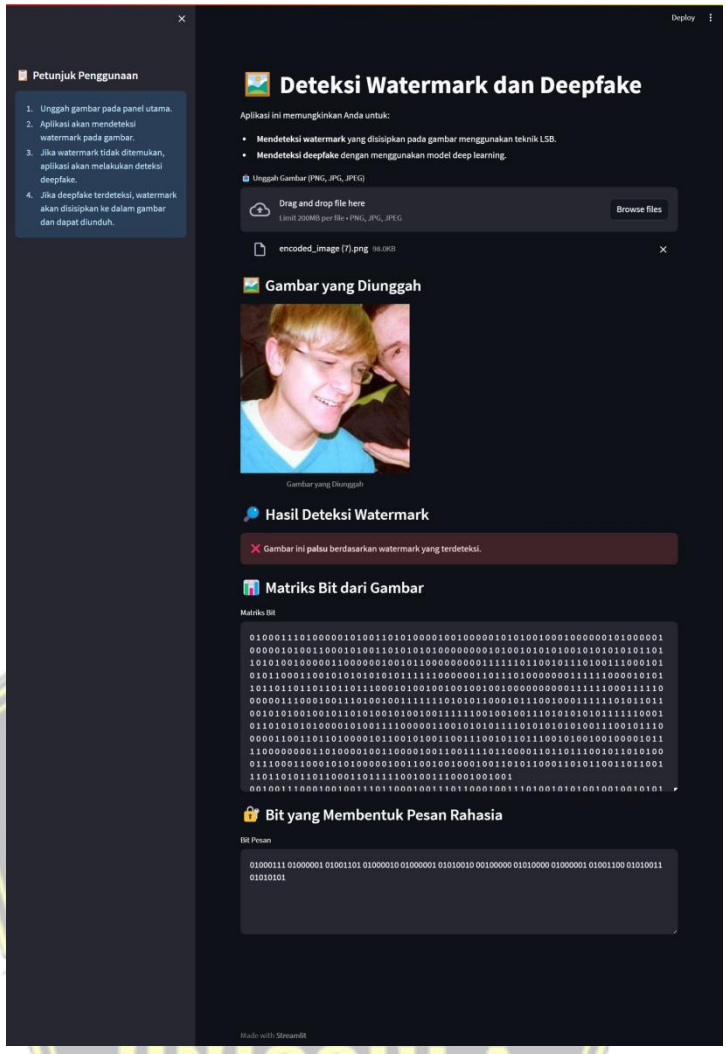
Dari tampilan tersebut user hanya perlu memasukan gambar dalam format PNG,JPG,JPEG dll, yang nantinya sistem akan memeriksa watermark didalam gambar jika gambar yang dimasukkan tidak memiliki watermark maka akan dilanjutkan pada proses deteksi gambar oleh model CNN densenet-121,dan jika gambar yang dimasukkan oleh user memiliki watermark nantinya watermark yang disisipkan oleh system akan di baca dan menampilkan bit dari pada pesan yang disisipkan di dalam gambar.

Dalam pengujian sistem deteksi deepfake, gambar yang telah dimanipulasi menggunakan remaker.ai digunakan sebagai data uji untuk mengevaluasi performa model. Remaker.ai merupakan salah satu alat berbasis kecerdasan buatan yang mampu mengubah, menyunting, atau memodifikasi wajah dalam gambar dengan tingkat realisme tinggi. Dengan memanfaatkan gambar hasil manipulasi dari remaker.ai, sistem deteksi deepfake dapat diuji dalam mendeteksi perubahan subtil yang mungkin sulit dikenali oleh mata manusia, seperti perbedaan tekstur kulit, pencahayaan yang tidak wajar, atau ketidaksesuaian antara ekspresi wajah dan bentuk struktur aslinya. Penggunaan dataset ini membantu menguji sejauh mana model deep learning dapat membedakan antara gambar asli dan deepfake, serta meningkatkan keakuratan deteksi dengan teknik augmentasi dan fine-tuning parameter model.



Gambar 4. 11 Hasil deteksi gambar tanpa watermark

Ketika hasil deteksi keluar maka user akan diberi pilihan untuk mendownload gambar yang sudah diberikan watermark oleh system untuk validasi bahwa gambar yang dimasukkan sudah dicek keasliannya dan nantinya ketika user memasukkan lagi gambar yang sudah didownload maka system hanya perlu memberikan hasil berdasarkan watermark yang sudah disisipi sebelumnya sehingga disini fungsi watermark akan mempercepat proses deteksi selain menjadi validasi bahwa gambar asli atau palsu.



Gambar 4. 12 Hasil deteksi gambar dengan watermark

Dari gambar diatas didapatkan bit yang membentuk pesan rahasia yang disisipkan oleh sistem sebelumnya yang mana jika di konversi maka akan membentuk pesan “GAMBAR ASLI” atau “GAMBAR PALSU” sesuai dengan hasil deteksi model DenseNet-121.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan implementasi dan evaluasi sistem deteksi deepfake, dapat disimpulkan bahwa integrasi model CNN DenseNet-121 dengan teknik watermarking LSB telah menghasilkan solusi yang efektif dalam verifikasi keaslian gambar digital. Model yang dilatih menggunakan optimasi Adam dan binary cross-entropy menunjukkan kinerja tinggi dengan akurasi 91%, precision 0.92, recall 0.91, dan F1-score 0.91. Hasil ini menunjukkan bahwa model tidak hanya mampu mendeteksi deepfake dengan presisi tinggi, tetapi juga menjaga keseimbangan antara recall dan precision, sehingga meminimalkan kesalahan klasifikasi.

Selain itu, teknik watermarking LSB memungkinkan penyisipan bukti forensik digital yang imperceptible ke dalam gambar tanpa mengurangi kualitas visualnya. Watermark ini dapat diekstraksi kembali dengan akurasi tinggi, memberikan lapisan tambahan dalam memastikan integritas gambar. Ketika watermark tidak terdeteksi, sistem tetap dapat mengklasifikasikan gambar sebagai asli atau palsu dengan tingkat keandalan yang tinggi. Pendekatan integratif ini membuktikan bahwa kombinasi deep learning dan watermarking dapat meningkatkan keamanan serta kepercayaan terhadap data digital.

5.2 Saran

Berdasarkan hasil implementasi dan kesimpulan yang diperoleh, terdapat beberapa saran yang dapat dijadikan pertimbangan untuk pengembangan dan penelitian ini. Diperlukan pengujian lebih lanjut dengan menggunakan arsitektur *Deep Learning* yang lebih kompleks atau mengombinasikan beberapa model (ensemble learning) untuk meningkatkan akurasi dan ketahanan sistem dalam mendeteksi berbagai jenis manipulasi citra yang lebih halus dan kompleks. Disarankan untuk menguji sistem dengan dataset yang lebih besar dan bervariasi, termasuk berbagai jenis format gambar, resolusi, dan kondisi pencahayaan yang berbeda. Hal ini dapat membantu mengukur kemampuan generalisasi sistem dalam mendeteksi *deepfake* di berbagai situasi nyata.

DAFTAR PUSTAKA

A C, P., & M T, S. (2023). An Overview on Research Trends, Challenges, Applications and Future Direction in Digital Image *Watermarking*. *International Research Journal on Advanced Science Hub*, *5*(01), 8–14. https://doi.org/10.47392/irjash.2023.002

Albelwi, S. A. (2022). Deep Architecture based on DenseNet-121 Model for Weather Image Recognition. *International Journal of Advanced Computer Science and Applications*, *13*(10), 559–565. https://doi.org/10.14569/IJACSA.2022.0131065

Almars, A. M. (2021). *Deepfake*s detection techniques using *Deep Learning*: a survey. *Journal of Computer and Communications*, *9*(05), 20–35.

Alzahrani, A. (2024). Digital Image Forensics: An Improved DenseNet Architecture for Forged Image Detection. *Engineering, Technology and Applied Science Research*, *14*(2), 13671–13680. https://doi.org/10.48084/etasr.7029

Begum, M., & Uddin, M. S. (2020). Digital image *watermarking* techniques: A review. *Information (Switzerland)*, *11*(2). https://doi.org/10.3390/info11020110

Boato, G., Conotter, V., De Natale, F. G. B., & Fontanari, C. (2009). *Watermarking* robustness evaluation based on perceptual quality via genetic algorithms. *IEEE Transactions on Information Forensics and Security*, *4*(2), 207–216. https://doi.org/10.1109/TIFS.2009.2020362

Bose, S., Madhulika, Acharjee, S., Chowdhury, S. R., Chakraborty, S., & Dey, N. (2014). Effect of *watermarking* in vector quantization based image compression. *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies, ICCICCT 2014*, *December*, 503–508. https://doi.org/10.1109/ICCICCT.2014.6993014

Chopra, D. (2012). Lsb Based Digital Image *Watermarking* For Gray Scale Image. *IOSR Journal of Computer Engineering*, *6*(1), 36–41. https://doi.org/10.9790/0661-0613641

Faheem, Z. Bin, Ali, M., Raza, M. A., Arslan, F., Ali, J., Masud, M., &

Shorfuzzaman, M. (2022). Image *Watermarking* Scheme Using LSB and Image Gradient. *Applied Sciences (Switzerland)*, *12*(9), 1–12. https://doi.org/10.3390/app12094202

Hussain, M., Bird, J. J., & Faria, D. R. (2019). A study on CNN transfer learning for image classification. *Advances in Intelligent Systems and Computing*, *840*, 191–202. https://doi.org/10.1007/978-3-319-97982-3_16

Karnouskos, S. (2020). Artificial Intelligence in Digital Media: The Era of *Deepfake*s. *IEEE Transactions on Technology and Society*, *1*(3), 138–147. https://doi.org/10.1109/tts.2020.3001312

Khan, S. A., & Dang-Nguyen, D.-T. (2023). *Deepfake Detection: A Comparative Analysis*. *1*(1), 1–28. http://arxiv.org/abs/2308.03471

Lecun, Y., Bengio, Y., Hinton, G., Lecun, Y., Bengio, Y., & Hinton, G. (2023). *Deep Learning*. *521*(7553), 436–444.

License, C. A., & Rizvee, M. M. (2023). *Retracted : Comparative Analysis of Deepfake Image Detection*. *2021*. https://doi.org/10.1155/2021/3111676

Masood, M., Nawaz, M., Malik, K. M., Javed, A., Irtaza, A., & Malik, H. (2023). *Deepfake*s generation and detection: state-of-the-art, open challenges, countermeasures, and way forward. *Applied Intelligence*, *53*(4), 3974–4026. https://doi.org/10.1007/s10489-022-03766-z

Memon, N., & Wong, P. W. (1998). Protecting Digital Media Content. *Communications of the ACM*, *41*(7), 35–43. https://doi.org/10.1145/278476.278485

Parthasarathy, C., & Nagar, J. (2009). Increased Robustness of Lsb Audio Steganography By Reduced Distortion Lsb. *Audio*.

Patel, Y., Tanwar, S., Bhattacharya, P., Gupta, R., Alsuwian, T., Davidson, I. E., & Mazibuko, T. F. (2023). An Improved Dense CNN Architecture for *Deepfake* Image Detection. *IEEE Access*, *11*(March), 22081–22095. https://doi.org/10.1109/ACCESS.2023.3251417

Praveen Chakravarthy, S., Gunasundari, C., Selva Bhuvaneswari, K., Sharma, B., & Chowdhury, S. (2022). *Convolutional Neural Network* (CNN) for Image Detection and Recognition in Medical Diagnosis. *IET Conference*

Proceedings, *2022*(26), 357–361. https://doi.org/10.1049/icp.2023.0579

Sanivarapu, P. V., Rajesh, K. N. V. P. S., & Hosny, K. M. (2022). *applied sciences Digital Watermarking System for Copyright Protection and Authentication of Images Using Cryptographic Techniques*.

Satrio, B., Poetro, W., Mulyono, S., & Pramesti, V. A. (n.d.). *Prediksi Penyakit Batu Ginjal dengan Menerapkan Convolutional Neural Network*. 153–162.

Sharma, R., Sharma, K., & Khanna, A. (2020). *Study of Supervised Learning and Unsupervised Learning*. *June*.

Shrestha, A., & Mahmood, A. (2019). Review of *Deep Learning* algorithms and architectures. *IEEE Access*, *7*, 53040–53065. https://doi.org/10.1109/ACCESS.2019.2912200

Subramanyam, A. V., Emmanuel, S., & Kankanhalli, M. S. (2012). Robust *watermarking* of compressed and encrypted JPEG2000 images. *IEEE Transactions on Multimedia*, *14*(3 PART 2), 703–716. https://doi.org/10.1109/TMM.2011.2181342

Swaminathan, A., Mao, Y., & Wu, M. (2006). Robust and secure image hashing. *IEEE Transactions on Information Forensics and Security*, *1*(2), 215–230. https://doi.org/10.1109/TIFS.2006.873601

Tolosana, R., Vera-Rodriguez, R., Fierrez, J., Morales, A., & Ortega-Garcia, J. (2020). *Deepfake*s and beyond: A Survey of face manipulation and fake detection. *Information Fusion*, *64*, 131–148. https://doi.org/10.1016/j.inffus.2020.06.014

Wan, W., Wang, J., Zhang, Y., Li, J., Yu, H., & Sun, J. (n.d.). *A Comprehensive Survey on Robust Image Watermarking A R T I C L E I N F O*. 1–26.