

BAB III METODE PENELITIAN

3.1 Pendekatan Penelitian

Pendekatan penelitian yang digunakan dalam membuat skripsi berjudul "Implementasi *Face Recognition* Dan *Passive-Active Liveness Detection* Berbasis Insightface, Yolo11n, Dan Mediapipe Pada Sistem Presensi Online" adalah pendekatan kuantitatif. Pendekatan ini dipilih karena penelitian ini bertujuan menguji kemampuan algoritma pengenalan wajah dan menganalisis data numerik agar bisa menghasilkan hasil validasi kehadiran yang tepat, aman, dan tidak memihak. Dengan pendekatan ini, hasil penelitian bisa dinilai secara statistik berdasarkan kemampuan model kecerdasan buatan dalam mendeteksi dan metrik pengujian sistem.

Penelitian ini mengusulkan verifikasi biometrik dua lapis (*two-stage cascaded liveness verification*). Pada lapis pertama, algoritma YOLO11n digunakan sebagai bagian utama untuk mendeteksi kehidupan pasif (*anti-spoofing*) dengan tujuan membedakan wajah asli dari wajah palsu, seperti foto cetak atau gambar dari layar digital. Di lapisan kedua, fitur MediaPipe Face Landmarker digunakan di sisi klien untuk melakukan deteksi *active liveness* berbasis metode *challenge-response*, misalnya mengenali gerakan arah kepala atau ekspresi wajah. Selanjutnya, pustaka InsightFace digunakan untuk mengekstrak fitur wajah dan mengubahnya menjadi representasi berupa vektor numerik (*face embeddings*). Proses mengenali identitas pengguna kemudian dihitung dengan algoritma *Cosine Similarity* yang membandingkan sudut antara vektor fitur wajah pengguna saat absensi dengan data wajah yang tersimpan di dalam database PostgreSQL.

Penilaian terhadap hasil prediksi yang dihasilkan oleh model kecerdasan buatan dilakukan dengan memakai beberapa metrik statistik seperti akurasi, presisi, *recall*, dan *F1-score* agar pengevaluasian kualitas model klasifikasi menjadi lebih tepat. Seluruh data dari uji coba dan interaksi langsung dianalisis agar bisa memberikan informasi yang jelas dalam bentuk angka tentang seberapa efektif algoritma dalam mencegah kecurangan (*spoofing*) dan seberapa kokoh arsitektur API Flask dalam memproses data kehadiran secara cepat dan tepat.

Sementara itu, metode pembuatan perangkat lunak digunakan dalam merancang struktur sistem yang terdistribusi, yaitu *microservices*. Komunikasi

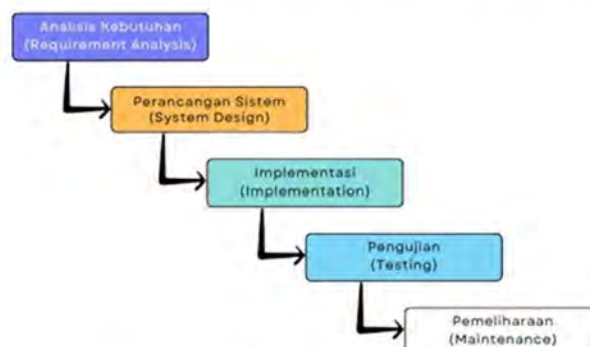
asinkron antara antarmuka klien (React JS), server inferensi kecerdasan buatan (Flask API), dan sistem backend logika bisnis (Laravel) diukur performanya menggunakan statistik deskriptif berupa rata-rata waktu yang dibutuhkan untuk merespons (Response Time) dalam satuan milidetik (ms).

3.2 Desain Penelitian

Penelitian ini menggunakan pendekatan kuantitatif deskriptif dan eksperimental dalam merancang penelitian. Pendekatan ini dipilih karena penelitian bertujuan membuat, menerapkan, dan mengevaluasi kemampuan sistem biometrik dalam memvalidasi kehadiran berdasarkan parameter visual yang terukur.

Penelitian ini melakukan eksperimen untuk mengecek sejauh mana model mampu bertahan terhadap berbagai jenis serangan palsu, seperti menggunakan foto berupa kertas atau video ulang. Analisis dilakukan dengan metode statistik deskriptif untuk mengevaluasi bagaimana algoritma YOLO11n dan perhitungan *Cosine Similarity* bekerja dalam memproses frame gambar yang diterima secara langsung dari antarmuka React JS.

Dalam membuat sistem Presensi ini, digunakan pendekatan pengembangan perangkat lunak dengan metode SDLC (Software Development Life Cycle) Waterfall. Metode Waterfall dipilih karena sistem yang dikembangkan memiliki desain arsitektur microservices yang jelas dan terorganisir dari awal, mulai dari bagian frontend kamera, backend menggunakan Laravel, hingga API dengan Flask. Setiap langkah harus dikerjakan dengan baik dan lengkap sebelum lanjut ke tahap berikutnya agar kesalahan dalam menggabungkan bagian-bagian bisa dihindari di akhir proses. Berikut adalah tahapan metode Waterfall yang diterapkan:



Gambar 3.1 Metode Waterfall

1. *Requirement Analysis and Definition* (Analisis Kebutuhan)

Di tahap ini, dilakukan pengecekan kebutuhan fungsional dan non fungsional dari sistem presensi. Kebutuhan utama adalah menyediakan antarmuka kamera berbasis web yang ringan, layanan *backend* untuk mengelola data pengguna seperti mahasiswa magang dan *freelance*, serta proses pemodelan Machine Learning yang dioptimalkan menggunakan ONNX Runtime dengan provider eksekusi CPU agar tidak memberatkan memori server.

2. *System and Software Design* (Perancangan Sistem)

Tahap ini mencakup perencanaan secara menyeluruh tentang struktur sistem, mencakup Entity Relationship Diagram (ERD) menggunakan PostgreSQL, pembuatan desain *User Interface* (UI), serta pengaturan model komunikasi alur API RESTful berbasis *stateless* antara React JS, Laravel, dan Flask.

3. *Implementation and Unit Testing* (Implementasi)

Desain perangkat lunak diubah menjadi barisan kode program. Antarmuka kamera dan modul MediaPipe yang berbasis WebAssembly dibangun dengan menggunakan React JS, manajemen data dan aturan bisnis dibangun dengan *framework* Laravel, sedangkan modul AI untuk mendeteksi wajah seperti YOLO11n dan InsightFace dijalankan dengan menggunakan Microframework Flask berbasis Python. Pengujian pada setiap unit dilakukan secara terpisah untuk masing-masing modul.

4. *Integration and System Testing* (Pengujian Sistem)

Pengujian dilakukan dengan menggabungkan semua komponen sistem secara utuh. Model diuji dengan menggunakan *Confusion Matrix*, sedangkan sistem web secara keseluruhan diuji dengan metode *Black Box Testing* untuk memastikan fungsi presensi Masuk, Pulang, dan Registrasi 15 sampel wajah berjalan dengan baik. Selain itu, *User Acceptance Testing* (UAT) dilakukan untuk mengukur tingkat penerimaan pengguna, dan pengujian latensi diterapkan guna memastikan transmisi *real-time* beroperasi tanpa hambatan.

5. *Operation and Maintenance* (Pemeliharaan)

Setelah aplikasi diuji dan terbukti stabil, sistem kemudian dijalankan (*deployment*) untuk digunakan oleh mahasiswa magang di Universitas Mercu Buana. Tahap ini melibatkan memantau sistem, memperbaiki masalah *bug*, serta mengoptimalkan penggunaan RAM jika terjadi gangguan pada memori cache Pandas DataFrame.

3.3 Subject Penelitian

Subjek penelitian dalam penelitian ini adalah penggunaan model Machine Learning (YOLO11n dan InsightFace), pustaka visi komputer MediaPipe, serta arsitektur microframework Flask untuk membuat sistem presensi yang aman dan tidak bisa dimanipulasi. Pendekatan yang digunakan bersifat kuantitatif dan deskriptif, dengan tujuan untuk menunjukkan secara terukur bagaimana frame gambar pengguna diproses agar dapat menghasilkan keputusan autentikasi.

Secara teknis, subjek penelitian terdiri dari tiga komponen utama. Pertama, modul *liveness* detection menggabungkan *passive liveness* yaitu model hasil pelatihan YOLO11n dengan format ONNX yang dirancang untuk mendeteksi wajah *real* (asli) dan *fake* (palsu) secara *real-time*, dan *active liveness* yaitu MediaPipe Face Landmarker untuk memverifikasi interaksi pengguna secara langsung di sisi klien.

Kedua, modul pengenalan identitas menggunakan model pengenalan wajah *buffalo_sc* dari pustaka Insightface untuk mengekstraksi wajah yang tervalidasi *real* menjadi *array embedding* berdimensi tinggi. Dan algoritma *cosine similarity* yang digunakan dengan *threshold* akurasi ≥ 0.5 untuk mencocokkan wajah dengan database.

Ketiga, arsitektur sistem web dengan laravel dan react, bertujuan mengelola antarmuka pengguna berbasis komponen (DOM), mengaktifkan kamera dengan fitur *getUserMedia*, manajemen sesi presensi, serta mengintegrasikan endpoint (API) untuk komunikasi antar layanan.

Fokus utama penelitian ini adalah mengukur ketahanan arsitektur dua lapis dalam menekan angka kecurangan presensi, serta mengevaluasi kestabilan sistem dalam mengelola aliran data video yang dikirim terus-menerus menggunakan format Base64.

3.4 Instrumen Penelitian

Dalam penelitian ini, berbagai instrumen digunakan untuk memastikan proses berjalan lancar, mulai dari mengumpulkan data, merancang struktur sistem, melatih model *object detection*, hingga tahap peluncuran aplikasi yang siap digunakan. Instrumen ini dipilih karena sesuai dengan struktur teknologi kecerdasan buatan terbaru. Instrumen penelitian dibagi menjadi dua jenis utama, yaitu perangkat keras dan perangkat lunak.

Penggunaan instrumen perangkat lunak sangat penting dalam penelitian bidang rekayasa perangkat lunak ini. Python digunakan sebagai bahasa untuk memproses matriks dan pembelajaran mesin, sedangkan PHP dan JavaScript mengatur aliran trafik jaringan. Platform Ultralytics HUB digunakan sebagai instrumen berbasis awan untuk melatih bobot model, kemudian bobot tersebut diekspor ke dalam format ONNX agar model dapat dijalankan dengan sangat ringan. Seluruh detail instrumen perangkat lunak dapat dilihat pada Tabel 3.1.

Tabel 3.1 Instrumen Perangkat Lunak

Kategori	Nama Perangkat Lunak / Teknologi	Keterangan / Fungsi
Sitem Operasi (<i>OS Lokal</i>)	Windows 11 (64-bit)	Lingkungan sistem operasi utama untuk pengembangan aplikasi (<i>development</i>)
Sistem Operasi (<i>OS Server</i>)	Linux Ubuntu 24.04.4 LTS	Lingkungan sistem operasi produksi pada server kampus untuk tahap <i>deployment</i> sistem
Bahasa Pemrograman	PHP (v8.2+), Python (v3.12.3), JavaScript	Bahasa Pemrograman yang digunakan dalam pengembangan logika web, <i>machine learning</i> , dan reaktivitas antarmuka

Framework Web	Laravel (v11.0), React JS (v19.2)	Laravel bertugas sebagai manajemen data (logika bisnis), sedangkan untuk React JS bertugas sebagai antarmuka pemindaian kamera secara asinkron
Microframework AI	Flask	Menyediakan REST API (<i>endpoint</i> /recognize) untuk berkomunikasi antara model dengan aplikasi web
Pustaka AI / ML	MediaPipe Face Landmarker, Ultralytics, ONNXRuntime, InsightFace (buffalo_sc), OpenCV, Scikit-learn	Komponen pengolahan citra (Computer Vision) dalam mendeteksi wajah anti spoofing menggunakan model hasil pelatihan YOLO11n, ekstraksi <i>embedding</i> wajah dengan model buffalo_sc dari Pustaka Insightface, dan kalkulasi <i>Cosine Similarity</i>
Sistem Basis Data	PostgreSQL	<i>Database Management System</i> (DBMS) utama dalam penyimpanan data relasional dan vector wajah peserta
Alat Manajemen DB	DBeaver, pgAdmin 4	Antarmuka grafis (<i>GUI</i>) klien dalam mengelola, membuat query, dan

		memantau table di PostgreSQL
Cloud AI Platform	Ultralytics HUB	Platform berbasis web untuk proses pelatihan <i>dataset</i> dengan model YOLO11n dan export model ke format ONNX
Web Server (Lokal)	Built-in PHP, Node.js (Vite), Werkzeug (Flask) – Lokal Nginx / Apache & Gunicorn - Produksi	Server eksekusi program di lingkungan local dan lingkungan produksi server kampus
IDE / Alat Klien	VS Code, PyCharm	VS Code digunakan untuk penulisan kode web (PHP/JS), sedangkan PyCharm khusus untuk skrip AI (Python).

Selain perangkat lunak, diperlukan spesifikasi perangkat keras yang cukup baik agar proses pembuatan kecerdasan buatan bisa berjalan lancar tanpa hambatan. Perangkat keras digunakan di berbagai lingkungan, mulai dari tahap pengembangan hingga ditempatkan di server fisik di kampus.

Tabel 3.2 Instrumen Perangkat Keras

Lingkungan Komputasi	Komponen	Spesifikasi / Keterangan
Pelatihan Model	Mesin Komputasi	Ultralytics Cloud Infrastructure
	Akselerator (GPU)	Virtual Dedicated GPU
Server Pengembangan Lokal (<i>Development</i>)	Perangkat Dasar	Laptop MSI Bravo 15 C7V
	Prosesor (CPU)	AMD Ryzen 7

	Memori (RAM)	16 GB
	Kartu Grafis (GPU)	NVIDIA GeForce RTX 4050 6GB GDDR6.
	Penyimpanan	512 GB SSD
Server Produksi (Deployment)	Perangkat Dasar	Virtual Machine (VM) yang dialokasikan di atas infrastruktur <i>Bare Metal Server</i> Universitas Mercu Buana.
	Prosesor (vCPU)	2 Core (QEMU Virtual CPU @ 2.0GHz).
	Memori (RAM)	4 GB
	Penyimpanan	30 GB

3.5 Teknik Pengumpulan Data

Sebelum membangun layanan kecerdasan buatan, dirancang jalur pemrosesan data untuk melatih model *object detection* Anti Spoofing menggunakan YOLO11n. Pengumpulan data dilakukan melalui penggabungan data primer dan data skunder, penggabungan ini berguna untuk meningkatkan variabilitas dan ketahanan model.

Metode pertama adalah Pengumpulan Data Primer, di mana peneliti mengambil citra langsung dari 7 partisipan. Pengambilan sampel wajah asli (*real*) dilakukan secara dinamis melalui variasi pencahayaan dan jarak dari depan *webcam*, menghasilkan 2.000 citra. Sementara itu, untuk menyimulasikan kejahatan digital, peneliti mengumpulkan 1.440 citra wajah palsu (*fake*) dengan cara menghadapkan foto hasil cetak kertas dan pantulan layar gawai (monitor/HP) ke arah kamera.

Metode kedua Pengumpulan Data Skunder: Menggunakan *dataset* public *Anti Spoofing Real Dataset* dari Kaggle serta *face-anti-spoofing-dataset*, penambahan data skunder ini digunakan supaya model hasil pelatihan bisa meningkatkan variabilitas dan ketahanan model dalam mengenali object asli dan palsu. Karena format data sekunder ini berupa rekaman video berkelanjutan, peneliti menyusun skrip khusus untuk mengekstraksi bingkai video tersebut menjadi gambar

berformat statis (JPEG), yang menghasilkan tambahan 800 citra asli dan 1.360 citra palsu.

Seluruh koleksi gambar dari sumber primer dan sumber sekunder digabungkan, sehingga terbentuk sebuah *dataset* yang lengkap berisi 2.800 gambar dengan kelas asli (1) dan 2.800 gambar dengan kelas palsu (0). Data tersebut lalu diberi label menggunakan format *bounding box* standar YOLO dan dibagi secara acak dengan distribusi rasio standar AI: 70% untuk pelatihan, 20% untuk validasi, dan 10% untuk pengujian.

3.6 Teknik Analisis Data

Teknis Analisis data dilakukan dengan cara kuantitatif komparatif dan deskriptif untuk mengecek parameter keberhasilan sistem tersebut. Analisis diajukan dalam dua bidang, yaitu prestasi kecerdasan buatan dan prestasi perangkat lunak.

1. Analisis Kinerja Model YOLO11n (*AI Evaluation*)

Kinerja deteksi *liveness* dinilai dengan menggunakan instrumen *Confusion Matrix*. Evaluasi ini membandingkan hasil prediksi model dengan label sebenarnya pada 10% data uji, dengan mencakup angka *True Positive*, *True Negative*, *False Positive*, dan *False Negative*. Hasil matriks tersebut selanjutnya dihitung secara matematis untuk memperoleh nilai Akurasi, Presisi, Recall, serta mean Average Precision (mAP50).

2. Analisis Komunikasi dan Fungsionalitas (*System Evaluation*)

Data waktu respons (latensi) yang terkumpul selama komunikasi asinkron dianalisis secara deskriptif untuk mencari nilai rata-rata (*Mean*). Analisis ini bertujuan menunjukkan bahwa metode *caching* menggunakan memori (Pandas DataFrame) dalam Flask API dapat menjaga kecepatan sistem tetap stabil dalam rentang waktu respons yang ditargetkan, meskipun terus-menerus menerima permintaan berupa serangan *looping* dari klien. Sementara itu, logika bisnis seperti penolakan absensi ketika ID wajah tidak sesuai diperiksa menggunakan pengujian skenario yang terstruktur (*Black Box Testing*). Selanjutnya, tingkat kelayakan antarmuka sistem bagi

pengguna akhir dievaluasi menggunakan kuesioner *User Acceptance Testing* (UAT).

3. Analisis *User Acceptance Testing* (UAT)

Evaluasi akhir dilakukan untuk menilai seberapa baik antarmuka sistem diterima dan digunakan oleh pengguna akhir. Pengujian ini mencakup distribusi kuesioner tertutup kepada responden yang mewakili subjek operasional. Hasil pengujian *User Acceptance Testing* (UAT) dianalisis dengan menghitung persentase kelayakan menggunakan persamaan matematika (3.1) berikut ini (Annisa et al., n.d.):

$$\text{Persentase} = \frac{\text{skor yang diperoleh}}{\text{skor maksimal}} \times 100\% \quad 3.1$$

Setelah persentase kelayakan diperoleh, angka tersebut diinterpretasikan ke dalam kriteria kualitatif. Mengacu pada standar pengukuran skala Likert yang diadaptasi dari penelitian (Annisa et al., n.d.), kriteria kelayakan sistem dibagi ke dalam lima rentang persentase sebagaimana dijabarkan pada Tabel 3.3.

Tabel 3.3 Kriteria Kelayakan *User Acceptance Testing*

Persentase	Kriteria Kelayakan
0% – 20%	Sangat Tidak Layak
21% – 40%	Tidak Layak
41% – 60%	Cukup Layak
61% – 80%	Layak
81% – 100%	Sangat Layak

3.7 Prosedur Penelitian

Prosedur penelitian ini disusun untuk merinci langkah-langkah kerja praktis dan operasional dari implementasi kerangka konseptual *Waterfall*. Penjabaran aktivitas ini menggambarkan perjalanan teknis penelitian sejak dari pengumpulan informasi hingga sistem siap digunakan oleh khalayak akhir.

1. Fase Analisis Kebutuhan: Memulai penelitian dengan melakukan wawancara langsung bersama Biro Sumber Daya Manusia Mercu Buana dan Diskusi bersama dengan Biro Teknologi Informasi Mercu Buana. Proses ini menemukan kelemahan utama dari mesin presensi fisik yang mudah dimanipulasi dengan foto cetak, dan menentukan siapa saja yang bisa mengakses sistem dengan hak berbeda, seperti SDM, Kabiro, Kabag, dan Peserta.
2. Fase Perancangan Sistem: Merancang bagaimana antar komponen perangkat lunak berinteraksi dengan membuat diagram Use Case, diagram Aktivitas yang mencakup proses persetujuan, dan diagram Sequence untuk menunjukkan cara komunikasi yang asinkron berlangsung. Aktivitas ini ditutup dengan membuat model struktur tabel basis data relasional menggunakan spesifikasi ERD.
3. Fase Implementasi dan Konstruksi Kode: Mengunggah *dataset* gambar ke server cloud Ultralytics untuk melatih model YOLO11n menggunakan metode optimasi AdamW, kemudian menghasilkan dan mengekspor bobot model ke dalam format ONNX.
 - a. Membuat skrip microservice menggunakan Flask Python untuk memuat InsightFace dan melakukan perhitungan *Cosine Similarity* secara matematika.
 - b. Membuat struktur antarmuka pengguna yang bisa diinteraksikan dengan React JS yang memiliki kemampuan merekam video berkualitas namun dalam ukuran yang lebih kecil dan pemeriksaan *active liveness* lokal.
 - c. Membuat logika untuk memulai transaksi atomik dalam *framework* PHP Laravel agar basis data dapat dilindungi dari masuknya informasi absensi yang rusak karena terputusnya koneksi jaringan.
4. Fase Pengujian Terintegrasi: Mengumpulkan data dengan mensimulasikan penggunaan sistem nyata, menggunakan wajah manusia, foto cetak berukuran medium pada kertas, serta gambar potret yang dipantulkan dari layar ponsel. Selama fase ini, peneliti juga mengawasi log peramban web

(*console browser*) untuk mencatat nilai latensi jaringan secara terus-menerus.

5. Fase Pemeliharaan dan Peluncuran: Membagikan repositori kode yang sudah melewati uji kelayakan ke dalam lingkungan server dengan sistem operasi Ubuntu Linux. Aktivitas setelah peluncuran meliputi pengawasan terhadap peningkatan penggunaan memori RAM agar tidak terjadi gangguan layanan.

3.8 Evaluasi Hasil Penelitian

Melalui serangkaian pengujian terintegrasi, arsitektur sistem presensi yang menggabungkan antarmuka React JS, layanan kecerdasan buatan Flask, dan *framework* Laravel telah dievaluasi secara komprehensif. Hasil pengujian membuktikan bahwa Sistem keamanan dua lapis (*two-stage cascaded liveness verification*) menggunakan model YOLO11n berformat ONNX sebagai *passive liveness* dan MediaPipe Face Landmarker berhasil beroperasi dengan baik sebagai solusi keamanan berbasis perangkat lunak yang kemampuannya dapat dibandingkan dengan perangkat keras biometrik mahal. Model ini secara ketat memvalidasi keaslian wajah pengguna sebagai syarat mutlak sebelum pustaka InsightFace diizinkan mengeksekusi gambar menjadi *vector embeddings*, lalu pencocokan identitas melalui algoritma *Cosine Similarity* dengan ambang batas ≥ 0.5 .

Selain keandalan kecerdasan buatan, efisiensi arsitektur sistem juga terbukti unggul berkat implementasi *caching embedding* pada memori RAM Flask yang secara signifikan menghilangkan beban komputasi ganda pada basis data PostgreSQL. Didukung oleh metode transmisi data secara *asynchronous*, sistem ini menunjukkan ketahanan saat menghadapi pengujian beban (*stress testing*) berupa serangan bingkai palsu tanpa mengalami perlambatan yang berarti. Dengan terjaminnya keamanan integritas data transaksi pada sisi *backend*, perangkat lunak ini secara meyakinkan layak untuk diimplementasikan di Universitas Mercu Buana.