

BAB III

METODE PENELITIAN

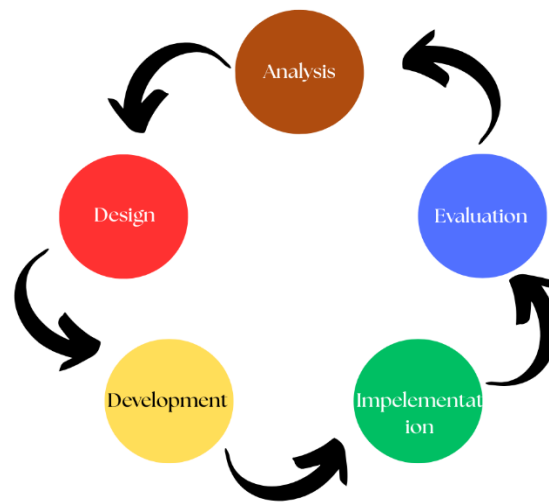
Bab ini memaparkan pendekatan penelitian dan tahapan-tahapan yang ditempuh dalam proses untuk penelitian ini. Tahapan dalam metode ini meliputi : Desain Penelitian, Alur Penelitian, Deskripsi Umum, Perancangan Sistem, Metode Pengujian Model, Metode Pengujian Web. Setiap tahapan disusun secara sistematis guna memastikan kesesuaian antara kerangka eksperimen, pemilihan arsitektur model, serta prosedur validasi untuk memperoleh hasil yang terukur dan dapat direproduksi.

3.1. Desain Penelitian

Jenis penelitian ini adalah *Research and Development* (R&D) dengan pendekatan kuantitatif. Pendekatan ini bertujuan untuk menghasilkan suatu produk dan menguji tingkat keefektifannya melalui tahapan yang sistematis dan terukur. Menurut Sugiyono (2019, hlm. 297), penelitian dan pengembangan merupakan metode penelitian yang digunakan untuk menghasilkan produk tertentu serta menguji keefektifan produk tersebut. Dalam penelitian ini, model pengembangan yang digunakan mengacu pada desain instruksional ADDIE, yang terdiri dari lima tahap utama, yaitu *Analysis*, *Design*, *Development*, *Implementation*, dan *Evaluation*.

Dalam penelitian ini, produk yang dikembangkan adalah sistem deteksi video *deepfake* berbasis arsitektur DL dan *optical flow*, yang dirancang untuk bekerja secara efisien dalam lingkungan terbatas seperti aplikasi web. Penelitian ini tidak hanya menganalisis performa model deteksi, tetapi juga menghasilkan prototipe sistem yang dapat diuji, baik dari segi akurasi maupun efisiensi implementasi.

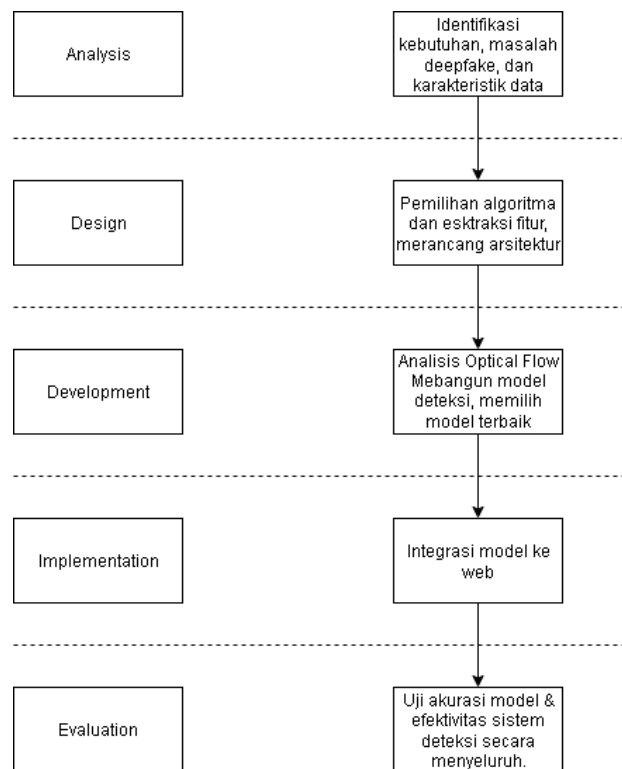
Model ADDIE dipilih karena sesuai dengan tujuan penelitian ini, yaitu mengembangkan suatu produk berbasis teknologi yang dapat diuji kelayakannya dalam konteks praktis. Dan juga pendekatan R&D dengan model ADDIE memberikan alur pengembangan yang sistematis, mulai dari analisis kebutuhan, perancangan sistem, pengembangan model, implementasi dalam skenario terbatas, hingga evaluasi performa seperti pada **Gambar 3.1**.



Gambar 3. 1 Model ADDIE

3.2. Alur Penelitian

Model pengembangan yang digunakan dalam penelitian ini adalah model ADDIE, yang merupakan singkatan dari *Analysis*, *Design*, *Development*, *Implementation*, dan *Evaluation*. Model ini dikembangkan oleh Dick dan Carey dan merupakan salah satu pendekatan sistematis yang banyak digunakan dalam pengembangan sistem pembelajaran dan teknologi informasi (Omoregie dkk., 2025). Model ADDIE dinilai sesuai karena memberikan struktur pengembangan yang iteratif dan fleksibel, serta memungkinkan evaluasi dilakukan pada setiap tahap untuk memastikan efektivitas dan kelayakan produk seperti pada **Gambar 3.2**.



Gambar 3. 2 Alur Penelitian

Dalam penelitian ini, model ADDIE digunakan sebagai kerangka dalam merancang dan mengembangkan sistem deteksi video *deepfake* berbasis DL dan *optical flow* yang dirancang untuk dapat berjalan secara efisien pada perangkat dengan sumber daya terbatas. Penjabaran dari masing-masing tahap dalam model ADDIE adalah sebagai berikut:

3.2.1. Analysis

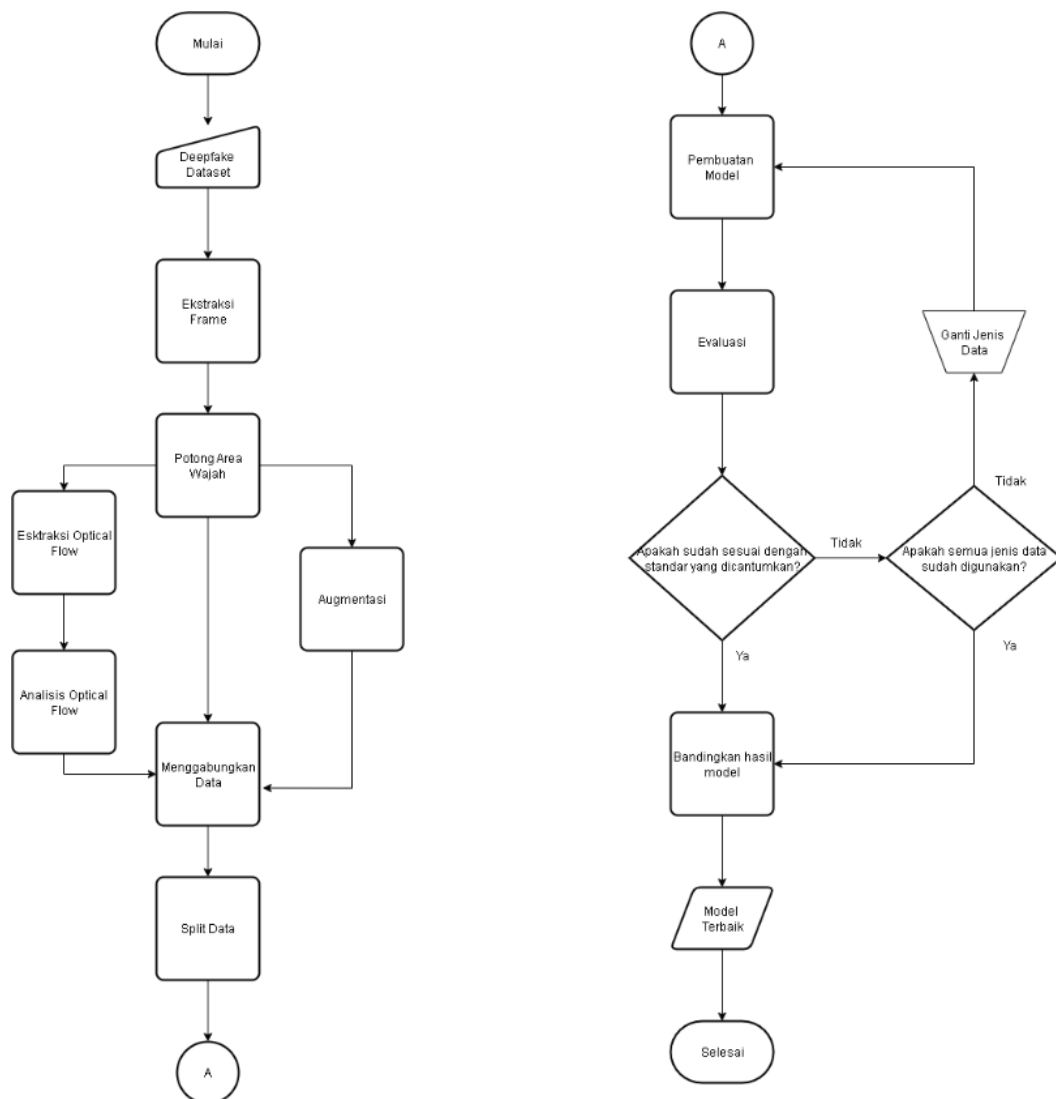
Tahap analisis merupakan tahap awal dalam proses pengembangan sistem. Pada tahap ini dilakukan identifikasi terhadap kebutuhan peneliti, masalah yang ada terkait deteksi video *deepfake*, serta analisis terhadap teknologi dan metode yang relevan. Peneliti melakukan kajian pustaka terhadap arsitektur DL, teknik *optical flow*, serta sistem deteksi yang telah dikembangkan sebelumnya. Selain itu, dilakukan pula analisis terhadap karakteristik *dataset* yang akan digunakan, yaitu *subset* FaceForensics++. Tujuan utama dari tahap ini adalah memahami kebutuhan fungsional dan teknis yang harus dipenuhi oleh sistem deteksi yang akan dikembangkan.

3.2.2. Design

Tahap desain dalam penelitian ini bertujuan untuk merancang alur pengembangan sistem deteksi *deepfake* berbasis web. Perancangan difokuskan pada pengembangan model DL, termasuk arsitektur jaringan, metode pelatihan, dan integrasi dengan proses *optical flow* untuk meningkatkan akurasi deteksi. Sistem dikembangkan secara modular menggunakan Streamlit, di mana seluruh komponen mulai dari upload video, ekstraksi *frame*, pemrosesan *optical flow* menggunakan Farneback, hingga klasifikasi dengan model DL dijalankan secara *real-time*. Tampilan antarmuka dirancang sederhana dan intuitif, menampilkan hasil *crop* wajah, visualisasi *optical flow*, dan prediksi klasifikasi.

3.2.3. Development

Tahap pengembangan merupakan inti dari proses realisasi sistem deteksi *deepfake* berbasis web. Tahapan ini berfokus pada pembangunan *pipeline* lengkap mulai dari pemrosesan data hingga pelatihan dan integrasi model ke dalam sistem web.



Gambar 3. 3 Flowchart Pembuatan Model

Adapun langkah-langkah pengembangan sistem ini seperti pada Gambar 3.3 dijelaskan sebagai berikut:

1) Pengolahan *Dataset*

Penelitian ini menggunakan *dataset benchmark* FaceForensics++, yang terdiri dari video asli dan hasil manipulasi menggunakan metode *deepfakes*, FaceSwap, Face2Face, dan NeuralTextures. *Dataset* ini dipilih karena memiliki struktur yang baik dan mencerminkan berbagai skenario manipulasi wajah di dunia nyata.

2) *Preprocessing Data*

Setiap video pada *dataset* diekstrak menjadi 50 *frame*. Setelah itu, dilakukan pemotongan (*cropping*) pada area wajah menggunakan deteksi *bounding box* cascade agar hanya bagian relevan yang digunakan sebagai input.

3) Analisis *Optical Flow*

Optical Flow dihitung menggunakan algoritma Farneback untuk mendapatkan representasi pergerakan piksel antar *frame*. Teknik ini digunakan untuk menangkap ketidaksesuaian gerakan mikro yang menjadi indikasi utama adanya manipulasi subpiksel pada *deepfake*.

4) Augmentasi Data

Untuk meningkatkan generalisasi model, dilakukan augmentasi terbatas pada *frame* RGB, seperti rotasi, *flipping*, dan variasi kontras. Augmentasi ini diterapkan secara selektif pada skenario *input* tertentu.

5) Penggabungan *Data Input*

Sistem dikembangkan untuk mengakomodasi tiga jenis *input*:

- Hanya *Optical Flow*
- *Optical Flow* + RGB Frame
- *Optical Flow* + RGB Frame + Augmentasi Frame

Setiap skenario disiapkan dalam *pipeline* berbeda agar dapat digunakan sebagai eksperimen terpisah dalam pelatihan model. Tujuan penggabungan model adalah untuk memahami batasan seberapa jauh model bisa membaca data berdasarkan *input optical flow*.

6) *Split Dataset*

Data dibagi menjadi data pelatihan, data validasi dan data pengujian dengan rasio tertentu (70:20:10), menggunakan teknik *traintestsplit* agar distribusi label tetap seimbang.

7) Pelatihan Model *Deep Learning*

Model pelatihan yang digunakan meliputi :

- CNN : MobileNetV3, EfficientNetv2
- *Transformers* : ViViT, MobileViT

Pelatihan dilakukan di Google Colab menggunakan GPU (tipe L4) dan *library* TensorFlow serta PyTorch dengan *epoch* sebanyak 50. Model dilatih untuk mendeteksi klasifikasi *binary (real vs fake)* berdasarkan representasi *optical flow* dan RGB.

Dalam pelatihatannya, ditambahkan juga paramter tambahan seperti ReduceLROnPlateau dan earlystop. ReduceLROnPlateau ditambahkan untuk menurunkan learning rate apabila nilai loss tidak turun pada tiap *epoch* nya, lalu untuk earlystop digunakan untuk menghentikan pelatihan model apabila nilai *loss* nya tidak turun dalam kurun *epoch* tertentu (dalam pelatihan ini digunakan batas 5 *epoch*).

8) Perbandingan Model

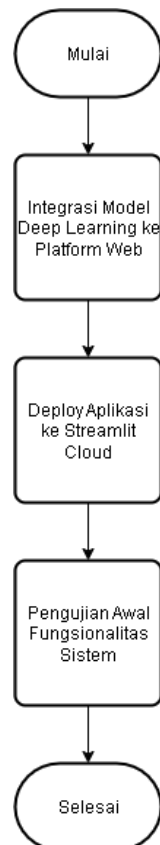
Setiap model dievaluasi menggunakan metrik akurasi dan *loss* pada pelatihan dan validasi. Hasil evaluasi dibandingkan untuk menentukan model terbaik berdasarkan keseimbangan antara performa deteksi.

Selain membangun sistem berbasis DL, penelitian ini juga memasukkan analisis deskriptif dan analisis statistik sebagai bagian dari tahapan pengembangan. Analisis deskriptif dilakukan untuk memahami karakteristik awal *dataset*, misalnya distribusi jumlah video per kelas (*real* dan *fake*), variasi panjang durasi video, serta persebaran nilai hasil ekstraksi fitur. Analisis ini membantu memberikan gambaran umum mengenai struktur data yang digunakan sebelum dimasukkan ke tahap pemodelan.

Sementara itu, analisis statistik dilakukan untuk memperkuat pemahaman terhadap pola pada data. Penggunaan metrik seperti nilai rata-rata, standar deviasi, hingga korelasi antarfitur membantu dalam mengidentifikasi potensi bias, redundansi, maupun variabel penting yang berpengaruh terhadap hasil deteksi.

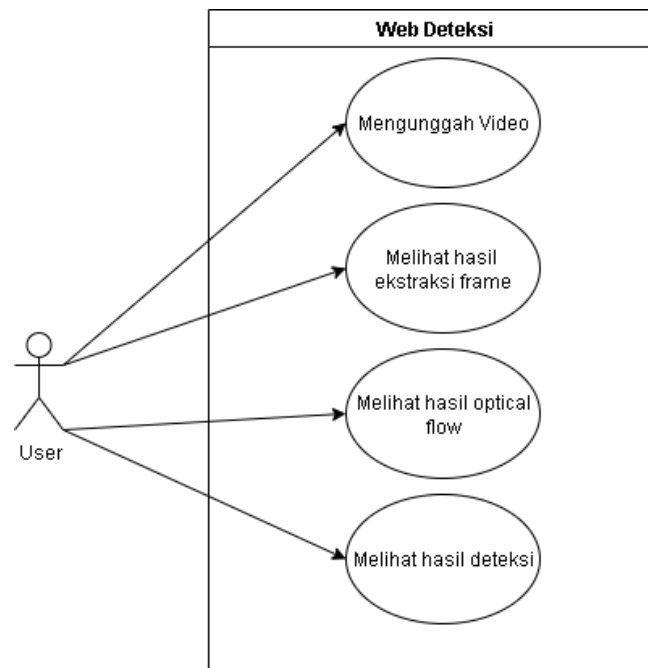
Dengan demikian, kombinasi analisis deskriptif dan statistik menjadi langkah awal yang penting dalam memastikan kualitas data sekaligus mendukung rancangan model yang lebih terarah.

3.2.4. *Impelementation*



Gambar 3. 4 Flowchart Implementasi

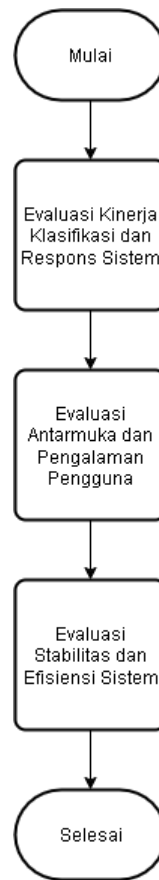
Tahap implementasi dilakukan dengan menerapkan sistem deteksi *deepfake* yang telah dikembangkan ke dalam platform web berbasis Streamlit. Pada tahap ini, model DL terbaik hasil pelatihan sebelumnya diintegrasikan secara langsung ke dalam alur aplikasi, yang memungkinkan proses unggah video, ekstraksi *frame*, pemrosesan *optical flow*, serta klasifikasi kelas. Aplikasi dihosting secara daring melalui layanan Streamlit Cloud agar dapat diakses tanpa instalasi lokal. Pengujian awal dilakukan untuk memastikan bahwa seluruh fungsi berjalan dengan baik, termasuk pemrosesan *input* video, visualisasi *frame* wajah dan hasil *optical flow*, serta keluaran prediksi model.



Gambar 3. 5 Diagram usecase web

Berdasarkan diagram *use case* pada Gambar 3.5, sistem menyediakan empat fungsionalitas utama bagi pengguna, yaitu: (1) mengunggah video yang akan dianalisis, (2) menampilkan hasil ekstraksi *frame* dari video, (3) menampilkan hasil pemrosesan *optical flow* untuk memvisualisasikan pergerakan pada area wajah, dan (4) menampilkan hasil akhir proses deteksi *deepfake*. Seluruh fungsionalitas ini disusun secara berurutan sehingga pengguna dapat mengikuti setiap tahap proses deteksi mulai dari pemasukan data hingga memperoleh hasil akhir dengan mudah dan terstruktur.

3.2.5. Evaluation



Gambar 3. 6 Flowchart Evaluasi

Tahapan evaluasi difokuskan untuk menilai kinerja sistem deteksi *deepfake* secara menyeluruh seperti pada Gambar 3.6, terutama pada aspek integrasi antara model DL dan aplikasi web yang telah dikembangkan. Evaluasi dilakukan melalui serangkaian uji fungsional dan pengukuran performa sistem, meliputi ketepatan hasil klasifikasi terhadap video *deepfake*, kecepatan respons sistem dalam memproses *input* secara *real-time*, serta kemudahan penggunaan antarmuka oleh pengguna. Selain itu, dievaluasi pula stabilitas dan efisiensi sistem saat dijalankan di lingkungan *hosting* (Streamlit Cloud), guna memastikan bahwa sistem dapat digunakan secara konsisten tanpa gangguan. Hasil evaluasi ini menjadi dasar dalam menentukan keberhasilan sistem dan potensi pengembangan lebih lanjut.

Sebagai bagian dari proses evaluasi tersebut, dilakukan juga pengumpulan penilaian dari seorang ahli melalui pengisian kuesioner **Tabel 3.1** hingga **Tabel 3.2**. Melalui kuesioner ini, para ahli diminta memberikan evaluasi menyeluruh terhadap sistem, dengan mempertimbangkan berbagai aspek seperti keakuratan teknis, performa dan responsivitas sistem, struktur arsitektur, kualitas antarmuka pengguna. Setiap pernyataan dalam kuesioner disertai pilihan penilaian, dan responden diminta memberikan tanda centang (✓) pada kolom yang paling sesuai dengan pandangan profesional mereka. Di bagian akhir, tersedia pula ruang untuk menyampaikan saran atau masukan terbuka guna mendukung pengembangan sistem lebih lanjut. Penilaian diharapkan dilakukan secara objektif berdasarkan kompetensi dan pengalaman teknis yang dimiliki.

- Bagian A –Akurasi dan Performa Sistem

Tabel 3. 1 Bagian A Kuesioner

Indikator	Sangat Rendah	Rendah	Cukup	Baik	Sangat Baik
Soal					
Bagaimana Anda menilai akurasi sistem dalam mendeteksi video <i>deepfake</i> berdasarkan hasil pengujian					
Apakah proses klasifikasi dan deteksi dilakukan secara efisien dalam konteks sistem <i>real-time</i> ?					
Apakah integrasi antara model DL dan <i>pipeline</i> pemrosesan video (<i>Optical Flow, frame extraction</i> , klasifikasi) sudah berjalan optimal?					

- Bagian B – UI/UX

Tabel 3. 2 Bagian B Kuesioner

Indikator	Sangat Rendah	Rendah	Cukup	Baik	Sangat Baik
Soal					
Bagaimana Anda menilai pemilihan pemilihan arsitektur DL dan metode <i>Optical Flow</i> (Farneback) yang digunakan dalam sistem ini?					
Apakah pendekatan modular dengan Streamlit dalam pengembangan sistem dapat diterima dalam konteks produksi atau pengembangan sistem serupa?					
Bagaimana Anda menilai skalabilitas dan potensi pengembangan sistem ke skala produksi (misalnya untuk penggunaan massal atau institusional)?					

- Bagian C – Stabilitas dan Efisiensi Sistem

Tabel 3. 3 Bagian C Kuesioner

Indikator	Sangat Rendah	Rendah	Cukup	Baik	Sangat Baik
Soal					
Apakah platform <i>hosting</i> (Streamlit Cloud) yang digunakan cukup andal dan stabil untuk pengujian sistem?					
Bagaimana pendapat Anda mengenai efisiensi sistem dalam hal pemrosesan dan penggunaan sumber daya (CPU, RAM, <i>bandwidth</i>)?					