

## BAB 3 METODOLOGI PENELITIAN

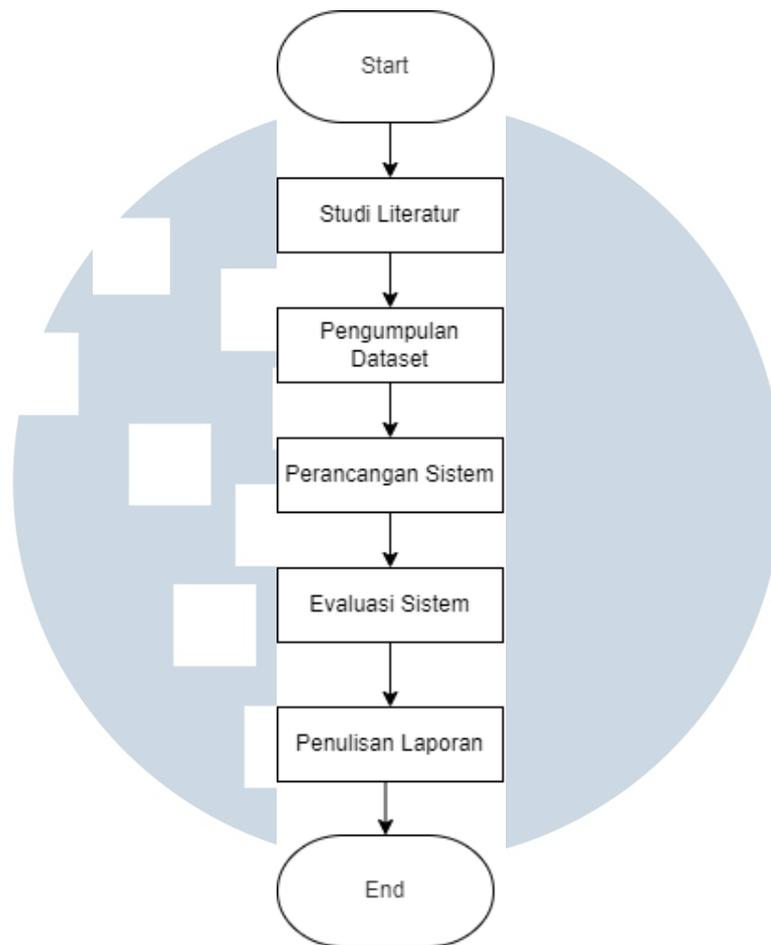
### 3.1 Desain Penelitian

Penelitian ini menggunakan metode eksperimen dengan pendekatan deep learning berbasis arsitektur kombinasi antara ResNet-50 dan Long Short-Term Memory (LSTM) untuk mendeteksi konten *deepfake* secara efektif dan akurat. Dalam proses deteksi ini, *dataset* ForgeryNet digunakan sebagai sumber data utama karena menyediakan berbagai variasi video manipulatif yang merepresentasikan kondisi nyata dalam skenario penyebaran *deepfake*. ResNet-50, yang merupakan *deep convolutional neural network* yang telah terbukti handal dalam tugas-tugas klasifikasi dan ekstraksi fitur visual, digunakan dalam tahap awal untuk mengekstraksi fitur spasial dari setiap *frame* video. Hal ini penting karena model ResNet-50 secara struktural hanya menerima input dalam bentuk gambar, bukan video secara langsung. Oleh karena itu, setiap video pada *dataset* perlu diproses terlebih dahulu dengan diekstraksi menjadi serangkaian *frame* individual.

Ekstraksi ini bertujuan untuk memperoleh informasi visual yang detail, seperti tekstur wajah, ketajaman garis tepi, pola kulit, serta kondisi pencahayaan yang mungkin menunjukkan adanya manipulasi. Setelah fitur spasial dari masing-masing *frame* diperoleh melalui ResNet-50, fitur-fitur tersebut kemudian disusun secara berurutan dan diberikan sebagai input ke model LSTM. LSTM, yang merupakan varian dari *Recurrent Neural Network* (RNN), dirancang untuk menangani data sekuensial dan memiliki kemampuan dalam mengingat serta memahami konteks temporal antar elemen dalam urutan tersebut. Pada konteks ini, LSTM digunakan untuk menangkap pola temporal dalam video, seperti pergerakan wajah, perubahan ekspresi dan kedipan mata lainnya yang sering kali sulit ditiru secara konsisten oleh model generatif pembuat *deepfake*.

### 3.2 Alur Kerja Penelitian

Alur kerja dalam penelitian ini dapat dilihat pada gambar 3.1, yang menunjukkan tahapan-tahapan dalam proses deteksi *deepfake* menggunakan ResNet-50 dan LSTM.



Gambar 3.1. Alur Kerja Penelitian proses deteksi *deepfake*

### 1. Studi Literatur

Tahapan awal dalam penelitian ini adalah melakukan studi literatur yang mendalam untuk memahami konteks dan tantangan dalam deteksi video *deepfake*. Penelusuran literatur dilakukan terhadap jurnal-jurnal terkini, repositori model *deep learning*, serta dokumentasi teknis terkait arsitektur *Convolutional Neural Network* (CNN) seperti ResNet-50 dan *Recurrent Neural Network* (RNN) seperti *Long Short-Term Memory* (LSTM). Penelitian sebelumnya tentang deteksi manipulasi visual, teknik ekstraksi fitur spasial dan temporal, serta metode evaluasi seperti *precision*, *recall*, dan *F1-score* juga dikaji sebagai landasan teoritis. Studi literatur ini menjadi dasar dalam menyusun metodologi yang tepat, termasuk dalam pemilihan *dataset*, model, dan strategi pelatihan.

### 2. Pengumpulan *Dataset*

*Dataset* yang digunakan dalam penelitian ini adalah ForgeryNet, yang terdiri dari 6000 video yaitu 3000 video *deepfake* dan 3000 video *real*. Proses pengumpulan melibatkan unduhan data, verifikasi integritas *file*, dan dokumentasi metadata. Selanjutnya dilakukan *preprocessing* berupa ekstraksi *frame*, *resize* ke 224x224 piksel, normalisasi, serta augmentasi seperti rotasi dan *flipping*. *Dataset* kemudian dibagi menjadi data latih, validasi, dan uji dengan proporsi 70:20:10.

### 3. Perancangan Sistem

Tahapan ini merupakan inti dari proses penelitian, di mana sistem deteksi video *deepfake* dirancang dan diimplementasikan menggunakan pendekatan dua tahap: ekstraksi fitur spasial dan analisis fitur temporal. Ekstraksi fitur visual dilakukan menggunakan ResNet-50, yang telah terbukti efektif dalam mengenali pola visual pada data citra. Model ini juga memanfaatkan bobot *pre-trained* dari *ImageNet* untuk mempercepat *convergence*. Hasil dari ResNet-50 berupa vektor fitur dari setiap *frame*, kemudian diurutkan dan dimasukkan ke dalam jaringan *LSTM*, yang bertugas menangkap dinamika temporal antar-*frame*. Jaringan *LSTM* membantu mengenali pola manipulasi yang tersembunyi dalam pergerakan wajah atau ekspresi yang tidak konsisten. Hasil akhir dari *LSTM* diteruskan ke *fully connected layer* dan diklasifikasikan melalui fungsi aktivasi *softmax* untuk menghasilkan prediksi probabilistik kelas (*real* atau *deepfake*). Proses ini dilengkapi dengan eksplorasi *hyperparameter* seperti jumlah unit *LSTM*, panjang *sequence*, *dropout rate*, jenis *optimizer*, *learning rate*, dan *weight decay*, yang dilakukan melalui pendekatan *grid search* untuk memperoleh kombinasi parameter terbaik dalam pelatihan.

### 4. Evaluasi Sistem

Setelah model selesai dilatih, evaluasi dilakukan menggunakan data uji yang belum pernah digunakan selama proses pelatihan. Evaluasi bertujuan untuk mengukur sejauh mana model mampu mengenali video *deepfake* secara akurat dan konsisten. Metrik evaluasi yang digunakan meliputi *loss*, *precision*, *recall*, dan *F1-score* yang memberikan gambaran secara menyeluruh terhadap performa sistem, baik dari segi kemampuan mengenali video *deepfake* maupun menghindari kesalahan klasifikasi terhadap video *real*. Pengujian sistem juga dilakukan dengan memasukkan video baru dan *data augmentasi* tambahan guna mengukur kemampuan generalisasi model

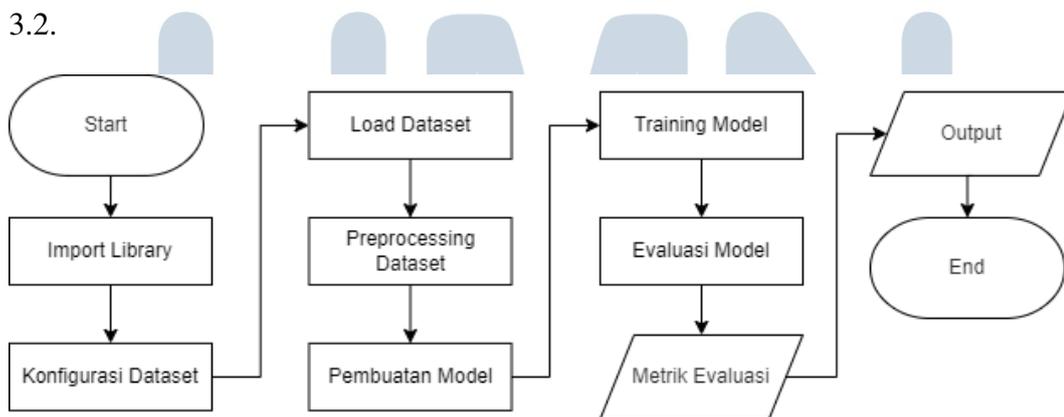
terhadap variasi data yang belum pernah dilihat sebelumnya. Hal ini penting untuk menjamin bahwa sistem tidak hanya menghafal pola pada data latih, tetapi juga mampu mendeteksi *deepfake* pada skenario nyata.

## 5. Penulisan Laporan

Tahap terakhir dalam alur kerja penelitian ini adalah penyusunan laporan secara sistematis. Laporan mencakup seluruh proses mulai dari latar belakang, studi literatur, pengumpulan dan pengolahan data, perancangan sistem, hasil eksperimen, hingga analisis performa model. Penulisan dilakukan berdasarkan struktur akademik yang baku, guna mendokumentasikan seluruh temuan dan metodologi secara transparan. Laporan ini tidak hanya berfungsi sebagai bentuk pertanggungjawaban ilmiah, tetapi juga sebagai referensi bagi penelitian lanjutan, termasuk untuk pengembangan sistem deteksi *deepfake* yang lebih adaptif dan efisien di masa mendatang.

### 3.3 Perancangan Sistem

Sistem yang dikembangkan dalam penelitian ini bertujuan untuk mendeteksi *deepfake* dalam video menggunakan algoritma ResNet-50 dan Long Short-Term Memory (LSTM). Rancangan sistem dalam penelitian ini dapat dilihat pada gambar 3.2.



Gambar 3.2. Alur eksekusi program deteksi *deepfake*

#### 1. *Import Library*

Langkah pertama dalam implementasi program adalah mengimpor berbagai pustaka (*library*) yang dibutuhkan. *Library* yang digunakan mencakup NumPy untuk manipulasi data numerik, Pandas untuk pengelolaan data

tabular, OpenCV untuk pemrosesan gambar dan video, serta Matplotlib dan Seaborn untuk visualisasi data. Selain itu, PyTorch dan Torchvision digunakan sebagai *framework* utama dalam membangun dan melatih model *deep learning*. Dengan mengimpor seluruh *library* ini di awal program, maka seluruh fungsi dan modul yang dibutuhkan akan tersedia sepanjang eksekusi berlangsung, sehingga proses pembuatan dan pelatihan model dapat berjalan lancar tanpa gangguan dependensi.

## 2. Konfigurasi *Dataset*

Setelah pustaka berhasil diimpor, langkah selanjutnya adalah mengatur konfigurasi *dataset* yang akan digunakan. Tahap ini sangat penting karena berkaitan langsung dengan cara data dibaca, diproses, dan disiapkan untuk dimasukkan ke dalam model. Peneliti menentukan direktori *dataset*, format *file*, struktur folder (misalnya folder *real* dan *fake*), serta parameter lainnya seperti ukuran gambar dan jumlah *channel* warna.

## 3. *Load Dataset*

Setelah konfigurasi dilakukan, tahap berikutnya adalah memuat *dataset* ke dalam program menggunakan *DataLoader*. *Dataset* video *deepfake* dan *real* dari ForgeryNet akan dimuat dan dikonversi menjadi representasi yang dapat diolah oleh model *deep learning*.

## 4. *Preprocessing Dataset*

Tahap *preprocessing dataset* bertujuan untuk mempersiapkan data agar sesuai dengan input yang diharapkan oleh model *deep learning*. Proses ini mencakup *resize* gambar ke dimensi tertentu, normalisasi nilai piksel ke rentang [0,1], dan transformasi lain yang sesuai dengan arsitektur model, dalam hal ini ResNet-50. Tidak hanya itu, dilakukan pula augmentasi data seperti rotasi, *flipping horizontal*, penyesuaian kecerahan, dan kontras untuk menambah variasi data latih. Augmentasi ini sangat penting untuk meningkatkan kemampuan generalisasi model dan mengurangi risiko *overfitting*. *Preprocessing* juga mencakup pelabelan data (*labeling*) sesuai dengan kelasnya, yaitu 0 untuk video *real* dan 1 untuk video *deepfake*. Semua proses ini dilakukan secara otomatis sebelum data dimasukkan ke model.

## 5. Pembuatan Model

Pembuatan model merupakan tahapan inti dari proses ini, di mana arsitektur model *deep learning* dirancang dan diimplementasikan. Dalam penelitian ini,

model yang digunakan terdiri dari dua komponen utama, yaitu ResNet-50 sebagai *feature extractor* dan LSTM sebagai *sequence analyzer*. ResNet-50 bertugas mengekstraksi fitur spasial dari setiap *frame* dalam video, seperti tekstur wajah dan pencahayaan. Hasil ekstraksi kemudian diteruskan ke LSTM yang akan menganalisis hubungan temporal antar *frame*, sehingga model dapat menangkap inkonsistensi ekspresi wajah dan gerakan mulut yang biasa terjadi pada video *deepfake*. Seluruh hasil dari LSTM akhirnya diproses oleh layer *fully connected* dan fungsi aktivasi *softmax* untuk menghasilkan *output* prediksi kelas.

#### 6. *Training Model*

Setelah model selesai dibuat, tahap berikutnya adalah melakukan pelatihan model menggunakan *dataset* yang telah diproses. *Training* dilakukan dengan menyesuaikan *hyperparameter* seperti *learning rate*, *batch size*, dan jumlah *epoch*. Proses ini melibatkan penggunaan teknik optimasi seperti RAdam serta fungsi *loss* seperti CrossEntropyLoss. Model akan belajar membedakan antara video asli dan video *deepfake* berdasarkan fitur yang diekstraksi dari *frame*. Pelatihan berlangsung dalam beberapa *epoch*, dan performa model dipantau secara berkala menggunakan data validasi. Selama proses ini, nilai *loss* dan akurasi dicatat untuk mengetahui sejauh mana model mengalami peningkatan performa.

#### 7. *Evaluasi Model*

Evaluasi model dilakukan setelah pelatihan selesai, menggunakan data uji yang belum pernah dilihat oleh model sebelumnya. Tujuannya adalah untuk mengukur kemampuan model dalam menggeneralisasi dan mendeteksi *deepfake* secara akurat. Evaluasi dilakukan dengan menghitung prediksi model terhadap video uji dan membandingkannya dengan label sebenarnya. Proses ini tidak hanya melihat akurasi semata, tetapi juga mencakup berbagai metrik penting lainnya seperti *precision*, *recall*, dan *F1-score*. Hasil evaluasi akan memberikan gambaran apakah model sudah cukup baik atau perlu dilakukan perbaikan lanjutan.

#### 8. *Metrik Evaluasi*

Pada tahap ini, hasil dari evaluasi model diolah lebih lanjut menjadi metrik evaluasi yang dapat digunakan untuk menganalisis performa secara kuantitatif. Metrik utama yang digunakan adalah *confusion matrix*, akurasi,

*precision*, *recall*, dan *F1-score*. *Confusion matrix* memberikan visualisasi distribusi kesalahan model dalam bentuk *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)*. *Precision* dan *recall* digunakan untuk menilai ketepatan dan sensitivitas model terhadap kelas *deepfake*. *F1-score* memberikan keseimbangan antara *precision* dan *recall*, dan sangat penting terutama jika distribusi kelas tidak seimbang. Hasil metrik ini menjadi dasar untuk menilai keefektifan model.

#### 9. *Output*

Tahap *output* adalah hasil akhir dari keseluruhan proses eksekusi program. Model yang telah dilatih dan dievaluasi akan menghasilkan prediksi berupa label untuk setiap video, disertai dengan nilai *confidence score*. Nilai ini menunjukkan seberapa yakin model terhadap hasil prediksinya. *Output* ini dapat digunakan untuk kebutuhan analisis lebih lanjut, visualisasi hasil, atau bahkan integrasi ke dalam sistem pendeteksi otomatis. Jika diperlukan, model juga dapat disimpan dalam format *.pt* agar dapat digunakan kembali tanpa perlu melakukan pelatihan ulang.

### 3.4 Teknik Evaluasi Kinerja

Evaluasi kinerja model dilakukan untuk memastikan kemampuan sistem dalam mendeteksi *deepfake* secara akurat. Beberapa metode yang digunakan dalam evaluasi ini meliputi:

#### 1. *Accuracy*

*Accuracy* mengukur proporsi keseluruhan video yang diklasifikasikan dengan benar, baik video *deepfake* maupun video asli, terhadap seluruh video dalam *dataset*. Metrik ini memberikan gambaran umum tentang kinerja model, khususnya ketika jumlah video *deepfake* dan asli relatif seimbang.

#### 2. *Loss*

*Loss* digunakan untuk mengukur kesalahan model selama proses pelatihan dan validasi. Nilai *loss* yang lebih rendah menunjukkan bahwa model semakin baik dalam membedakan antara video *deepfake* dan video asli. Penurunan *loss* secara konsisten merupakan indikator bahwa model mampu belajar fitur-fitur penting dengan efektif.

### 3. *Confusion Matrix*

Digunakan untuk menganalisis jumlah prediksi yang benar dan salah dalam mendeteksi video *deepfake* dan video asli. Matriks ini mencakup nilai *true positive* (TP), *false positive* (FP), *true negative* (TN), dan *false negative* (FN), yang memberikan gambaran mengenai efektivitas model dalam mengklasifikasikan video.

### 4. *Precision, Recall, dan F1-Score*

- *Precision*: Menunjukkan proporsi video yang benar-benar *deepfake* dari seluruh video yang terdeteksi sebagai *deepfake*.
- *Recall*: Mengukur seberapa banyak video *deepfake* yang berhasil diidentifikasi dibandingkan dengan jumlah total video *deepfake* yang ada dalam dataset.
- *F1-Score*: Merupakan rata-rata harmonik dari *precision* dan *recall*, yang digunakan untuk mendapatkan keseimbangan antara ketepatan dan cakupan dalam deteksi video *deepfake*.

## 3.5 Pengujian Sistem

Pengujian dilakukan untuk menilai kemampuan model dalam mendeteksi *deepfake* dari video yang belum pernah dianalisis sebelumnya. Beberapa metode yang digunakan dalam tahap pengujian ini meliputi:

### 1. Pengujian dengan Augmentasi Data

Model diuji menggunakan video yang telah dimodifikasi melalui teknik augmentasi, seperti perubahan pencahayaan, rotasi, skala, dan penyesuaian warna. Pengujian ini bertujuan untuk mengevaluasi ketahanan model terhadap variasi data yang mungkin terjadi di dunia nyata.

### 2. Uji Akurasi

Uji akurasi bertujuan untuk mengukur seberapa akurat model dalam melakukan deteksi dengan membandingkan jumlah hasil klasifikasi yang benar terhadap keseluruhan deteksi yang dilakukan.