

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu mengenai penulisan tugas akhir telah dilakukan oleh banyak peneliti dan akademisi di berbagai disiplin ilmu. Beberapa temuan umum yang ditemukan dalam penelitian.

Tabel 2.1 Penelitian Terdahulu

No	Topik	Metode	Hasil	Ref
1.	Pengembangan sistem presensi mahasiswa berbasis web yang mengintegrasikan deteksi wajah <i>real-time</i> dan keamanan <i>anti-spoofing</i> .	Menggunakan algoritma YOLOv11 untuk deteksi wajah, InsightFace untuk ekstraksi fitur, dan <i>Liveness Detection</i> untuk <i>anti-spoofing</i> . Sistem dibangun dengan <i>framework</i> Flask (<i>backend</i>), ReactJS (<i>frontend</i>), dan MySQL.	Pengujian menunjukkan YOLOv11 mampu mendeteksi wajah dengan cepat dan akurat. Metrik evaluasi menghasilkan <i>Precision</i> 96,8%, <i>Recall</i> 95,7%, Akurasi 96,3%, dan <i>F1-Score</i> 96,24%.	(Dewi et al., 2025)
2.	Peningkatan sistem kehadiran mahasiswa menggunakan teknologi biometrik wajah cerdas untuk mengatasi masalah absensi manual.	Menggunakan kamera HD untuk akuisisi citra, algoritma <i>Deep Learning</i> untuk deteksi, InsightFace untuk ekstraksi fitur, dan <i>Cosine Similarity</i> untuk pencocokan	Sistem berhasil melakukan verifikasi wajah dan pencatatan kehadiran secara otomatis dengan tingkat keberhasilan tinggi, mengurangi ketergantungan	(Mohaimen et al., 2025)

		identitas di server lokal.	pada metode manual dan meningkatkan integritas akademik.	
3.	Analisis komparatif kinerja deteksi objek antara YOLOv11 dan YOLOv12 (model <i>Nano</i> dan <i>Small</i>).	Eksperimen pelatihan dilakukan sebanyak 100 <i>epoch</i> pada kedua versi model. Parameter yang diukur meliputi waktu inferensi, kecepatan pemrosesan, penggunaan memori, dan ukuran model.	YOLOv11 terbukti lebih efisien dengan waktu eksekusi lebih cepat (219 detik vs 303 detik pada model N) dan penggunaan memori yang jauh lebih rendah (126 MB vs 3674 MB) dibandingkan YOLOv12.	(G. A. A. Putra & Wowor, 2025)
4.	Evaluasi kemampuan pustaka InsightFace sebagai solusi sistem presensi di lingkungan sekolah (SMP Negeri 2 Tarokan).	Pengumpulan data wajah melalui video dan foto, diproses menggunakan <i>OpenCV</i> dan pustaka <i>InsightFace</i> untuk ekstraksi fitur (<i>embedding</i>). Evaluasi menggunakan data	Sistem menunjukkan performa yang cukup tinggi dengan Akurasi mencapai 86,67%, membuktikan efektivitas <i>InsightFace</i> untuk aplikasi presensi berbasis wajah.	(Setiawan et al., 2025)

		uji yang belum pernah dilatih.		
5.	Pengembangan sistem deteksi otomatis untuk identifikasi jenis burung merpati menggunakan arsitektur YOLO terbaru.	Menggunakan algoritma YOLOv11 dengan tahapan: pengumpulan <i>dataset</i> , pra-pemrosesan, pelabelan data, dan pelatihan model. <i>Input</i> sistem dapat berupa unggahan gambar atau webcam.	Hasil pengujian sistem mencapai Akurasi 93,3%, <i>Precision</i> 90,6%, <i>Recall</i> 96,7%, dan <i>F1-Score</i> 93,6%, menunjukkan kemampuan YOLOv11 dalam klasifikasi objek yang mirip.	(Vanesa et al., 2025)
6.	Sistem deteksi tingkat kematangan tomat (mentah, setengah matang, matang) untuk efisiensi sortasi pasca panen.	Melatih 4 varian model YOLO11 (n, s, m, l) menggunakan platform Google Colab dengan 100 <i>epoch</i> . <i>Dataset</i> dilabeli via Roboflow dan dilakukan augmentasi data.	Model YOLO11 menunjukkan peningkatan dalam ekstraksi fitur dan presisi deteksi objek dibandingkan versi sebelumnya, memberikan performa yang lebih baik untuk klasifikasi visual.	(Alfian & Paputungan, 2025)
7.	Solusi sistem absensi cerdas yang menggabungkan pengenalan wajah dengan fitur	Menggunakan algoritma <i>Haar Cascade Classifier</i> sebagai inti deteksi wajah dan teknik	Metode yang diusulkan mencapai rata-rata Akurasi 98,90% dalam pelacakan	(Supriatna et al., 2024)

	keamanan <i>Anti-Spoofing</i> .	pemrosesan citra untuk fitur <i>anti-spoofing</i> guna mencegah kecurangan.	kehadiran dengan tingkat kepercayaan (confidence) secara konsisten di atas 80%.	
8.	Peningkatan performa deteksi wajah pada video 360 derajat menggunakan teknik augmentasi gambar tingkat lanjut.	Menggunakan algoritma YOLOv8 yang dikombinasikan dengan metode augmentasi ekuirektangular (<i>equirectangular augmentation</i>) untuk mengatasi distorsi geometri pada video panoramik.	Teknik augmentasi yang diusulkan meningkatkan akurasi deteksi sebesar 1,346% dibandingkan strategi augmentasi standar YOLOv8 pada lingkungan proyeksi ekuirektangular.	(Arady et al., 2025)
9.	Sistem pencatatan kehadiran mahasiswa/pegawai untuk menghilangkan ketergantungan pada kartu atau pencatatan manual.	Menggunakan metode Haar Cascade (Viola-Jones) untuk deteksi objek wajah. Pengujian dilakukan menggunakan <i>dataset</i> 10 mahasiswa dengan 50 citra wajah per orang.	Sistem berhasil mencapai Akurasi 100% dalam mengidentifikasi wajah pada posisi berbeda, dan 90% pada pengujian lanjutan, meningkatkan efisiensi proses absensi.	(Putra et al., 2024b)

10	Biometric Authentication based on <i>Liveness Detection</i> Using Face Landmarks and <i>Deep Learning</i> Model	Sistem dibangun menggunakan Python, OpenCV, dlib, MediaPipe, dan TensorFlow untuk mendeteksi 5 gerakan wajah (geleng, angguk, kedip, senyum, buka mulut). Fungsionalitasnya diuji secara <i>real-time</i> melalui prototipe aplikasi Android yang terhubung ke server API berbasis FastAPI.	Model CNN untuk deteksi senyum dan kedipan mata mencapai akurasi di atas 90%. Integrasi API dengan aplikasi Android berjalan lancar, membuktikan bahwa deteksi <i>liveness</i> secara <i>real-time</i> berhasil diterapkan pada perangkat berspesifikasi rendah.	(Jie et al., 2023)
11	Pengembangan solusi <i>anti-spoofing</i> canggih menggunakan kerangka kerja YOLO untuk mencegah serangan biometrik pada sistem pengenalan wajah.	Menggunakan CNN untuk menilai fitur wajah dan model YOLO untuk deteksi wajah secara <i>real-time</i> . Sistem dirancang untuk bekerja dengan kamera RGB standar tanpa perangkat keras khusus.	Hasil awal menunjukkan peningkatan signifikan dalam akurasi deteksi wajah asli vs <i>spoofing</i> dibandingkan metode tradisional, serta efektif digunakan pada <i>smartphone</i> .	(Vardhan et al., 2025)

12	Penerapan algoritma terbaru YOLO11 untuk mendeteksi dan menyortir spesies ikan laut secara otomatis.	Menggunakan model YOLO11n dan YOLO11s. <i>Dataset</i> dibagi dengan 3 variasi, salah satunya adalah rasio 70% <i>Training</i> , 20% <i>Validation</i> , 10% <i>Testing</i> . Dilakukan juga Pre-processing dan <i>Augmentation</i> .	Model YOLO11 berhasil diimplementasikan untuk membantu proses penyortiran. Metodologi pembagian data (Split Data) terbukti efektif untuk melatih model pada <i>dataset</i> yang beragam.	(Permatasari & Putra, 2025)
13	Perancangan purwarupa (prototype) sistem pengenalan wajah menggunakan algoritma deteksi objek terbaru YOLO11 yang diimplementasikan pada perangkat keras <i>mini-Raspberry Pi</i> .	Metode pengembangan prototipe dengan pembagian <i>dataset</i> citra wajah menjadi 80% (200 gambar) untuk <i>Training</i> , 12% (30 gambar) untuk <i>Validasi</i> , dan 8% (20 gambar) untuk <i>Testing</i> . Pengujian dilakukan secara <i>real-time</i> menggunakan <i>input</i> video bergerak.	Sistem berhasil mendeteksi dan mengenali wajah dengan performa sangat tinggi pada perangkat <i>Raspberry Pi</i> . Pengujian pada 10 video sampel menghasilkan tingkat Akurasi 95% dan Presisi 100%	(Listia Setiawati, 2025)
14	Perancangan sistem manajemen	Pengembangan full-stack	Terciptanya sistem verifikasi identitas	(Yadav, 2025)

	kehadiran berbasis AI untuk mengurangi kesalahan manusia dan ketergantungan perangkat keras eksternal.	menggunakan Python, Flask (sebagai <i>framework</i> web ringan), PostgreSQL (sebagai RDBMS), dan Dlib. Facial <i>embeddings</i> digunakan untuk verifikasi identitas.	<i>real-time</i> yang menghilangkan kebutuhan <i>input</i> manual. Penggunaan PostgreSQL terbukti handal untuk manajemen data absensi.	
15	Analisis implementasi dan perbandingan kinerja basis data relasional untuk sistem informasi.	Studi kasus pencatatan data mahasiswa menggunakan dua RDBMS (MySQL dan PostgreSQL) melalui serangkaian operasi CRUD (<i>Create, Read, Update, Delete</i>).	Kedua database efektif, namun PostgreSQL menawarkan keunggulan tambahan dalam dukungan tipe data yang lebih kompleks dan integritas data, mendukung penggunaannya untuk sistem yang lebih <i>robust</i> .	(Dachi & Suhada, 2025)
16	Face detection and <i>anti-spoofing</i> on desktop applications using you only look once	Sistem deteksi <i>liveness</i> wajah secara <i>real-time</i> yang dikembangkan menggunakan	Dalam kondisi laboratorium yang terkontrol, sistem mencapai metrik kinerja yang luar biasa dengan akurasi 1.0, presisi	(Faisal & Nurhaida, 2025)

		algoritma YOLOv8.	1.0, recall 1.0, dan mean Average Precision (mAP) sebesar 0.96.	
17	Sistem Pengenalan Wajah Real-Time Menggunakan YOLOv7 untuk Akses Gedung TVRI Palembang Berbasis Web	Merancang dan mengembangkan sistem pengenalan wajah secara <i>real-time</i> berbasis web menggunakan algoritma YOLOv7.	Dalam 100 <i>epoch</i> pelatihan, sistem mencapai akurasi deteksi wajah sebesar 96,6% dan akurasi pengenalan wajah sebesar 90%, serta berhasil mengidentifikasi hampir seluruh wajah yang terdaftar secara <i>real-time</i> .	(Indra et al., 2025)
18	Sistem Presensi Mahasiswa Berbasis Pengenalan Wajah Real-Time dengan Deteksi <i>Anti-Spoofing</i> Menggunakan YOLOv8 dan ArcFace	Sistem presensi berbasis pengenalan wajah secara <i>real-time</i> menggunakan algoritma YOLOv8 untuk deteksi wajah beserta fitur <i>anti-spoofing</i> , dan menggunakan ArcFace untuk pengenalan identitas.	Sistem mampu mengenali seluruh mahasiswa yang terdaftar dengan tingkat keberhasilan 100%, di mana model <i>anti-spoofing</i> YOLOv8 membedakan wajah asli dan palsu dengan nilai mAP@0.5 sebesar 0,995.	(Satria et al., 2026)

19	Deteksi YOLOv8 Dan Pengenalan Wajah Menggunakan RESNET50 Pada Gereja	Menggunakan metode YOLOv8 untuk deteksi ketersediaan kursi dan wajah, serta <i>Convolutional Neural Network</i> (CNN) berbasis model ResNet-50 untuk pengenalan wajah secara otomatis.	Hasil pengujian mendapatkan rata-rata akurasi keseluruhan mencapai 85,26% dan akurasi deteksi sebesar 95,46%, dengan model latih YOLOv8 mencapai 97% mAP50.	(Dony & Lubis, 2025)
20	Pengembangan Sistem <i>Anti-Spoofing</i> Berbasis <i>Face Recognition</i> Menggunakan Arsitektur YOLOv8n	Pengembangan sistem <i>anti-spoofing real-time</i> yang mengintegrasikan YOLOv8n untuk klasifikasi 4 kategori wajah (<i>real</i> , <i>printed</i> , <i>digital</i> , dan <i>mask</i>) dengan verifikasi <i>liveness</i> melalui deteksi kedipan berbasis <i>Eye Aspect Ratio</i> (EAR).	Model <i>pretrained</i> dari YOLOv8n mencapai precision 99,5%, recall 98,6%, mAP50 99,4%, dan mAP50-95 90,4%.	(Tanujaya et al., 2026)

2.2 Teori Utama

2.2.1 Face Recognition

Dalam era modernisasi infrastruktur teknologi informasi, kebutuhan akan sistem verifikasi identitas yang efektif dan efisien semakin mendesak, terutama di instansi pemerintahan dan pelayanan publik. Salah satu solusi teknologi yang relevan adalah sistem presensi berbasis pengenalan wajah. Teknologi ini menawarkan keunggulan dibandingkan metode konvensional karena mampu memberikan keamanan data yang lebih tinggi, meminimalisir peluang manipulasi kehadiran, serta memungkinkan identifikasi individu secara otomatis tanpa kontak fisik.

Secara teknis, pengenalan wajah bekerja dengan cara menghitung dan menganalisis karakteristik geometri wajah, seperti posisi, bentuk, dan sudut fitur wajah dari *input* visual (video atau gambar). Proses ini umumnya terdiri dari tiga tahapan utama: deteksi wajah (*detection*), ekstraksi fitur (*feature extraction*), dan pencocokan fitur (*matching*). Sistem akan membandingkan fitur wajah yang ditangkap secara *real-time* dengan data referensi yang tersimpan dalam basis data untuk memverifikasi keaslian identitas pengguna (Hidayat et al., 2025).

2.2.2 Deep Learning dan Convolutional Neural Network (CNN)

Untuk mendukung proses pengenalan wajah yang akurat, digunakan metode *Deep Learning* dengan arsitektur *Convolutional Neural Network* (CNN). CNN merupakan jenis jaringan saraf tiruan yang dirancang khusus untuk menangani komputasi data berbentuk grid, seperti citra digital. Arsitektur ini dinilai sangat efektif dalam tugas klasifikasi dan identifikasi pola visual karena cara kerjanya yang meniru mekanisme korteks visual pada otak manusia dalam memproses informasi visual (ANHAR & PUTRA, 2023). Kemampuan CNN dalam memahami detail citra menjadikannya metode standar dalam ekstraksi fitur wajah yang kompleks.

2.2.3 Algoritma Cosine Similarity

Metode *Cosine Similarity* pada dasarnya mengukur tingkat kemiripan antara dua vektor berdimensi dengan menghitung nilai kosinus dari sudut di antara kedua

vektor tersebut. Berbeda dengan pengukuran jarak Euclidean yang berfokus pada jarak lurus spasial antar titik data, *cosine similarity* lebih berfokus pada kedekatan sudut antar vektor. Karakteristik ini menjadikannya sangat efektif dan stabil untuk membandingkan objek dengan data berdimensi tinggi (Ciulla & Varadharajan, 2023).

Dalam konteks penelitian ini, *Cosine Similarity* diimplementasikan untuk menentukan nilai kemiripan identitas mahasiswa magang secara *real-time*. Pengukuran tidak dilakukan berdasarkan nilai piksel gambar mentah, melainkan dengan membandingkan representasi vektor numerik (*face embeddings*) yang telah diekstraksi oleh pustaka InsightFace. Perhitungan tingkat kemiripan tersebut dirumuskan dalam persamaan matematis berikut (Hikmah et al., 2022):

$$\text{Cosine Similarity}(X, Y) = \frac{X \cdot Y}{|X| \cdot |Y|} \quad 2.1$$

Dimana:

- X mewakili vektor fitur (*face embedding*) dari citra wajah yang ditangkap secara *real-time* saat proses presensi.
- Y mewakili vektor fitur wajah referensi yang telah didaftarkan dan tersimpan di dalam basis data (PostgreSQL).
- $X \cdot Y$ adalah hasil perkalian titik (*dot product*) antara vektor X dan vektor Y .
- $|X|$ dan $|Y|$ merupakan nilai magnitudo (panjang norma) dari masing-masing vektor X dan Y .

Pengukuran ini bekerja dengan menilai sejauh mana vektor wajah saat presensi (X) memiliki arah sudut yang sama dengan vektor wajah referensi (Y). Hasil perhitungan dari *Cosine Similarity* akan menghasilkan nilai metrik kesamaan. Nilai yang semakin mendekati 1 menunjukkan tingkat kemiripan identitas yang sangat tinggi antara wajah yang terdeteksi di kamera dengan wajah yang ada di basis data (Chiu et al., 2023). Sistem kemudian akan membandingkan hasil perhitungan ini dengan nilai ambang batas (*threshold*) tertentu guna memvalidasi apakah identitas tersebut dikenali untuk dicatat kehadirannya atau ditolak oleh sistem.

Nilai ambang batas yang paling tepat tergantung pada model embedding dan jenis data yang digunakan, sehingga tidak ada angka tetap yang cocok untuk semua sistem pengenalan wajah. Sebagai contoh, penelitian oleh (Dwiky Hermawan et al., 2026) yang menerapkan face embedding berbasis ArcFace dalam sistem kehadiran menetapkan nilai ambang sebesar 0,43, sedangkan penelitian oleh (Daiva Geralda et al., 2026) yang menggunakan model FaceNet dalam kasus yang sama menetapkan nilai ambang sebesar 0,6. Variasi ini menunjukkan bahwa nilai ambang batas perlu disesuaikan dengan sifat model ekstraksi fitur dan situasi kelompok pengguna di setiap sistem.

2.2.4 Liveness Detection (Anti-Spoofing)

Meskipun sistem pengenalan wajah memiliki akurasi tinggi, sistem ini rentan terhadap serangan manipulasi atau *spoofing* menggunakan foto, video, atau topeng. Oleh karena itu, diperlukan mekanisme keamanan tambahan berupa *Liveness Detection* atau *Anti-Spoofing*. Teknologi ini menerapkan berbagai teknik, seperti analisis tekstur dan deteksi gerakan, untuk membedakan antara wajah asli dan percobaan penipuan (Supriatna et al., 2024).

Dalam konteks biometrik, *Face Liveness Detection* didefinisikan sebagai kemampuan sistem untuk mendeteksi apakah wajah yang ditangkap berasal dari individu hidup yang hadir secara fisik, atau sekadar artefak palsu. Sistem ini menerapkan serangkaian fitur teknis untuk menangkal serangan *spoofing*, di mana replika wajah seperti cetakan gambar atau topeng 3D berbahan silikon sering digunakan untuk mengecoh pemindai guna melewati prosedur identifikasi dan otentikasi keamanan (Ardikatama, 2022).

Ditinjau dari mekanisme kerjanya, metode *liveness detection* umumnya dibagi menjadi dua pendekatan utama, yaitu *passive liveness* dan *active liveness*. Pendekatan *passive* bekerja dengan menganalisis gambar atau video tanpa mengganggu pengguna, tanpa membutuhkan aksi langsung dari mereka, misalnya melalui pengamatan pola tekstur, tanda-tanda layar, atau pola pencahayaan pada gambar yang diambil. Sebaliknya, pendekatan *active* menerapkan mekanisme *challenge-response*, di mana pengguna diminta melakukan tindakan atau ekspresi acak seperti menutup mata, membuka mulut, atau

memutar kepala untuk menunjukkan bahwa mereka benar-benar ada di sana secara langsung (Widjaya & Wicaksana, 2023). Kedua cara ini memiliki kelebihan yang saling melengkapi, metode pasif lebih cepat dan menawarkan pengalaman yang lebih mudah digunakan karena tidak memerlukan banyak usaha, tetapi rentan terhadap serangan berupa rekaman ulang yang bagus dan bisa menyerupai wajah asli. Sementara itu, metode active memberikan tingkat keamanan yang lebih baik karena jenis serangan seperti foto atau video yang sudah disiapkan tidak bisa menanggapi perintah acak secara langsung, meski prosesnya membutuhkan waktu lebih lama.

2.2.5 Algoritma YOLO (You Only Look Once) Versi 11

YOLO dikenal sebagai salah satu algoritma deteksi objek yang unggul dalam kecepatan dan akurasi karena kemampuannya memanfaatkan akselerasi perangkat keras berbasis *Graphics Processing Unit (GPU)* melalui kerangka kerja seperti Darknet (Zophie & Triharminto, 2022). Dalam perkembangannya, Ultralytics merilis varian terbaru yaitu YOLO11, yang menawarkan peningkatan efisiensi signifikan dibandingkan pendahulunya. Arsitektur YOLO11 mengadopsi pendekatan *single-stage detector*, di mana pemrosesan citra dilakukan dalam satu kali lintasan maju (*forward pass*). Mekanisme ini menghasilkan prediksi yang jauh lebih cepat dibandingkan metode *two-stage detector*. Model ini menyediakan berbagai skala mulai dari *nano* hingga *extra-large* untuk mendukung berbagai tugas seperti deteksi objek, segmentasi, dan klasifikasi, menjadikannya solusi ideal untuk aplikasi *real-time* (Alfian & Paputungan, 2025).

2.2.6 Pustaka InsightFace

InsightFace merupakan pustaka *open source* berbasis *Deep Learning* yang dirancang khusus untuk analisis wajah, meliputi tugas deteksi, penyelarasan (*alignment*), dan ekstraksi fitur (*embedding*) baik dalam format 2D maupun 3D. Fungsionalitas yang komprehensif ini memungkinkan proses identifikasi wajah berjalan secara akurat dan efisien (Setiawan et al., 2025). Insight Face menyediakan lima varian model (termasuk seri buffalo dan antelopev2) yang umumnya menghasilkan berbagai parameter *output* mulai dari *bounding box*, *landmark*,

hingga atribut demografis. Namun, terdapat pengecualian pada model *buffalo_sc* yang dirancang secara efisien untuk hanya menghasilkan data deteksi dan pengenalan wajah. Jika prioritas utama adalah ketepatan validasi, model *buffalo_l* merupakan pilihan terbaik karena memiliki tingkat akurasi yang tinggi dibandingkan varian lainnya (ARRJV, 2024).

2.2.7 Active Liveness Detection (Challenge-Response)

Sebagai lapisan verifikasi tambahan yang bersifat *decisive* (penentu akhir), penelitian ini mengimplementasikan metode *active liveness detection* berbasis *challenge-response*. Metode ini mengharuskan pengguna merespons instruksi acak yang dihasilkan sistem secara *real-time*, seperti menoleh ke arah tertentu atau tersenyum, sebelum sistem mengonfirmasi kehadiran fisik penggunanya (Widjaya & Wicaksana, 2023). Dibandingkan dengan pendekatan *passive* yang murni menganalisis tekstur citra statis, mekanisme *challenge-response* secara fundamental lebih sulit ditembus oleh serangan media foto cetak maupun *video replay* pra-rekam, karena media tersebut tidak memiliki kapasitas untuk merespons instruksi yang dihasilkan secara acak dan tidak dapat diprediksi sebelumnya.

Dalam konteks penelitian ini, mekanisme *active liveness* dirancang untuk beroperasi sebagai lapisan kedua setelah validasi *passive liveness* oleh YOLO11n dan pencocokan identitas oleh *Cosine Similarity* dinyatakan konsisten. Pendekatan bertingkat (*cascaded*) ini diadopsi untuk menutup celah keamanan yang melekat pada metode *passive* saja, di mana serangan media dinamis seperti pantulan layar ponsel berpotensi memiliki karakteristik tekstur yang menyerupai wajah asli sehingga dapat mengecoh model klasifikasi berbasis tekstur.

2.2.8 MediaPipe Face Landmarker dan Estimasi Pose Kepala

MediaPipe Face Landmarker adalah model computer vision yang dikembangkan oleh Google untuk mendeteksi 468 titik landmark tiga dimensi di wajah secara langsung, termasuk memperkirakan bentuk wajah tanpa perlu menggunakan sensor kedalaman khusus (Soomro et al., 2023). Selain memberikan koordinat landmark, model ini juga menyediakan *facial transformation matrix*, yaitu matriks rotasi 4x4 yang mengubah model wajah standar ke dalam sistem

koordinat kamera pada setiap frame(Davalos et al., 2025). Dengan menggunakan vektor arah (forward vector) yang diambil dari kolom matriks rotasi tersebut, sudut yaw (arah kiri-kanan) dan pitch (arah atas-bawah) kepala pengguna dapat dihitung secara matematika menggunakan fungsi trigonometri arctangent.

Dibandingkan dengan metode konvensional yang menggunakan perbandingan posisi landmark dua dimensi (seperti letak hidung terhadap dahi dan dagu), pendekatan estimasi pose berbasis *facial transformation matrix* dianggap lebih konsisten karena bekerja dalam ruang rotasi tiga dimensi yang tidak langsung dipengaruhi oleh jarak wajah terhadap kamera. Dalam penelitian ini, nilai delta yaw dan pitch dihitung berdasarkan nilai dasar yang diperoleh dari proses kalibrasi singkat saat wajah pengguna berada dalam posisi netral. Nilai dasar ini digunakan sebagai acuan sebelum pengguna diberi rangkaian instruksi arah kepala dan senyum secara acak tanpa dilakukan berulang kali secara terurut.

2.3 Teori Pendukung

2.3.1 Framework Laravel

Framework Laravel merupakan kerangka kerja pengembangan aplikasi web berbasis bahasa pemrograman PHP yang mengadopsi pola arsitektur *Model-View-Controller*. Implementasi pola MVC memungkinkan pemisahan komponen aplikasi ke dalam tiga lapisan fungsional independen, yang secara signifikan meningkatkan skalabilitas, kemudahan pemeliharaan kode, serta efisiensi dalam proses pengembangan(Harahap et al., 2022).

1. Model: Berperan sebagai representasi entitas data dan lapisan yang berinteraksi langsung dengan basis data relasional. Dalam ekosistem Laravel, manipulasi data difasilitasi oleh *Object-Relational Mapping* yang dikenal sebagai Eloquent, yang memungkinkan pengembang melakukan operasi basis data secara deklaratif tanpa perlu menyusun kueri SQL secara manual (Nguyen et al., 2022).
2. View: Bertanggung jawab atas penyajian antarmuka pengguna (*user interface*). Dalam paradigma sistem modern yang sebarterpisah (*decoupled architecture*), fungsi View dapat diakomodasi melalui layanan

API (*Application Programming Interface*) yang mentransmisikan representasi data berformat JSON kepada aplikasi eksternal atau pustaka *frontend*.

3. **Controller:** Berfungsi sebagai mediator yang menghubungkan Model dan View. Komponen ini mengelola *input* dari pengguna melalui mekanisme *routing*, mengeksekusi permintaan, berinteraksi dengan Model, serta menentukan respons akhir yang dikembalikan kepada klien.

Dalam konteks administrasi digital, Laravel berfungsi sebagai landasan utama dalam manajemen logika bisnis dan kontrol alur data relasional. Sistem manajemen presensi memerlukan validasi aturan yang ketat—mencakup pengelolaan data pengguna, pemetaan otoritas akses, pencatatan waktu kehadiran, hingga penyusunan laporan periodik. Melalui struktur arsitekturnya, Laravel mampu memfasilitasi manipulasi data pada basis data relasional secara aman dari kerentanan siber seperti SQL Injection, sekaligus menjamin konsistensi dan integritas dalam pelaksanaan aturan operasional sistem presensi (Nguyen et al., 2022).

2.3.2 React JS

React JS merupakan pustaka JavaScript *open-source* yang bersifat deklaratif, efisien, dan fleksibel, yang dirancang untuk pengembangan antarmuka pengguna, terutama pada aplikasi berbasis *Single Page Application* (Nabilla & Ichwani, 2022). Dua pilar fundamental yang mendasari kemampuan React JS dalam memproses data secara dinamis adalah *Reactive Component* dan *Virtual DOM*.

1. **Reactive Component:** Antarmuka dalam ekosistem React JS dibangun melalui komponen-komponen modular yang terisolasi dan mandiri. Setiap komponen memiliki manajemen data internal yang disebut *state*. Sifat reaktif terealisasi saat terjadi perubahan pada nilai *state*, sistem secara otomatis memperbarui komponen terkait agar sinkron dengan data mutakhir tanpa mengintervensi komponen lainnya.
2. **Virtual DOM:** Secara konvensional, setiap perubahan data pada halaman web menuntut peramban untuk memperbarui keseluruhan struktur pohon DOM fisik, sebuah proses yang intensif terhadap waktu dan sumber daya

komputasi. React memitigasi kendala ini dengan mengimplementasikan replika DOM di dalam memori yang dikenal sebagai *Virtual DOM*. Saat terjadi pembaruan data, React menerapkan algoritma *diffing* untuk mengidentifikasi perbedaan spesifik antara *Virtual DOM* baru dan lama, sehingga hanya memperbarui segmen DOM fisik yang mengalami perubahan (Vyas, 2022).

Efisiensi manipulasi DOM ini sangat krusial bagi sistem yang melibatkan interaksi perangkat keras intensif, seperti antarmuka kamera. Dalam proses pemindaian wajah, aplikasi dituntut untuk menangkap alur *frame* gambar dari *webcam* secara kontinu dan *real-time*. Implementasi *Virtual DOM* dan *Reactive Component* memungkinkan antarmuka untuk menangkap, memproses, serta mentransmisikan *frame* gambar ke server secara dinamis tanpa memicu pemuatan ulang halaman (*page reload*), sehingga menjamin pengalaman pengguna yang responsif dan berkesinambungan (*seamless*) (A Jartarghar et al., 2022).

Karakteristik reaktif ini juga digunakan dalam penerapan verifikasi state machine *active liveness*, di mana antarmuka secara otomatis berpindah dari satu tahap ke tahap lainnya (seperti kalibrasi, instruksi tantangan, hingga hasil verifikasi) berdasarkan perubahan state yang diterima dari proses inferensi MediaPipe di perangkat pengguna, tanpa memerlukan pemuatan ulang halaman.

2.3.3 Open Neural Network Exchange (ONNX)

Open Neural Network Exchange merupakan standar terbuka yang diformulasikan untuk merepresentasikan model *machine learning* dan *deep learning* secara lintas platform. ONNX memfasilitasi interoperabilitas yang signifikan antar berbagai *framework* kecerdasan buatan, memungkinkan model yang dikembangkan dalam lingkungan spesifik untuk ditransfer dan diimplementasikan pada arsitektur penyebaran (*deployment*) yang berbeda (Zaccardi et al., 2023). Secara operasional, eksekusi model berbasis ONNX didukung oleh pustaka *onnxruntime*, yang berperan dalam mengoptimasi grafik komputasi guna meminimalkan utilisasi sumber daya perangkat keras (Rane et al., 2024).

Dalam konteks penelitian ini, model YOLO11n yang telah melalui fase pelatihan dikonversi ke dalam format ONNX guna mereduksi beban komputasi yang inheren pada *framework* asalnya (Dai & Fan, 2022). Transformasi model YOLO11n ke dalam format ONNX terbukti meningkatkan efisiensi proses inferensi khususnya pada tugas deteksi keaslian wajah (*liveness detection*) menjadikannya lebih ringan, stabil, dan memiliki latensi yang lebih rendah. Optimalisasi *runtime* dilakukan dengan memanfaatkan penyedia komputasi berbasis CPU (*CPUExecutionProvider*). Atribut performa ONNX yang hemat memori serta latensi yang minimal menjadikan teknologi ini sangat kompatibel untuk diintegrasikan dengan *microframework* Flask, sehingga menghasilkan solusi kecerdasan buatan yang responsif tanpa ketergantungan pada akselerasi perangkat keras GPU (*Graphics Processing Unit*) (Rane et al., 2024).

2.3.4 *Microframework* Flask dan Rest API

Flask diklasifikasikan sebagai *microframework* berbasis bahasa pemrograman Python yang berfungsi sebagai kerangka kerja pengembangan aplikasi web. Istilah "*micro*" merujuk pada karakteristiknya yang ringan dan tidak memiliki ketergantungan bawaan terhadap pustaka tertentu, seperti validasi formulir atau basis data, yang biasanya diserahkan kepada pihak ketiga (Yusuf Azhari, 2022). Fleksibilitas ini memungkinkan pengembang untuk membuat struktur aplikasi yang terorganisir. Dalam konteks sistem kecerdasan buatan, Flask berperan penting untuk mengintegrasikan skrip deteksi dan pengenalan wajah menjadi sebuah *Application Programming Interface* (API), yang berfungsi sebagai jembatan komunikasi data antara model pemrosesan citra dan antarmuka pengguna (C. S. Putra et al., 2024).

Melalui mekanisme ini, antarmuka kamera pada React JS akan menangkap bingkai citra digital (*frame*) secara berkala dan mentransmisikannya satu per satu ke server menggunakan Fetch API melalui protokol HTTP berbasis REST API. Pendekatan ini memastikan bahwa setiap data *frame* dikirimkan dalam bentuk permintaan HTTP independen yang terstruktur, sehingga memudahkan pengelolaan status (*stateless*) dan pemrosesan antrian data pada sisi server Flask (Rezvani et al., 2022)

2.3.5 Basis Data PostgreSQL

PostgreSQL merupakan *Relational Database Management System* (RDBMS) berbasis *open-source* yang dikembangkan oleh Departemen Ilmu Komputer Berkeley. Sama halnya dengan MySQL, sistem ini menggunakan SQL sebagai bahasa kueri utamanya. Dengan rekam jejak penggunaan selama kurang lebih tiga dekade, PostgreSQL dirancang untuk mengelola data dalam tabel yang saling terhubung (relasional). Kemampuannya dalam menangani pemrosesan data yang kompleks menjadikannya solusi yang banyak diterapkan pada berbagai platform, mulai dari aplikasi seluler, aplikasi web, hingga keperluan analitik (Santi, 2024). Berbeda dengan MySQL, PostgreSQL lebih diunggulkan dalam hal stabilitas dan kepatuhan ketat terhadap standar SQL. Sistem ini dilengkapi dengan fitur-fitur mutakhir seperti dukungan tipe data JSONB dan *array*, kepatuhan ACID penuh, serta kontrol konkurensi versi ganda (MVCC). Hal ini membuat PostgreSQL sangat andal untuk menangani beban kerja berat yang melibatkan analisis data kompleks dan integritas transaksi tinggi (Dachi & Suhada, 2025).

2.3.6 Metode Pengembangan SDLC Waterfall

Metode *Software Development Life Cycle* (SDLC) *Waterfall* adalah model yang digunakan untuk mengembangkan sistem informasi secara sistematis dan sekuensial. Karakteristik utama dari metode ini adalah setiap tahap pengerjaan harus diselesaikan terlebih dahulu secara utuh sebelum melanjutkan ke tahap berikutnya. Karena tidak adanya pekerjaan yang berjalan paralel, fokus pada masing-masing tahapan dapat dimaksimalkan (Ningki & P, 2023). Metode ini memandang pengembangan perangkat lunak sebagai urutan proses yang searah, tidak dapat dikembalikan ke tahap sebelumnya, dan terdiri dari langkah-langkah yang membentang mulai dari analisis kebutuhan, perencanaan, desain, implementasi, pengujian, hingga pemeliharaan.

Adapun tahapan utama dalam metode *Waterfall* menurut Sommerville yang dikutip dalam (Anindyati et al., 2023) terdiri dari lima langkah utama sebagai berikut:

1. **Requirement analysis and definition:** Mendefinisikan layanan dari sebuah sistem, batasan, dan tujuan yang ditentukan dari kebutuhan pengguna sistem. Pada tahapan ini, didefinisikan secara detail spesifikasi dari sistem yang akan dikembangkan. Dalam konteks penelitian ini, tahapan ini melibatkan pendefinisian spesifikasi fungsional sistem presensi, kebutuhan data citra untuk YOLO11n, serta struktur *liveness detection*.
2. **System and software design:** Dilakukan proses identifikasi dan penggambaran abstraksi fundamental dari sistem yang akan dibangun, beserta relasi antarkomponennya. Tahap ini mencakup perancangan arsitektur *microframework* Flask API, perancangan basis data PostgreSQL, pemodelan REST API, serta desain antarmuka (*user interface*).
3. **Implementation and unit testing:** Tahap di mana desain dari perangkat lunak direalisasikan menjadi kumpulan program atau unit program dan diujikan secara fungsi terkecil dari program. Logika bisnis diimplementasikan menggunakan Laravel, antarmuka sistem dikembangkan dengan React JS, dan modul *liveness detection* dibangun menggunakan Python.
4. **Integrating and system testing:** Proses integrasi unit program dan pengujian sistem secara utuh dilakukan pada tahap ini. Tahap ini bertujuan untuk memastikan kebutuhan sistem yang didefinisikan sudah terpenuhi. Pengujian ini mencakup validasi komunikasi data antara antarmuka React JS, Flask API, dan *backend* Laravel, serta menguji ketahanan model AI terhadap upaya *spoofing*.
5. **Operation and maintenance:** Tahapan pemeliharaan yang dilakukan setelah sistem terpasang dan beroperasi di lingkungan pengguna. Fokus pada tahap ini adalah menjaga stabilitas sistem dan melakukan perbaikan jika ditemukan kendala pada saat operasional berjalan.

2.3.7 Metode Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah elemen penting dalam memastikan kualitas perangkat lunak (*software quality assurance*). Pengujian dilakukan untuk memastikan bahwa sistem yang dibangun sudah memenuhi spesifikasi

fungsionalitas, performa, serta keandalan keamanan yang diharapkan. Pada penelitian ini, pengujian dibagi menjadi tiga domain utama, yaitu pengujian model kecerdasan buatan, pengujian fungsionalitas aplikasi, dan pengujian integrasi serta performa.

2.3.7.1 Pengujian Confusion Matriks

Evaluasi kinerja model klasifikasi dilakukan menggunakan metode *Confusion Matrix*, yang mengukur kemampuan sistem dalam mengklasifikasikan data dengan membandingkan hasil prediksi terhadap data aktual. Metode ini menggunakan empat parameter utama: *True Positive* (TP) untuk data positif yang terdeteksi benar, *True Negative* (TN) untuk data negatif yang terdeteksi benar, *False Positive* (FP) untuk kesalahan deteksi data positif, dan *False Negative* (FN) untuk kesalahan deteksi data negatif (Azzam Ash'shobir et al., 2025).

2.3.7.2 Pengujian Fungsionalitas (*Black Box Testing*)

Black Box Testing, atau yang dapat disebut juga sebagai *Behavioral Testing*, merupakan metode pengujian perangkat lunak dari segi spesifikasi fungsional tanpa perlu menguji struktur internal, desain, atau kode program yang ada di dalamnya. Metode ini menguji perangkat lunak yang telah dibangun, baik pada unit-unit kecil maupun hasil yang telah terintegrasi, untuk mengetahui apakah fungsi, masukan (*input*), dan keluaran (*output*) dari perangkat lunak tersebut sesuai dengan spesifikasi yang dibutuhkan (Abdillah et al., 2023).

Pengujian ini dilakukan di akhir pembuatan perangkat lunak dengan memberikan masukan tertentu pada sistem dan secara langsung mengamati keluaran yang dihasilkan untuk memastikan sistem berfungsi dengan baik sesuai dengan skenario yang telah dirancang (Waliyansyah et al., 2022).

Dalam penelitian ini, *Black Box Testing* diterapkan untuk menguji keamanan dan ketahanan antarmuka sistem presensi melalui berbagai skenario dunia nyata, seperti pengujian presensi normal (wajah asli), deteksi serangan foto cetak atau digital (*spoofing*), pengujian silang akun antar pengguna, hingga percobaan *mismatch ID*. Pengujian ini melibatkan

berbagai partisipan untuk memvalidasi bahwa sistem mampu merespons setiap kondisi masukan dengan tepat, aman, dan menghasilkan *output* yang sesuai dengan logika bisnis.

2.3.7.3 Pengujian Integrasi dan Performa (Latensi)

Pengujian integrasi (*Integration Testing*) adalah tahapan pengujian perangkat lunak yang dilakukan setelah pengujian unit (*unit testing*) dengan tujuan untuk memverifikasi fungsionalitas, kinerja, dan keandalan komunikasi antar modul atau layanan yang berbeda ketika digabungkan. Proses ini mendemonstrasikan bahwa sistem perangkat lunak yang dibangun telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional (Prasetyo et al., 2022).

Pada arsitektur sistem berbasis *microframework* dan *Application Programming Interface* (API), pengujian integrasi sangat krusial untuk memastikan aliran transmisi data berjalan tanpa hambatan maupun kehilangan informasi (*data loss*). Implementasi saluran komunikasi yang andal berbasis RESTful API memastikan data citra dapat ditransmisikan dengan aman melalui jaringan tanpa risiko korupsi data, sehingga memungkinkan pengujian dan penerapan algoritma *Face Recognition Liveness Detection* (FRLD) berjalan dengan baik dalam skenario dunia nyata (Jie et al., 2023). Dalam penelitian ini, pengujian integrasi digunakan untuk memvalidasi komunikasi *end-to-end*, mulai dari pengambilan *frame* citra di antarmuka React JS, transmisi data ke Flask API untuk pemrosesan *Computer Vision*, hingga penyimpanan status akhir presensi di *backend* Laravel dan basis data PostgreSQL.

Selain fungsionalitas dan integrasi, keandalan sebuah sistem *real-time* juga diukur melalui pengujian performa, khususnya pada aspek latensi atau waktu respons (*response time*). Waktu respons didefinisikan secara menyeluruh sebagai total waktu yang dibutuhkan oleh pengguna sejak inisiasi permintaan (*request*) dikirimkan hingga sistem menyelesaikan pemrosesan dan menerima hasil atau respons yang sesuai (*appropriate response*) pada perangkat (Acheampong et al., 2023). Pengukuran latensi, yang umumnya dihitung dalam satuan milidetik (ms), menjadi parameter

penting untuk memastikan bahwa proses ekstraksi wajah dan *liveness detection* beroperasi secara efisien tanpa menimbulkan *bottleneck* komputasi yang dapat mengganggu pengalaman pengguna.

2.3.7.4 User Acceptance Testing (UAT)

User Acceptance Testing adalah tahap pengujian terakhir di mana pengguna sebenarnya melakukan penilaian untuk memastikan sistem yang telah dibuat sesuai dengan kebutuhan dan harapan mereka dalam penggunaan sehari-hari, seperti yang disebutkan oleh (Novita et al., 2022). UAT berbeda dengan BlackBox Testing yang dilakukan oleh pengembang, karena dalam UAT pengguna sendiri yang menjadi penguji utama, sehingga hasilnya mencerminkan sejauh mana sistem diterima dari sudut pandang orang yang akan menggunakan sistem tersebut dalam kehidupan sehari-hari. Uji coba ini biasanya dilakukan dengan menggunakan kuesioner yang sudah disusun rapi, bertujuan untuk menilai seberapa mudah sistem itu dipakai serta sejauh mana pengguna merasa puas dengan sistem yang dibuat (Khusna et al., 2021).

2.4 Gap Penelitian

2.4.1 Summarize

Berbagai literatur akademis secara konsisten menegaskan bahwa penerapan arsitektur *You Only Look Once* serta model ekstraksi fitur berbasis *Deep Learning* menunjukkan performa yang sangat andal dalam tugas identifikasi personal dan mitigasi kecurangan (*anti-spoofing*). Studi terdahulu oleh (Faisal & Nurhaida, 2025; Satria et al., 2026) mengindikasikan bahwa model YOLO mampu membedakan antara wajah asli dan representasi tiruan dengan tingkat presisi serta *mean Average Precision* yang optimal pada lingkungan terkontrol. Di sisi lain, integrasi antara modul lokalisasi objek dengan model pengenalan wajah seperti ArcFace atau ResNet50 terbukti efektif untuk mengotomatisasi manajemen kehadiran berskala menengah hingga besar (Dony & Lubis, 2025). Selain itu, terdapat pergeseran tren implementasi sistem presensi menuju arsitektur berbasis *web* yang beroperasi secara *real-time* melalui kamera standar, guna menekan biaya operasional dari penggunaan perangkat keras biometrik terdedikasi (*dedicated*

hardware) (Indra et al., 2025; Tanujaya et al., 2026). Metodologi yang lazim digunakan dalam konteks ini melibatkan kombinasi *Convolutional Neural Network* dengan model ekstraksi metrik wajah berbasis jarak sudut (*angular margin*).

2.4.2 Synthesize

Berdasarkan hasil dari tinjauan pustaka pada penelitian-penelitian terdahulu, ditemukan beberapa kesamaan arsitektur dan fungsional yang dapat dihubungkan dengan penelitian ini. Berikut adalah tabel yang merangkum sintesis kesamaan antara penelitian terdahulu dengan sistem presensi yang diusulkan:

Tabel 2.2 Synthesize

Penelitian Terdahulu	Kesamaan dengan Penelitian Ini
Face Detection And <i>Anti-Spoofing</i> On Desktop Applications Using You Only Look Once (Faisal & Nurhaida, 2025)	Kedua penelitian sama-sama menggunakan arsitektur keluarga YOLO sebagai modul klasifikasi tekstur untuk mendeteksi <i>liveness</i> (keaslian wajah) guna menangkal serangan manipulasi identitas (<i>spoofing</i>) secara <i>real-time</i> .
Sistem Pengenalan Wajah Real-Time Menggunakan YOLOv7 untuk Akses Gedung TVRI Palembang Berbasis Web (Indra et al., 2025)	Kedua penelitian mengembangkan sistem presensi <i>real-time</i> berbasis antarmuka <i>web</i> yang murni memanfaatkan tangkapan kamera standar, tanpa memerlukan sensor fisik tambahan.
Sistem Presensi Mahasiswa Berbasis Pengenalan Wajah Real-Time dengan Deteksi <i>Anti-Spoofing</i> Menggunakan YOLOv8 dan ArcFace (Satria et al., 2026)	Kedua penelitian mengusulkan alur kerja hibrida, di mana algoritma YOLO bertugas memvalidasi <i>liveness</i> , sedangkan pustaka dari keluarga InsightFace (ArcFace) bertugas mengekstraksi wajah untuk

	pencocokan identitas di lingkungan akademik.
Deteksi YOLOv8 Dan Pengenalan Wajah Menggunakan RESNET50 Pada Gereja (Dony & Lubis, 2025)	Kedua penelitian memanfaatkan kapabilitas YOLO sebagai modul deteksi/lokalisasi wajah yang bekerja secara beriringan di depan model ekstraktor wajah utama sebelum divalidasi ke dalam basis data.
Pengembangan Sistem <i>Anti-Spoofing</i> Berbasis <i>Face Recognition</i> Menggunakan Arsitektur YOLOv8n (Tanujaya et al., 2026)	Kedua penelitian mengutamakan efisiensi pemrosesan berbasis <i>web</i> dengan cara memanfaatkan arsitektur varian <i>nano</i> (YOLOv8n pada peneliti terdahulu, dan YOLO11n pada penelitian ini) sebagai modul teringan untuk inferensi visual yang cepat.

2.4.3 Comparison

Studi mengenai integrasi pengenalan wajah dengan deteksi *liveness* telah berkembang pesat. namun, mayoritas literatur masih dihadapkan pada tantangan terkait arsitektur sistem dan efektivitas penanganan variasi serangan. Beberapa penelitian seperti (Tanujaya et al., 2026) cenderung bergantung pada metode deteksi *liveness* konvensional, seperti analisis *Eye Aspect Ratio*, yang rentan terhadap manipulasi serangan *replay* video resolusi tinggi. Di sisi lain, perangkat biometrik industri meskipun menawarkan ketahanan tinggi, sering kali terkendala oleh biaya implementasi yang signifikan serta kerentanan terhadap serangan media cetak. Selain itu, banyak implementasi YOLO dalam ranah akademis masih terbatas pada format aplikasi monolitik yang kurang mendukung pemisahan antara *Application Programming Interface* dan logika bisnis utama.

Penelitian ini bertujuan untuk mengisi celah (*gap*) tersebut dengan mengimplementasikan YOLO11n sebagai solusi keamanan berbasis perangkat lunak (*software-based security*) untuk analisis tekstur citra secara independen tanpa ketergantungan pada sensor fisik. Selain itu, arsitektur sistem yang diusulkan

mengadopsi pendekatan *decoupled*, yang memisahkan layanan kecerdasan buatan melalui Flask API berbasis REST (HTTP POST) dengan sistem manajemen bisnis berbasis Laravel. Pendekatan ini mengatasi batasan sistem presensi konvensional dengan memfasilitasi transmisi data visual berlatensi sangat rendah (*ultra-low latency*), sehingga proses pencocokan vektor menggunakan *Cosine Similarity* dapat dieksekusi secara instan dan tangguh.

2.4.4 Contribution

Penelitian ini memberikan kontribusi substansial dalam meningkatkan keamanan sistem informasi biometrik dengan mengusulkan solusi keamanan berbasis perangkat lunak yang memiliki dua tahap deteksi *liveness* (two-stage cascaded liveness verification), yang modern, efisien, dan sangat aman. Berbeda dengan penelitian sebelumnya yang lebih fokus pada pengujian model klasifikasi pasif secara terpusat, penelitian ini menggunakan arsitektur yang menggabungkan deteksi *liveness* berbasis tekstur menggunakan YOLO11n sebagai langkah pertama, yang kemudian dilanjutkan dengan mekanisme verifikasi aktif berbasis *challenge-response* yang menggunakan MediaPipe Face Landmarker sebagai langkah kedua yang bersifat decisive, sehingga mengisi celah keamanan yang ditemukan ketika kedua langkah tersebut bekerja secara terpisah.

Secara praktis, implementasi sistem presensi ini menjadi solusi atas tantangan administratif di Universitas Mercu Buana, terutama dalam memitigasi isu tingginya *turnover* serta keterbatasan digitalisasi bagi mahasiswa magang dan pekerja lepas. Dengan demikian, solusi ini menyediakan standar manajemen kehadiran modern yang transparan, mudah direplikasi, serta tangguh terhadap berbagai upaya manipulasi fisik maupun digital.