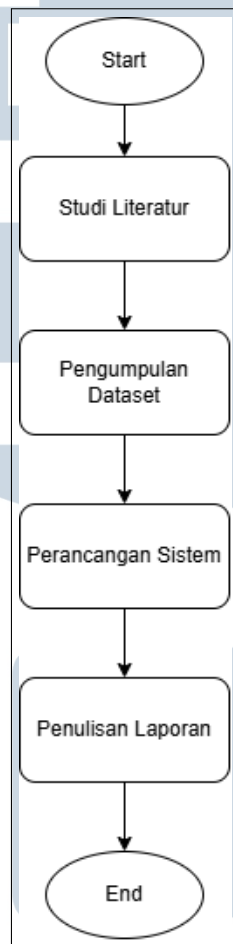


BAB 3

METODOLOGI PENELITIAN

Penelitian dilakukan klasifikasi manipulasi wajah pada gambar *deepfake* dengan *fast fourier transform* dan *EfficientNet*. Gambar 3.1 merupakan beberapa tahap penelitian yang dilakukan.



Gambar 3.1. Gambaran umum penelitian

3.1 Studi Literatur

Studi literatur dilakukan untuk pemahaman, pengkajian serta eksplorasi dari semua teori terkait dengan klasifikasi *deepfake*, *EfficientNet*, dan *fast fourier transform* yang dapat mendukung penelitian ini. Proses ini bertujuan untuk mengidentifikasi celah penelitian, metode terbaru, dan pendekatan yang relevan untuk memastikan dasar teoritis yang kuat dalam pengembangan model deteksi

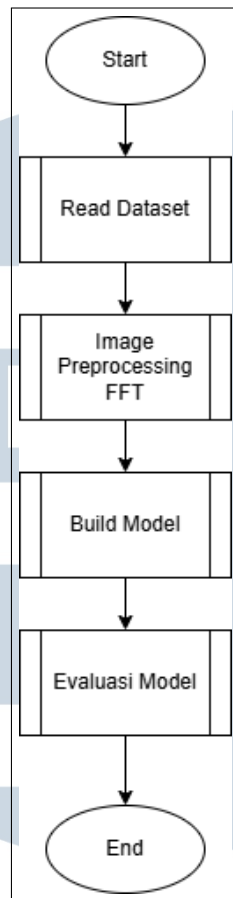
deepfake. Proses ini dilakukan dengan memanfaatkan sejumlah sumber akademis terpercaya seperti buku, jurnal, dan artikel yang berkaitan dengan topik penelitian yang dilakukan. Analisis terhadap semua sumber tersebut dapat memberikan wawasan kepada penulis tentang kelebihan dan keterbatasan metode yang ada sehingga dapat menjadi acuan dalam perancangan solusi yang lebih efektif,

3.2 Pengumpulan Dataset

Untuk mengembangkan model deteksi *deepfake*, diperlukan pengumpulan *dataset* yang relevan dengan penelitian ini. Pada tahap ini, digunakan *dataset* yang diperoleh dari Kaggle melalui tautan (<https://www.kaggle.com/datasets/manjilkarki/deepfake-and-real-images>). *Dataset* ini berisi kumpulan gambar yang sudah diberi label berdasarkan kelasnya yaitu *real* dan *fake*. Setiap gambar mencakup berbagai variasi teknik manipulasi wajah, sehingga proses pelatihan model dapat dilakukan secara efektif. Struktur *dataset* terdiri dari 4.206 gambar untuk pelatihan (*training*), 400 gambar untuk validasi (*validation*), dan 200 gambar untuk pengujian (*testing*), dengan rasio pembagian yang cukup seimbang untuk memastikan kualitas evaluasi model.

3.3 Perancangan Sistem

Gambar 3.2 merupakan *flowchart* perancangan sistem yang menunjukkan tahapan-tahapan dalam mengklasifikasikan gambar *deepfake*. Proses dimulai dengan membaca *dataset* yang telah diunduh sebelumnya, kemudian dilanjutkan dengan praproses gambar menggunakan metode *Fast Fourier Transform* (FFT) untuk mengubah gambar ke dalam format domain frekuensi. Setelah gambar dalam format frekuensi diperoleh, langkah selanjutnya adalah membangun model klasifikasi menggunakan EfficientNet. Model tersebut kemudian dilatih menggunakan data hasil transformasi FFT. Tahap terakhir adalah evaluasi model yang bertujuan untuk mengukur performa sistem deteksi yang telah dibangun, dengan menggunakan metrik seperti akurasi, presisi, recall, dan F1-score. Metrik-metrik tersebut digunakan untuk menilai seberapa efektif model dalam mendeteksi gambar *deepfake*.



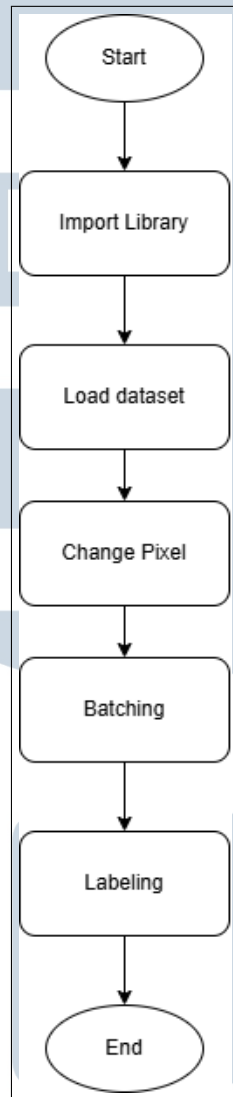
Gambar 3.2. Flowchart Perancangan Sistem

3.3.1 Read Dataset

Dalam proses ini, dilakukan *import* dataset yang telah diunduh dari Kaggle untuk pelatihan, validasi, dan pengujian model. Dataset tersebut tersusun dalam tiga folder utama, yaitu folder *training* yang berisi 4.206 data, folder *validation* dengan 400 data, serta folder *testing* yang terdiri dari 200 data. Setiap folder memiliki subfolder yang dinamai sesuai dengan label kelas, yaitu *Real* dan *Fake*, yang memudahkan dalam pengorganisasian dan pelabelan data secara otomatis.

Seluruh gambar dalam folder tersebut dimuat menggunakan library *TensorFlow* melalui fungsi `image_dataset_from_directory`, yang secara efisien membaca gambar sekaligus mengonversi label berdasarkan struktur folder. Selanjutnya, setiap gambar diubah ukurannya agar memiliki dimensi pixel yang konsisten, yang penting agar data dapat diproses secara seragam dalam model. Data gambar kemudian dibagi ke dalam *batch* dengan ukuran tertentu untuk mengoptimalkan penggunaan memori dan mempercepat proses pelatihan. Label

dari gambar diberikan dalam format integer agar dapat langsung digunakan dalam proses klasifikasi biner. Seluruh proses pengambilan dan penyiapan dataset ini dapat dilihat secara lebih rinci pada flowchart pada Gambar 3.3.



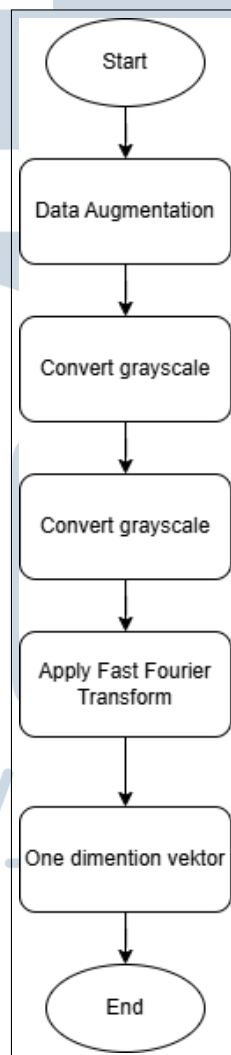
Gambar 3.3. Flowchart Read Dataset

3.3.2 Image Preprocessing menggunakan Fast Fourier Transform

Proses ini menggunakan transformasi domain frekuensi untuk mengekstraksi fitur tambahan dari gambar dengan memanfaatkan *Fast Fourier Transform* (FFT). Sebelum ekstraksi fitur dilakukan, data pelatihan terlebih dahulu mengalami proses *data augmentation* yang meliputi teknik seperti *horizontal flip*, rotasi acak, dan zoom acak. Tujuan dari augmentasi ini adalah untuk meningkatkan

variasi dan keragaman data sehingga model dapat belajar dari berbagai variasi gambar, sehingga mengurangi risiko *overfitting* dan meningkatkan kemampuan generalisasi model.

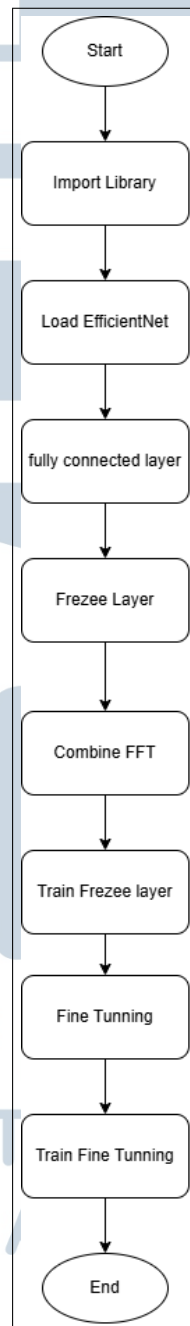
Setelah augmentasi, gambar kemudian dikonversi ke dalam format *grayscale* untuk menyederhanakan data dan mengurangi kompleksitas komputasi. Selanjutnya, gambar diubah ukurannya agar konsisten dan siap untuk proses FFT, yang digunakan untuk mentransformasi data gambar dari domain spasial ke domain frekuensi. Dari hasil FFT, nilai magnitudo diambil sebagai representasi fitur frekuensi yang penting dan diubah menjadi vektor satu dimensi yang dapat digunakan sebagai input tambahan dalam model pembelajaran. Seluruh proses ini dapat dilihat secara rinci pada Gambar 3.4.



Gambar 3.4. Flowchart Data Preprocessing dan mengaplikasikan Fast Fourier Transform

3.3.3 Build Model

Gambar 3.5 merupakan *flowchart* dari proses pembangunan model menggunakan arsitektur *EfficientNet*. Pada tahap ini, dilakukan proses pelatihan (*training*) model dengan menggunakan *dataset* yang telah dilakukan *image preprocessing*.



Gambar 3.5. Flowchart build model EfficientNet

Penelitian ini menggunakan model dasar *EfficientNet* yang secara default model *EfficientNet* telah dilatih sebelumnya menggunakan dataset *ImageNet*. Model ini dimanfaatkan untuk melakukan ekstraksi fitur visual dari gambar, dengan menghapus bagian *fully connected layer* (*include_top=False*) agar dapat disesuaikan dengan kebutuhan klasifikasi biner pada penelitian ini. Pada tahap awal pelatihan, seluruh lapisan dari model *EfficientNetB0* dibekukan (*frozen*) untuk menjaga bobot awal hasil pelatihan sebelumnya. Selain itu, ditambahkan pula input tambahan berupa fitur domain frekuensi hasil transformasi *Fast Fourier Transform* (*FFT*) yang telah dikonversi menjadi vektor satu dimensi. Fitur visual dari *EfficientNetB0* dan fitur frekuensi dari *FFT* kemudian digabungkan menggunakan lapisan *Concatenate* dan diteruskan ke lapisan klasifikasi akhir dengan fungsi aktivasi *sigmoid* untuk menghasilkan output biner (*Real* atau *Fake*).

Model dikompilasi menggunakan *Adam optimizer* dengan fungsi loss *binary crossentropy*, serta diukur menggunakan metrik *accuracy*. Proses pelatihan awal dengan dataset pelatihan dan validasi. Setelah tahap pelatihan awal, dilakukan proses *fine-tuning* dengan membuka kembali beberapa lapisan terakhir dari model dasar agar bobot dapat diperbarui sesuai dataset yang digunakan dalam penelitian.

3.3.4 Evaluasi Model

Dalam proses ini, evaluasi kinerja model dilakukan menggunakan data dari folder *testing* yang berisi gambar-gambar baru yang belum pernah dilihat oleh model selama pelatihan maupun validasi. Hal ini penting untuk mengukur kemampuan model dalam menggeneralisasi terhadap data nyata di luar data latih. Dengan menggunakan data *testing* yang independen, hasil evaluasi menjadi lebih objektif dan mencerminkan performa model di kondisi dunia nyata.

Evaluasi dilakukan dengan menggunakan metode *confusion matrix*, yang memetakan hasil prediksi model terhadap label sebenarnya untuk dua kelas, yaitu *Real* dan *Fake*. Dari *confusion matrix* ini kemudian dihitung metrik-metrik evaluasi seperti *akurasi* yang menunjukkan proporsi prediksi yang benar, *presisi* yang mengukur ketepatan prediksi positif, *recall* yang menunjukkan kemampuan model dalam menemukan semua kasus positif, serta *F1-score* yang merupakan rata-rata antara presisi dan recall. Metrik-metrik ini memberikan gambaran menyeluruh tentang kekuatan dan kelemahan model dalam klasifikasi biner.