

BAB IV HASIL DAN PEMBAHASAN

4.1 Hasil Analisis Kebutuhan (*Requirement*)

Tahap pertama dalam metode Waterfall dilakukan dengan mengumpulkan dan menganalisis kebutuhan fungsional serta non-fungsional dari sistem tersebut. Tahap ini dilakukan berdasarkan hasil observasi dan wawancara langsung dengan Biro Sumber Daya Manusia (SDM) dan Biro Teknologi Informasi (BTI) di Universitas Mercu Buana.

Dari hasil wawancara, terungkap bahwa mesin biometrik konvensional rentan terhadap serangan pemalsuan wajah, seperti menggunakan foto cetak atau layar ponsel. Selain itu, belum ada sistem digital yang terorganisasi untuk mengelola peserta magang dan pekerja lepas. Berdasarkan temuan tersebut, kebutuhan sistem disusun dalam tabel berikut:

4.1.1 Analisis Kebutuhan Aktor

Sistem presensi online ini membagi hak otoritas menjadi empat tingkatan peran (role) struktural di dalam lingkungan Universitas Mercu Buana.

Tabel 4.1 Analisis Kebutuhan Aktor

Aktor / User	Deskripsi
SDM	Aktor tingkat manajemen utama yang bertanggung jawab mengelola struktur master data, mengatur periode presensi, memantau seluruh peserta magang/freelance, serta mengeksekusi pencetakan laporan presensi akhir.
Kabiro (Kepala Biro)	Aktor manajemen di tingkat Unit yang bertugas memantau seluruh peserta pada bagian-bagian di bawah naungannya, serta memberikan pengesahan (tanda tangan) rekapitulasi presensi pada setiap akhir periode bulan.
Kabag (Kepala Bagian)	Aktor pengawas teknis di tingkat Bagian yang bertugas memantau peserta di divisinya secara spesifik, memberikan penilaian kinerja, serta

	mengesahkan (tanda tangan) presensi harian peserta.
Peserta	Aktor operasional (mahasiswa magang atau <i>freelance</i>) yang berkewajiban mendaftarkan data biometrik wajah, melakukan absensi harian, melaporkan rincian kerja, serta mengelola data profil pribadinya.

4.1.2 Analisis Kebutuhan Fungsional

Kebutuhan fungsional mendefinisikan layanan, proses, atau fitur yang wajib disediakan oleh sistem agar aktor dapat menjalankan tugasnya sesuai dengan alur proses bisnis (*business workflow*).

Tabel 4.2 Fungsional

Aktor	Fungsionalitas	Deskripsi
Semua Aktor	Otentikasi & Keamanan Dasar	Sistem memfasilitasi semua pengguna untuk dapat masuk (<i>login</i>), mengubah kata sandi (<i>change password</i>), dan keluar dari sesi aplikasi (<i>logout</i>).
Semua Aktor kecuali Peserta	Peralihan Peran (Choose Role)	Sistem menyediakan fitur peralihan hak akses (<i>switch role</i>) bagi aktor manajemen yang memiliki jabatan atau otoritas ganda di dalam sistem.
SDM	Dashboard	SDM bisa melihat dan memantau data peserta

		magang dan freelance yang aktif
	Manajemen Data Master (CRUD)	Sistem memungkinkan aktor membuat, membaca, mengubah, dan menghapus data Unit, Bagian, Kabirol, Kabag, Tahun Akademik, Periode Bulan, dan data Peserta. Pembuatan entitas dibatasi oleh logika relasional (misal: Bagian hanya bisa dibuat jika Unit sudah ada).
	Monitoring Peserta & Cetak Rekap Absensi	Sistem menampilkan rekap absen seluruh peserta dan memungkinkan SDM untuk mencetak (Print) rekap data yang telah disetujui.
	Dashboard	Kabirol dapat melihat dan memantau peserta magang yang memiliki unit yang sama dengan kabirol tersebut.
Kabirol	Monitoring Peserta & Pengesahan Bulanan	Kabirol dapat memantau absen harian di setiap bagian pada unit tersebut, melihat detail data peserta, dan melakukan

		pengesahan di akhir periode bulan
Kabag	Dashboard	Kabag dapat melihat dan memantau semua peserta yang memiliki bagian yang sama dengan kabag tersebut
	Monitoring Peserta & Pengesahan Harian	Kabag dapat melihat daftar peserta pada divisinya, melihat detail data peserta, dan melakukan pengesahan presensi harian.
	Penilaian Peserta	Kabag dapat melakukan penilaian pada peserta divisinya.
Peserta	Dashboard	Peserta dapat melihat rekap presensi harian, card kehadiran, izin, telat, dan alpa.
	Profile	Peserta dapat mengubah Alamat dan No Telephone.
	Pendaftaran Wajah	Peserta diharuskan mendaftarkan wajahnya sebanyak 15 sample, yang di ekstraksi ke dalam database, guna memanfaatkan fitur presensi pengenalan wajah

	Presensi berbasis GPS dan <i>Face Recognition</i>	Sistem memvalidasi radius lokasi (GPS). Jika valid, sistem memicu mekanisme <i>active liveness</i> (MediaPipe) di sisi klien. Jika tantangan gerakan terlewati, pemindaian <i>passive liveness</i> (YOLO11n) dan <i>Face Recognition</i> dijalankan.
--	---	--

4.1.3 Analisis Kebutuhan Non Fungsional

Kebutuhan non-fungsional menetapkan batasan perilaku dan standar mutu (*quality attributes*) perangkat lunak, guna memastikan sistem berjalan aman, cepat, dan andal dalam lingkungan produksi.

Tabel 4.3 Non Fungsional

Kategori	Deskripsi Kebutuhan
Keamanan	1. Autentikasi ganda pada sisi klien dengan validasi titik koordinat pengguna (<i>Geofencing</i>) dan pemindaian kecerdasan buatan. 2. Lapisan keamanan <i>passive</i> dengan model <i>Anti-Spoofing</i> menggunakan YOLO11n dan <i>active</i> dengan MediaPipe yang memblokir keras upaya manipulasi menggunakan foto cetak atau tampilan layar digital (gawai).

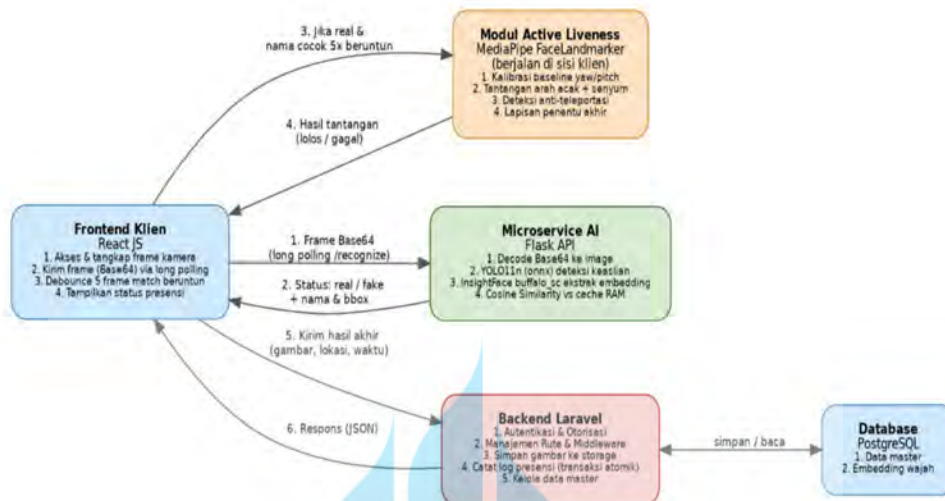
	3. Enkripsi kata sandi seluruh aktor menggunakan <i>hashing algorithms</i> bawaan Laravel (Bcrypt).
Kinerja & Latensi	Komunikasi <i>Long Polling</i> pengiriman data citra dari React JS ke Flask API harus memiliki waktu respons (<i>response time</i>) yang efisien (rata-rata latensi inferensi AI kurang lebih 500 milidetik per frame) guna menjamin pengalaman presensi <i>real-time</i> yang mulus (tidak <i>lagging</i>).
Ketersediaan	Arsitektur Virtual Machine pada server kampus memungkinkan web dan ai dapat berkomunikasi serta berjalan dengan baik.
Kebergunaan	Antarmuka pengguna (UI) dirancang dengan prinsip responsive web design, sehingga dapat diakses dengan berbagai device.
Interoperabilitas	Aplikasi berbasis antarmuka web murni, sehingga pengguna tidak memerlukan memasang perangkat didalam device nya, karena bisa langsung diakses melalui web browser.

4.2 Perancangan Sistem (*User Design*)

Berdasarkan hasil analisis kebutuhan, sistem dirancang dengan menggunakan model arsitektur berbasis objek, serta didukung oleh Unified Modeling Language (UML) dan perancangan struktur basis data.

4.2.1 Arsitektur Sistem Terdistribusi (*Decoupled Architecture*)

Sistem presensi dibuat terpisah menjadi tiga bagian utama agar tugas kerja server dapat dibagi secara merata.



Gambar 4.1 Arsitektur Sistem

Alur kerja pada Gambar 4.1 menunjukkan bahwa React JS bekerja untuk menangkap gambar dari kamera, lalu mengirim gambar tersebut dalam bentuk string Base64 melalui metode *long polling* ke API Flask. API Flask melakukan penguraian gambar, menjalankan model YOLO11n untuk deteksi keaslian (*real/fake*), mengekstrak embedding wajah menggunakan InsightFace, dan membandingkan identitas menggunakan pencarian di memori RAM. Hasil akhir berupa data dalam format JSON dikirim kembali ke aplikasi React JS, lalu diteruskan ke sistem backend berbasis Laravel untuk disimpan di database PostgreSQL.

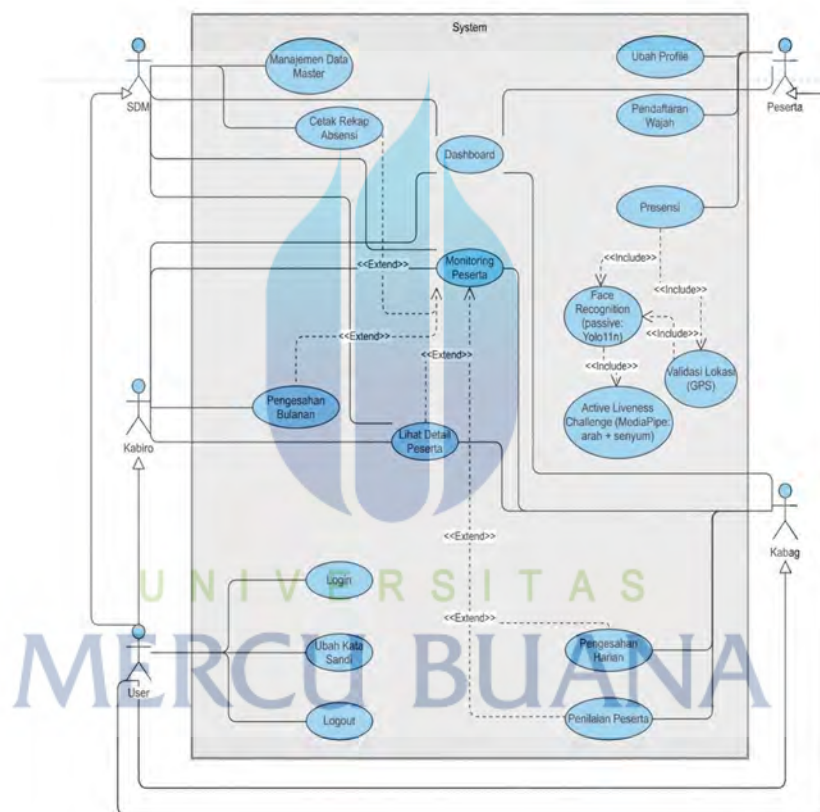
4.2.2 Pemodelan UML (*Unified Modeling Language*)

Sistem informasi presensi ini memiliki ekosistem fungsional yang komprehensif dengan empat tingkat otorisasi pengguna (SDM, Kabiro, Kabag, dan Peserta) untuk mendukung tata kelola birokrasi di Universitas Mercu Buana. Namun, guna menjaga fokus penelitian pada ranah Kecerdasan Buatan dan Rekayasa Perangkat Lunak, pemodelan sistem pada bab ini dibatasi dan difokuskan secara mendalam pada fungsionalitas inti, yaitu mekanisme registrasi biometrik

wajah, presensi *face recognition anti spoofing* dengan lapisan *active liveness*, serta alur hierarki persetujuan *approval workflow*.

1. Use Case

Use Case Diagram memetakan hubungan interaksi antara batasan sistem, aktor pengguna, dan fungsi-fungsi utama yang disediakan oleh aplikasi. Pemodelan ini memberikan gambaran visual mengenai hak akses dari masing-masing peran di dalam ekosistem aplikasi presensi.

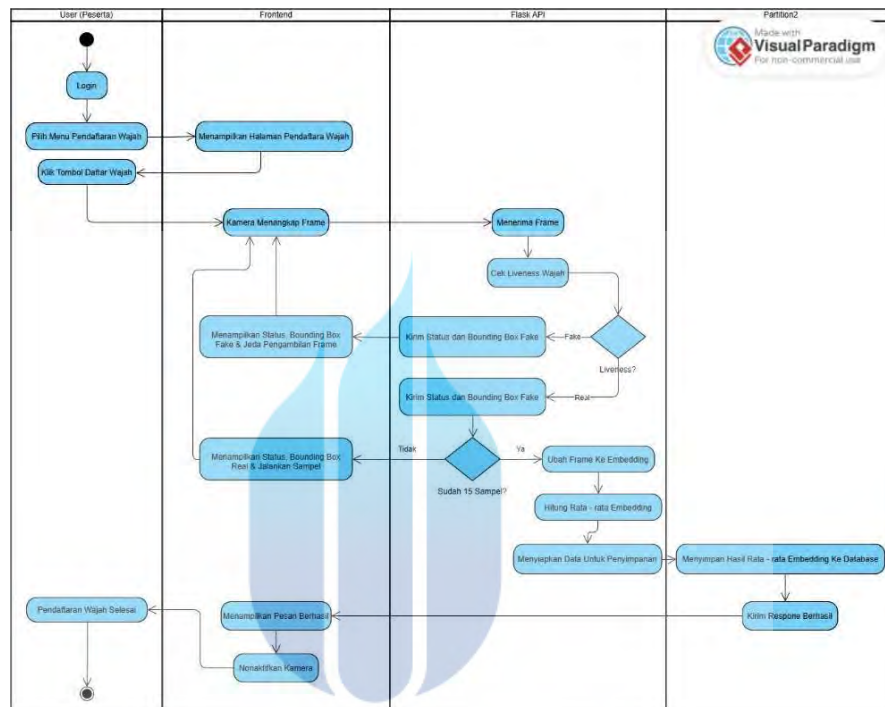


Gambar 4.2 *Use Case*

Gambar 4.2 menampilkan semua hak akses dan izin penggunaan fitur sistem, yang mengatur bahwa SDM mengelola data master, Kabag menyetujui kehadiran harian, Kabiro menyetujui kehadiran bulanan, dan Peserta melakukan pendaftaran serta pemindaian wajah.

2. Activity Diagram Pendaftaran Wajah

Activity Diagram pendaftaran wajah menunjukkan langkah – langkah proses Ketika peserta mendaftarkan wajah mereka untuk pertama kali sebagai data acuan identitas di dalam sistem. Alur aktivitas ini melibatkan interaksi *real time* antara React JS dan Flask API.



Gambar 4.3 Activity Diagram Pendaftaran Wajah

Gambar 4.3 menunjukkan proses pendaftaran wajah, di mana React JS terus-menerus mengambil gambar ke Flask API hingga model YOLO11n mengonfirmasi bahwa gambar tersebut benar-benar terdeteksi valid wajah manusia asli, kemudian model buffalo_sc mengekstrak sebanyak 15 sampel wajah dari gambar tersebut. Setelah nilai rata-rata vektor selesai dihitung, data tersebut disimpan ke dalam PostgreSQL dan juga memperbarui memori cache RAM pada server.

3. Activity Diagram Presensi Masuk dan Pulang

Diagram aktivitas ini menggambarkan urutan logika bisnis dan lapisan keamanan biometrik yang digunakan saat peserta melakukan kehadiran harian. Alur ini dibuat dengan ketat agar data yang masuk

[illegible]

55

Gambar 4.4 menunjukkan alur presensi harian yang ketat, yang dimulai dari validasi titik koordinat lokasi (GPS) pengguna oleh Laravel. Jika koordinat sudah benar, tombol "lanjut" ke halaman pengenalan wajah akan aktif. Peserta klik tombol "scan wajah" di halaman tersebut, lalu kamera React JS akan terbuka dan mengirim gambar ke Flask API untuk diuji oleh model YOLO11n. Jika gambar palsu, maka akan ditolak. Jika wajah dinyatakan asli dan nama cocok, lanjut ke tantangan *Active Liveness*. Selain itu, tingkat kemiripan wajah akan dihitung menggunakan algoritma *Cosine Similarity* dengan ambang batas ≥ 0.5 . Setelah semua lolos, frame akhir ditangkap dan dikirim ke backend Laravel untuk disimpan sebagai data presensi.

4. Activity Diagram Alur Persetujuan

Diagram aktivitas ini menunjukkan tata kelola birokrasi berjenjang terhadap data kehadiran yang telah disubmit oleh peserta magang. Diagram ini menggambarkan bagaimana data divalidasi dari tingkat bawah hingga siap cetak di tingkat manajemen.



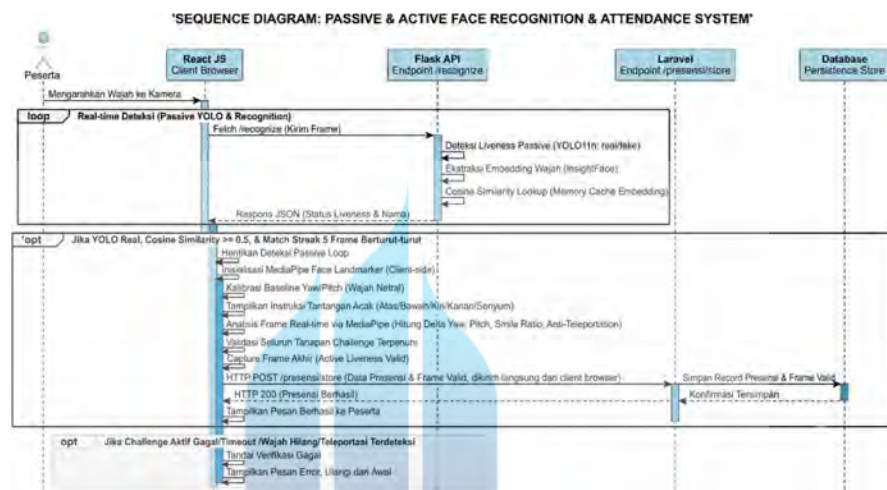
Gambar 4. 5 Activity Diagram Alur Persetujuan

Gambar 4.5 menjelaskan proses birokrasi berjenjang pasca-absensi, di mana data kehadiran harian muncul secara langsung di dashboard Kabag untuk ditandatangani, dan penilaian peserta biasanya dilakukan selama 2 minggu sekali. Pada akhir bulan, laporan dikompilasi menuju dashboard

Kabiro agar dapat ditandatangani, sehingga staf SDM mendapatkan izin lengkap untuk mengunduh serta mencetak laporan resmi dalam bentuk fisik.

5. Sequence Diagram Transmisi Presensi Face Recognition

Sequence Diagram menggambarkan interaksi berorientasi pesan antar komponen arsitektur sistem (React JS, Flask, Laravel, dan PostgreSQL) berdasarkan urutan waktu operasional run-time.

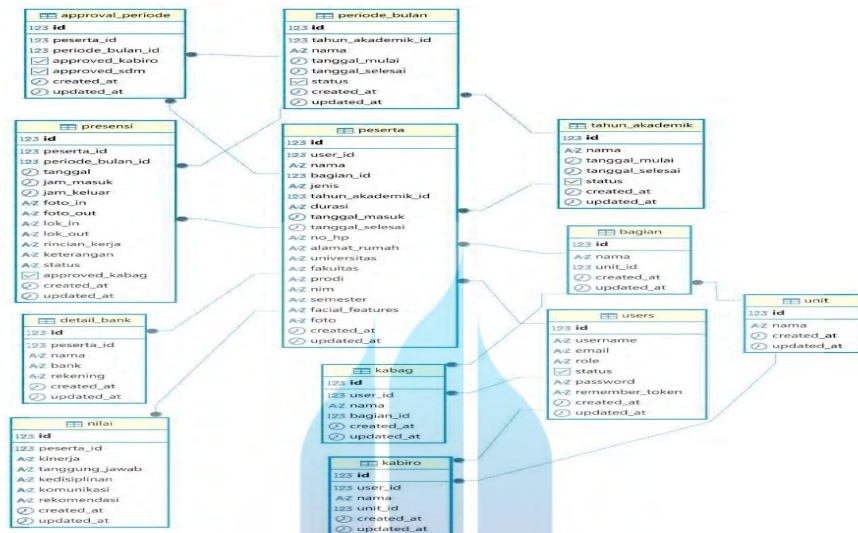


Gambar 4.6 Sequence Diagram

Gambar 4.6 menunjukkan *sequence diagram* interaksi sistem yang terbagi dalam dua tahap verifikasi. Tahap pertama, React JS secara berulang mengirimkan frame ke Flask API (/recognize) untuk deteksi *passive liveness* menggunakan YOLO11n, ekstraksi embedding dengan InsightFace, dan pencocokan identitas (Cosine Similarity). Jika status wajah dinilai *real* dengan kemiripan $\geq 0,5$ selama lima frame berturut-turut, sistem masuk ke tahap kedua (*active liveness*). Pada tahap ini, React JS menjalankan MediaPipe Face Landmarker secara *client-side* untuk kalibrasi wajah netral dan memberikan tantangan gerak acak (menoleh atau tersenyum). Sistem mengecek sudut wajah, rasio senyum, serta *anti-teleportation* secara *real-time*. Jika berhasil, React JS menangkap frame akhir dan langsung mengirimkan data presensi ke Laravel (/presensi/store) untuk disimpan ke basis data dengan respons HTTP 200. Namun, jika tantangan gagal dipenuhi (karena *timeout*, wajah hilang, atau anomali posisi), sistem akan menampilkan pesan *error* dan meminta peserta mengulangi proses dari awal.

4.2.3 Perancangan Basis Data (*Entity Relationship Diagram*)

Sistem terhubung dengan database relasional PostgreSQL melalui desain ERD agar data biometrik tetap terhubung dengan baik.



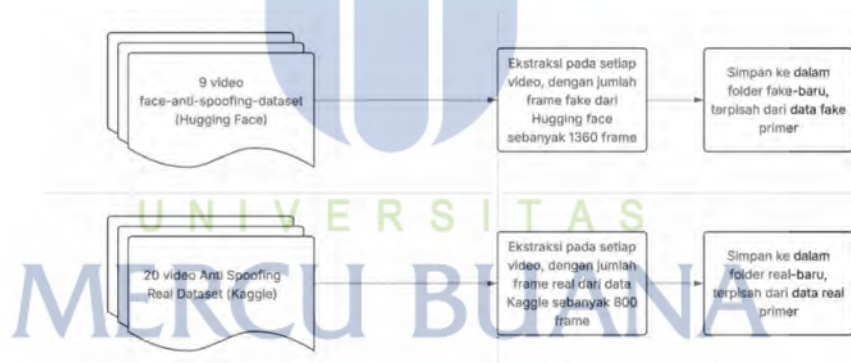
Gambar 4.7 *Entity Relationship Diagram*

Berdasarkan Gambar 4.7, entitas peserta memiliki hubungan One-to-Many dengan entitas presensi. Entitas peserta dirancang memiliki kolom khusus bernama *facial_features* dengan tipe data teks berupa JSON string, yang memiliki dimensi tinggi untuk menyimpan deret array vektor wajah yang diperoleh dari perhitungan AI.

4.2.4 Perancangan Prapemrosesan *Dataset AI*

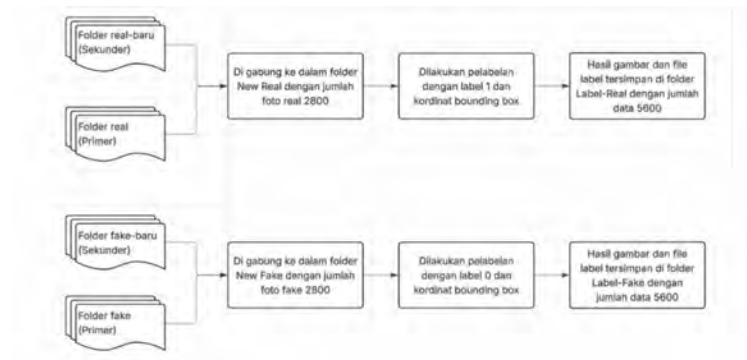
Sebelum membangun layanan kecerdasan buatan, dirancang jalur pemrosesan data untuk melatih model *object detection* Anti Spoofing menggunakan YOLO11n, Pengumpulan data dilakukan melalui penggabungan data primer dan data skunder, penggabungan ini berguna untuk meningkatkan variabilitas dan ketahanan model:

1. Data primer dikumpulkan secara langsung dari 7 partisipan menggunakan perangkat *webcam*, menghasilkan 2.000 citra kelas *real* (manusia asli) dan 1.440 citra kelas *fake* (wajah foto cetak/layar digital).
2. Pengumpulan Data Sekunder menggunakan *dataset* public *Anti-Spoofing Real Dataset* dari Kaggle serta *face-anti-spoofing-dataset* , penambahan data sekunder ini digunakan supaya model hasil pelatihan bisa meningkatkan variabilitas dan ketahanan model dalam mengenali object asli dan palsu.
 - Link *dataset* Kaggle:
<https://www.kaggle.com/datasets/trainingdatapro/anti-spoofing-live>
 - Link *dataset* Hugging Face:
[AxonData/face-anti-spoofing-dataset · Datasets at Hugging Face](https://huggingface.co/datasets/AxonData/face-anti-spoofing-dataset)
3. Ekstraksi Frame dan Pelabelan: Karena data sekunder yang di dapat berupa rekaman video, maka perlu mengekstraksi bingkai video menjadi gambar statis.



Gambar 4.8 Ekstraksi Frame

Dari Gambar 4.8 menjelaskan video yang di ambil dari sumber kaggle dan hugging face diekstrak dan jumlah frame ekstaknya disesuaikan dengan data primer.



Gambar 4.9 Pelabelan

Pada Gambar 4.9 yang merupakan tahap pelabelan data, yang dimana folder primer (*real* dan *fake*) dan folder sekunder (*real* dan *fake*) dijadikan satu folder yaitu folder New Real untuk keseluruhan data *real*, dan New Fake untuk keseluruhan data *fake*. Tahap selanjutnya adalah melabelkan data *real* dengan nilai 1 beserta *bounding box* yang sudah ditentukan, dan nilai 0 untuk data *fake*, keduanya tersimpan di folder berbeda setelah proses pelabelan yaitu Label-Real untuk data *real* dan Label-Fake untuk data *fake*.

4. Pembagian *Dataset*



Gambar 4.10 Pembagian *Dataset*

Terlihat dari Gambar 4.10 Seluruh citra digabungkan dalam satu folder bernama All data, kemudian didistribusikan secara acak ke dalam tiga sub-kategori dengan rasio 70% Data Latih (*Train*), 20% Data Validasi (*Val*), dan 10% Data Uji (*Test*). Manifest arsitektur folder ini dikonfigurasi pada file data.yaml yang tersimpan di dalam folder Splitdata untuk selanjutnya dikompresi menjadi format .zip dan diunggah ke platform Ultralytics.

4.3 Implementasi Sistem (*Implementation*)

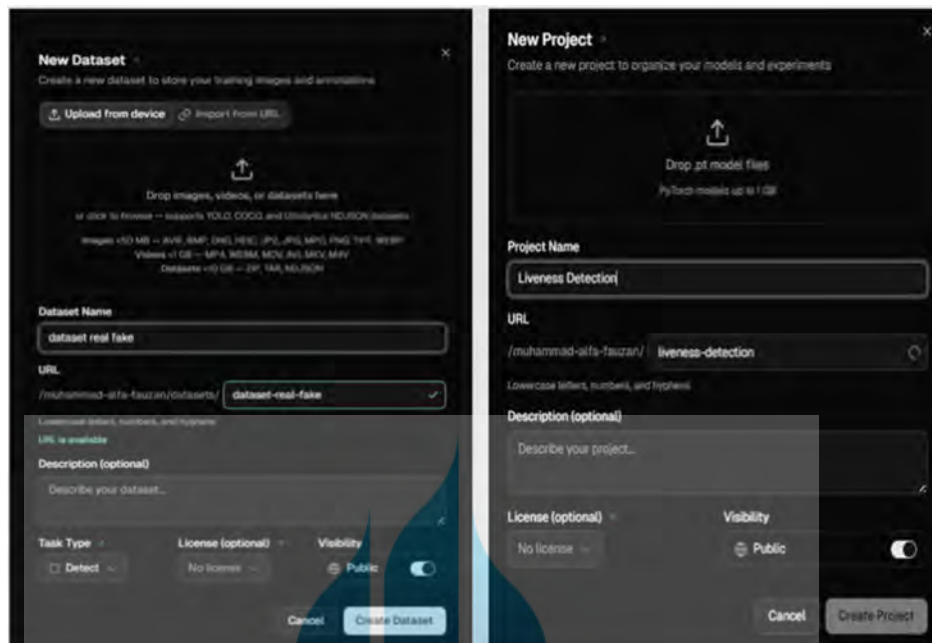
Tahap implementasi adalah proses mengubah hasil perancangan menjadi baris kode program yang bisa dipahami oleh komputer. Proses ini melibatkan membuat model pembelajaran mesin, menyusun modul aplikasi, dan mengatur server

4.3.1 Implementasi Pelatihan Model YOLO11n

Pada tahap ini, pelatihan model kecerdasan buatan dilakukan dengan menggunakan arsitektur YOLO11n versi nano. Tujuan utama dari tahap ini adalah memberi kemampuan pada model untuk mengenali, mengekstrak, dan membedakan ciri-ciri tekstur visual antara wajah manusia asli dengan benda-benda palsu seperti foto cetak di kertas atau tampilan layar gawai. Proses ini sangat penting karena akurasi bobot model yang dihasilkan menjadi dasar utama (lapisan pertahanan pertama) dalam mencegah serangan spoofing pada sistem kehadiran.

Seluruh proses pelatihan komputasi dilakukan menggunakan infrastruktur cloud computing yang dimiliki oleh platform Ultralytics HUB. Pendekatan ini dipilih agar waktu digunakan dengan lebih efisien dan penggunaan sumber daya perangkat keras lokal peneliti tetap terkontrol dengan baik. Selama proses pelatihan berlangsung, model belajar secara perlahan dengan memperbarui bobot parameter berdasarkan berbagai gambar yang diberikan. Pelatihan ini dijalankan secara terstruktur dalam tiga tahap utama, yaitu: persiapan *dataset*, pengaturan parameter, pengawasan metrik selama proses pelatihan, serta ekspor model ke dalam onnx.

1. Pembuatan Project dan Upload *Dataset*

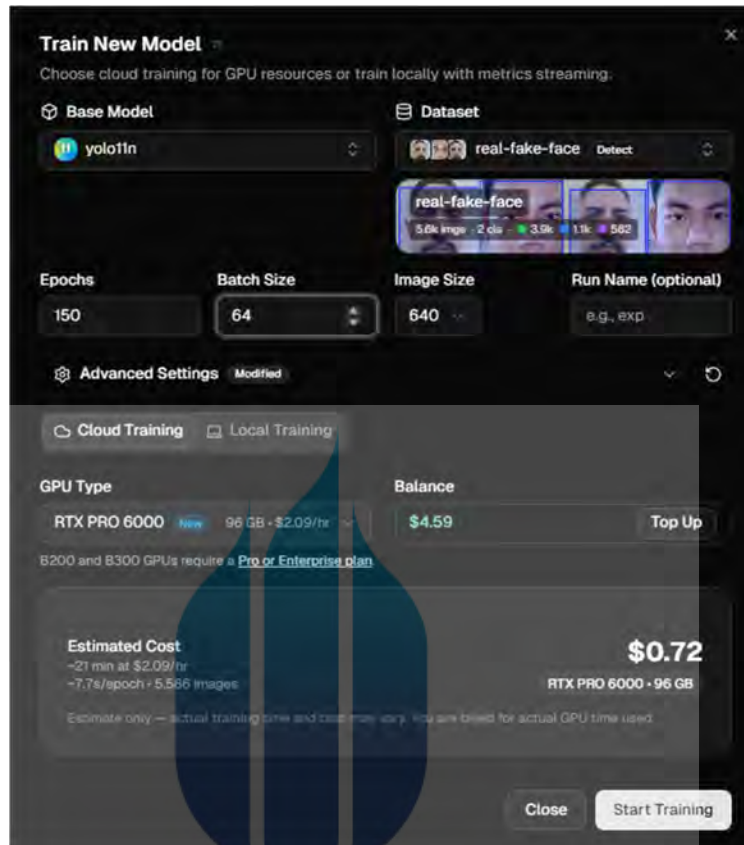


Gambar 4.11 Pembuatan Project dan Upload *Dataset*

Langkah pertama dimulai dengan pembuatan project di platform Ultralytics HUB . Setelah repositori proyek sudah siap, *dataset* deteksi keaslian dalam format berkas .zip yang sudah diproses sebelumnya diunggah ke dalam sistem. Platform kemudian memeriksa struktur direktori, apakah file anotasi *bounding box* sesuai, serta apakah kelas objek valid 0 untuk kelas *fake*, 1 untuk kelas *real*.

Berdasarkan penggabungan data primer dan sekunder, diperoleh *dataset* yang lengkap dan terpadu sebanyak 5.600 gambar, terdiri dari 2.800 gambar kategori asli dan 2.800 gambar kategori palsu. Data tersebut didistribusikan secara otomatis sesuai dengan konfigurasi dalam file data.yaml dengan perbandingan 70:20:10. Pembagian ini menghasilkan 3.920 gambar yang digunakan sebagai data latihan, 1.120 gambar untuk data validasi, dan 560 gambar yang dipakai sebagai data uji. Validasi proporsi ini penting agar model memiliki sampel pembelajaran yang seimbang dan mencegah model terlalu cocok dengan data pelatihan.

2. Konfigurasi Arsitektur dan Hyperparameter



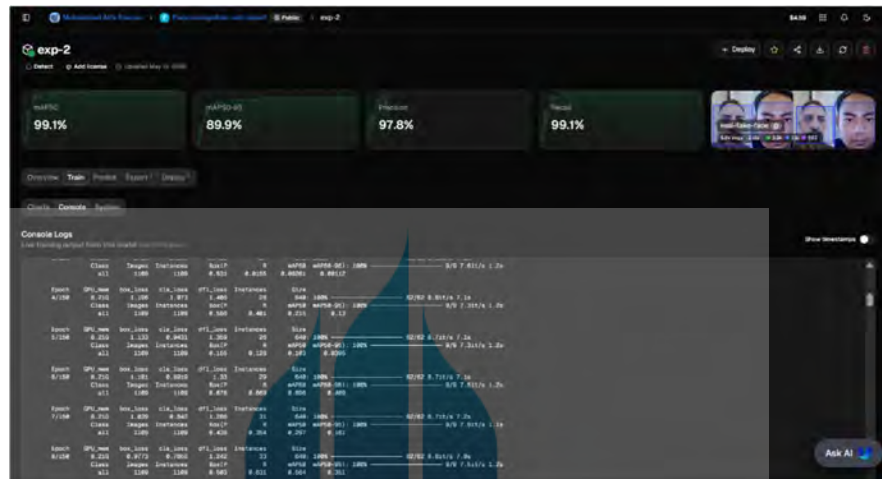
Gambar 4.12 Konfigurasi Arsitektur dan Hyperparameter

Setelah *dataset* berhasil diintegrasikan, proses selanjutnya adalah menentukan konfigurasi parameter pelatihan. Penelitian ini menggunakan arsitektur YOLO11n sebagai dasar modelnya. Untuk mendapatkan hasil deteksi manipulasi 2D yang kuat, parameter pelatihan disesuaikan secara rinci, jumlah iterasi pembelajaran ditentukan sebanyak 150 *epoch*, ukuran kelompok data (*batch size*) diatur menjadi 64, serta resolusi ukuran gambar masukan (*image size*) dipertahankan pada 640 piksel. Selain itu, fungsi optimasi diatur dengan metode Adaptive Moment Estimation (AdamW).

Untuk memperkuat kemampuan model dalam menghadapi berbagai jenis serangan spoofing yang baru, selama proses pelatihan digunakan teknik augmentasi data, yaitu dengan menerapkan Mosaic dengan skala 1.0 dan campuran piksel Mixup dengan skala 0.1. Seluruh tugas pengaturan komputasi yang kompleks tersebut dijalankan menggunakan Virtual

Dedicated GPU yang ditawarkan oleh layanan cloud Ultralytics. Ini mempercepat proses ekstraksi fitur dibandingkan dengan menggunakan komputasi lokal, sehingga model bisa memahami sifat data secara maksimal dalam setiap siklus *epoch*.

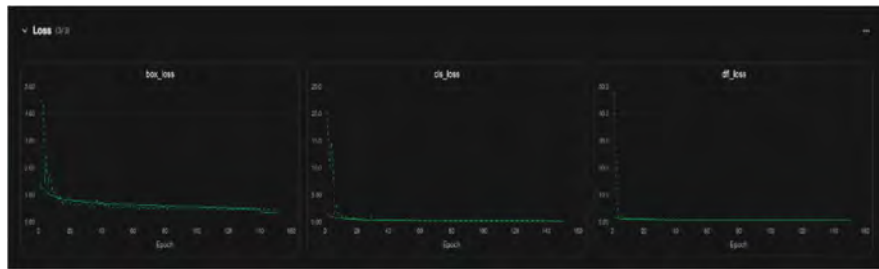
3. Eksekusi Pelatihan dan Pemantauan Metrik



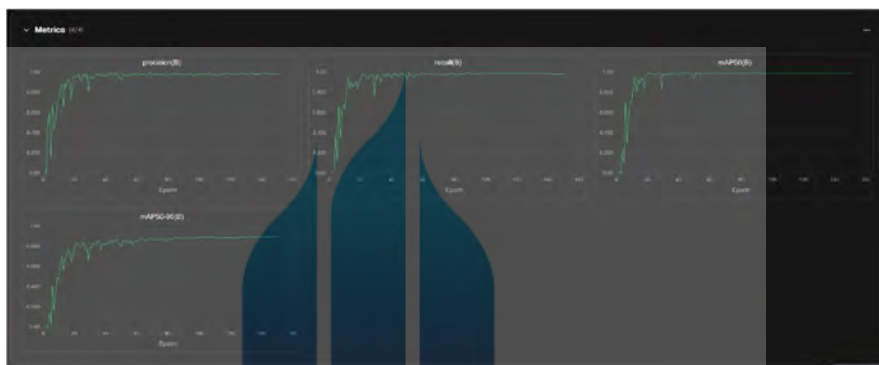
Gambar 4.13 Proses Pelatihan dan Pemantauan Metrik

Setelah semua hyperparameter ditetapkan, proses pelatihan dimulai melalui dashboard Ultralytics HUB. Platform ini memiliki antarmuka pemantauan yang memperlihatkan kemajuan pembelajaran secara langsung dan terus-menerus. Pada setiap kali *epoch* berubah, model akan menyesuaikan dan memperbarui bobotnya berdasarkan evaluasi nilai kesalahan.

Parameter evaluasi yang diperiksa secara langsung mencakup box



Gambar 4.14 Grafik *Loss*

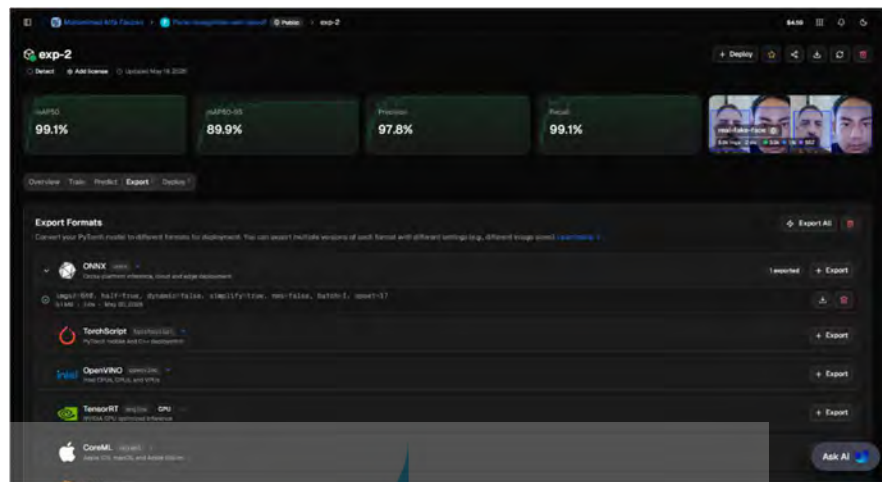


Gambar 4.15 Grafik *Metrics*

Berdasarkan Gambar 4.14, ketiga parameter *loss* (*box*, *cls*, *dfl*) menunjukkan penurunan yang sangat cepat di awal proses iterasi dan terus berkurang secara stabil hingga mencapai tingkat kestabilan terendah di akhir iterasi. Penurunan kehilangan klasifikasi yang halus menunjukkan bahwa model tidak bingung dalam membedakan wajah manusia asli dan wajah manipulasi 2D.

Hal ini didukung oleh Gambar 4.15, di mana kurva metrik Precision, Recall, dan mAP50 meningkat drastis pada 20 *epoch* pertama dan kemudian tetap stabil pada tingkat maksimal (hampir mencapai 1.0). Konvergensi kedua grafik ini menunjukkan bahwa model secara perlahan menjadi lebih pintar dalam mengenali anomali pada tekstur wajah yang dimanipulasi, tanpa terlalu menghafal data atau gagal mempelajari pola yang ada.

4. Ekspor Model Ke Dalam ONNX



Gambar 4.16 Eksport ONNX

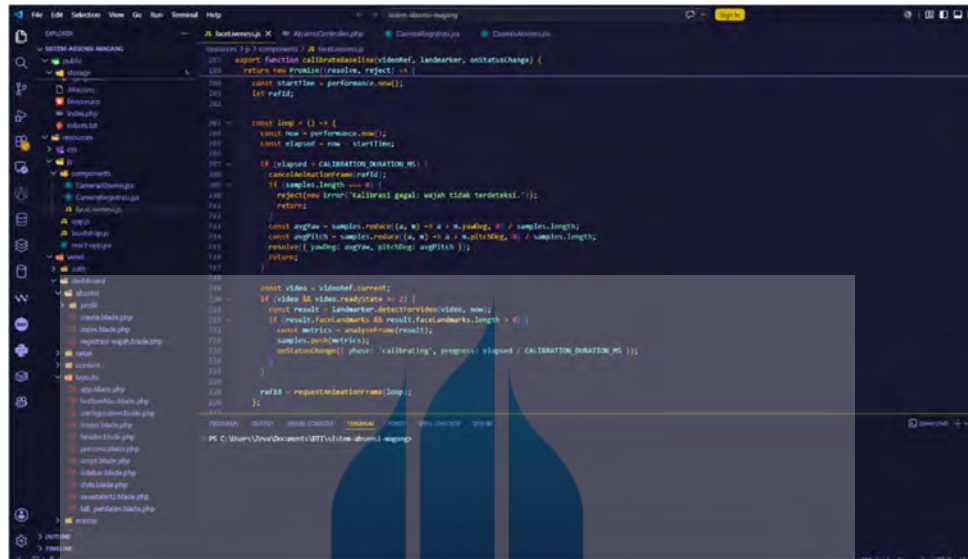
Setelah 150 *epoch* selesai, model menghasilkan bobot terbaik yaitu exp-2.pt, kemudian dikonversi ke dalam format Open Neural Network Exchange yaitu exp-2.onnx. Format hasil ekspor ini disiapkan agar model bisa dijalankan dengan ringan dan murni menggunakan CPU pada tahapan API microservice. Hasil pelatihan kemudian dievaluasi secara matematis dalam tahap pengujian menggunakan alat *Confusion Matrix* yang mencakup indikator Akurasi, Presisi, Recall, dan mAP50.

4.3.2 Library dan Konfigurasi Server AI

Layanan pemrosesan kecerdasan buatan dalam sistem ini dibuat dengan pendekatan terdistribusi yang melibatkan bagian klien dan bagian server. Di sisi klien, sistem memuat library @mediapipe/tasks-vision yang dikelola melalui modul faceLiveness.js untuk menjalankan model Face Landmarker. Model ini dijalankan secara lokal sekali saja (preload) ketika komponen kamera presensi dimulai. Untuk meningkatkan keakuratan dalam mendeteksi gerakan kepala, parameter output FacialTransformationMatrixes diaktifkan. Pendekatan ini memungkinkan sistem menghitung sudut putar kepala (yaw) dan gerakan naik/turun (pitch) secara langsung dari turunan matriks transformasi wajah, bukan hanya mengandalkan perbandingan posisi landmark 2D. Metode matriks transformasi terbukti lebih stabil dalam komputasi, terutama saat mendeteksi

gerakan pitch yang sering kali sulit terdeteksi dengan baik jika hanya menggunakan perhitungan rasio 2D biasa.

Sementara itu, layanan pengolahan di sisi server dikembangkan dengan menggunakan framework mikro Flask yang berbasis bahasa pemrograman Python.



Gambar 4.17 Workspace VS code Laravel

Untuk menjaga struktur kode tetap rapi dan mudah dipelihara, logika kecerdasan buatan dibagi menjadi dua file utama: `app.py` yang bertugas sebagai pusat pengontrolan Antarmuka Pemrograman Aplikasi (API), serta `face_rec.py` yang berfungsi sebagai modul perhitungan jarak vektor wajah.



Gambar 4.18 Workspace VS code

Kode server ini dikonfigurasi untuk memuat pustaka `ONNXRuntime` (untuk klasifikasi liveness) serta model `buffalo_sc` dari pustaka `InsightFace` (untuk

ekstraksi *embedding* berdimensi 512). Pencocokan vektor dilakukan menggunakan rumus *Cosine Similarity* dari pustaka Scikit-learn.

4.3.3 Implementasi Kode Program Utama (*Source Code*)

Implementasi kode sangat penting karena menentukan efisiensi komputasi, kelancaran latensi, dan keamanan integritas data, dan berikut penjelasannya:

1. Kode Pemuatan Cache Vektor Wajah (Flask API - Python)

Untuk menghindari beban *Input/Output* yang terlalu berat pada database fisik, API Flask dikonfigurasi dengan menggunakan pustaka *psycpg2* agar semua vektor wajah dimuat langsung ke dalam memori RAM melalui *Pandas DataFrame* ketika layanan dijalankan pertama kalinya.



Gambar 4.19 Kode Pemuatan Cache Vektor Wajah

2. Kode Antarmuka Transmisi Asinkron (React JS)

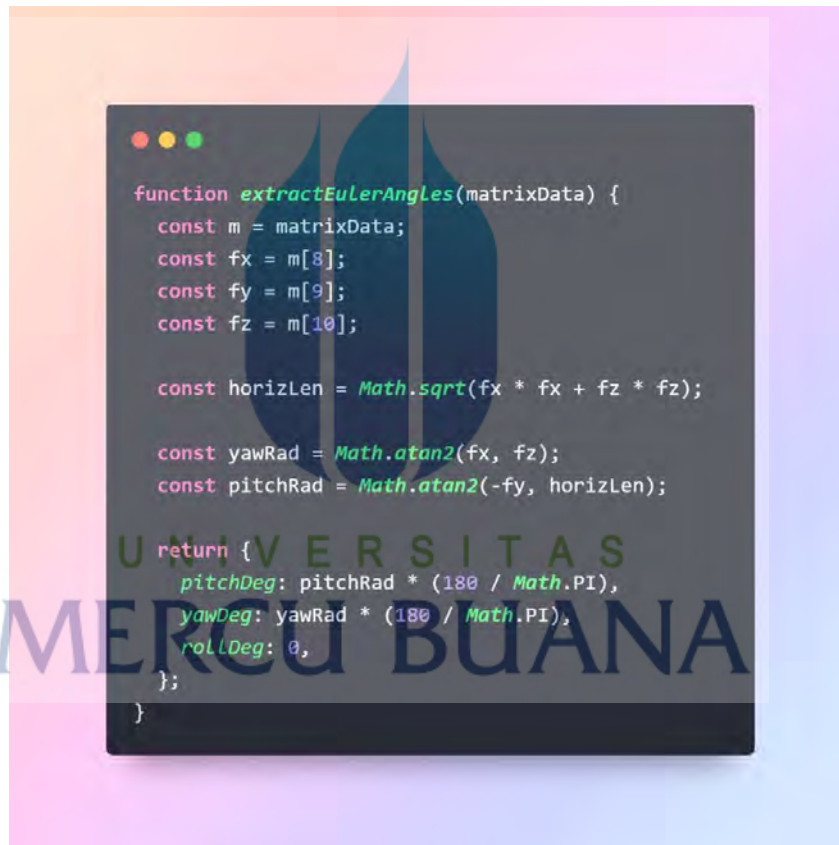
Logika JavaScript ini diterapkan pada komponen React JS untuk menggerakkan proses pengiriman gambar ke API Flask. Implementasi *await fetch* memastikan proses pengiriman dilakukan secara perlahan satu persatu (*long polling*) dan juga mengukur waktu yang dibutuhkan untuk menjalankan eksekusi AI



Gambar 4.20 Kode Antarmuka Transmisi Asinkron (React JS)

3. Kode Modul *Active Liveness Challenge*

Ekstraksi sudut kepala dilakukan dengan metode forward vector, yaitu dengan mengambil kolom ketiga dari sub-matriks rotasi pada *facial transformation matrix* (elemen dengan indeks 8, 9, dan 10 dalam representasi column-major), kemudian menghitung sudut yaw dan pitch berdasarkan arah vektor tersebut menggunakan fungsi `atan2`. Metode ini dipilih karena lebih stabil dibandingkan metode dekomposisi Euler langsung dari rasio landmark 2D.



Gambar 4.21 Ekstraksi Sudut Yaw/Pitch dari *Facial Transformation Matrix*

Sebelum tantangan dimulai, sistem melakukan kalibrasi dasar selama 800 milidetik dengan mengambil rata-rata dari beberapa sampel sudut wajah saat peserta tidak bergerak dan dalam posisi netral. Setiap langkah tantangan berikutnya diukur berdasarkan perbedaan dari baseline ini, sehingga batas arah (15 derajat untuk yaw dan 10 derajat untuk pitch) lebih stabil terhadap

perbedaan perangkat, posisi duduk, dan sudut kamera dibandingkan dengan batas absolut. Urutan empat arah gerakan diacak menggunakan algoritma Fisher–Yates agar tidak ada dua sesi presensi yang memiliki urutan instruksi yang sama, sebagai cara mengurangi risiko serangan berupa video yang direkam sebelumnya.



Gambar 4.22 Pengacakan Urutan Tantangan Arah

Selama setiap langkah tantangan berlangsung (batas waktu 5 detik), sistem membandingkan posisi hidung antar-frame terhadap lebar wajah untuk mendeteksi lompatan posisi yang tidak wajar (anti-teleportasi), yang mengindikasikan kemungkinan foto/video disodorkan mendadak ke depan kamera.



Gambar 4.23 Pemeriksaan Anti-Teleportasi Posisi Wajah

Fungsi orkestrator `runLivenessChallenge()` menjalankan semua sesi secara berurutan, yaitu kalibrasi baseline, kemudian empat tantangan arah yang acak, dan akhirnya satu tantangan senyum yang didasarkan pada rasio jarak sudut mulut terhadap lebar wajah. Sesi dianggap gagal (kembali false) jika terjadi timeout di langkah apa pun, ada pergerakan posisi yang tidak normal, atau kalibrasi awal gagal karena wajah tidak terdeteksi.

4. Kode Integritas Transaksi Relasional (Laravel Backend - PHP)

Fitur `DB::beginTransaction()` diimplementasikan pada controller . Perintah ini berfungsi sebagai metode transaksi atomik untuk melindungi database dari masuknya data kehadiran yang rusak karena gangguan jaringan.

```
DB::beginTransaction();
try {
    if($cek > 0) {
        // UPDATE Absen Pulang
        $presensi = Presensi::where('tanggal', $tanggal)->where('peserta_id',
        $peserta->id)->first();
        if ($presensi->jam_keluar) return "error|Anda sudah absen pulang hari ini";

        if (empty(trim($ rincian_kerja))) {
            return "error|Rincian kerja wajib diisi sebelum absen pulang.";
        }

        $fileName = null;
        if ($image_base64) {
            $folderPath = "public/uploads/absensi/pulang/";
            $fileName = $peserta->nia . "_" . $jam_file . "_" . $tanggal . ".
            pulang.png";
            Storage::put($folderPath . $fileName, $image_base64);
        }

        $presensi->update([
            'jam_keluar' => $jam, 'foto_out' => $fileName, 'lok_out' => $lokasi,
            'rincian_kerja' => $rincian_kerja,
        ]);

        DB::commit();
        return "success|Terimakasih, Sampai Jumpa!";
    } else {
        if ($jam < $batas_jam_masuk) {
            return "error|Maaf, batas waktu absensi masuk (12:00 WIB) telah habis.
            Silakan hubungi Staff Admin BTL.";
        }

        // INSERT Absen Masuk
        $fileName = null;
        if ($image_base64) {
            $folderPath = "public/uploads/absensi/masuk/";
            $fileName = $peserta->nia . "_" . $jam_file . "_" . $tanggal . ".
            masuk.png";
            Storage::put($folderPath . $fileName, $image_base64);
        }

        Presensi::create([
            'peserta_id' => $peserta->id, 'periode_bulan_id' => $periode->id,
            'tanggal' => $tanggal,
            'jam_masuk' => $jam, 'foto_in' => $fileName, 'lok_in' => $lokasi, 'status'
            => 'hadir',
        ]);

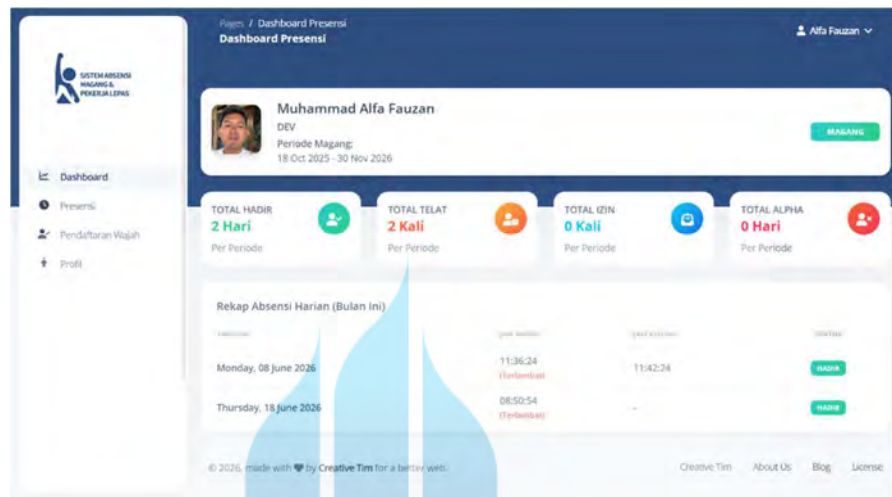
        DB::commit();
        Log::info("[INTEGRITAS DATA AMAN] Presensi ID Peserta: {$request->peserta_id}
        berhasil disiapkan utuh.");
        return "success|Terimakasih, Selamat Bekerja!";
    }
} catch (Exception $e) {
    DB::rollBack();
    Log::error($e);
    return "error|Terjadi kesalahan sistem.";
}
```

Gambar 4.24 Kode Integritas Transaksi Relasional

4.3.4 Implementasi Antarmuka Pengguna

Antarmuka pengguna diimplementasikan menggunakan perpaduan *framework* antarmuka Laravel Blade dan *library* React JS untuk manajemen kamera yang dinamis.

1. Halaman Dashboard Presensi

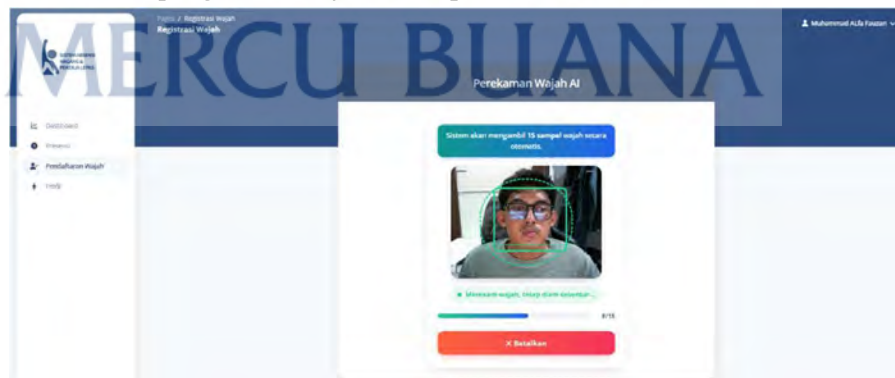


Gambar 4.25 Dashboard Peserta

Halaman ini dikembangkan menggunakan Laravel untuk menampilkan ringkasan data.

2. Halaman Registrasi Wajah

Peserta baru diwajibkan merekam 15 sampel wajah. Supaya bisa melakukan pengenalan wajah untuk presensi.



Gambar 4.26 Halaman Pendaftaran Wajah

3. Halaman Presensi (*Passive* dan *Active Liveness*)

Terdapat dua lapisan saat presensi wajah, yaitu lapisan *passive liveness* (Yolo11n) dan *active liveness* (MediaPipe).

- *Passive Liveness*

Lapisan pertama mendeteksi dua kelas secara *real-time*, yaitu asli dengan *bouding box* hijau, dan palsu berwarna merah. Serta pendeteksian bukan pemilik akun



Gambar 4.27 Terdeteksi Wajah Asli



Gambar 4.28 Terdeteksi Wajah Palsu



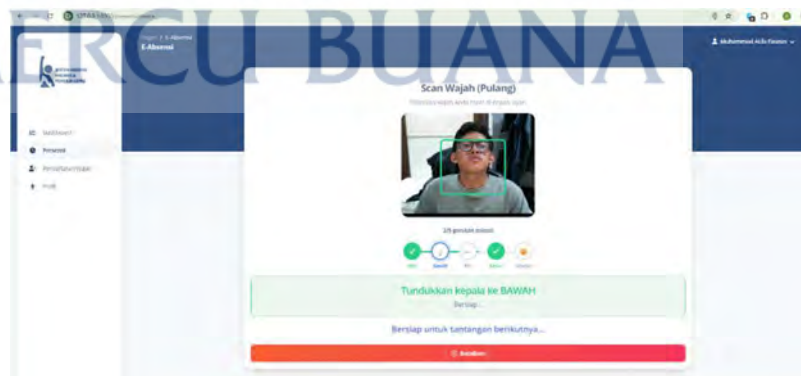
Gambar 4.29 Wajah Bukan Pemilik Akun

- *Active Liveness*

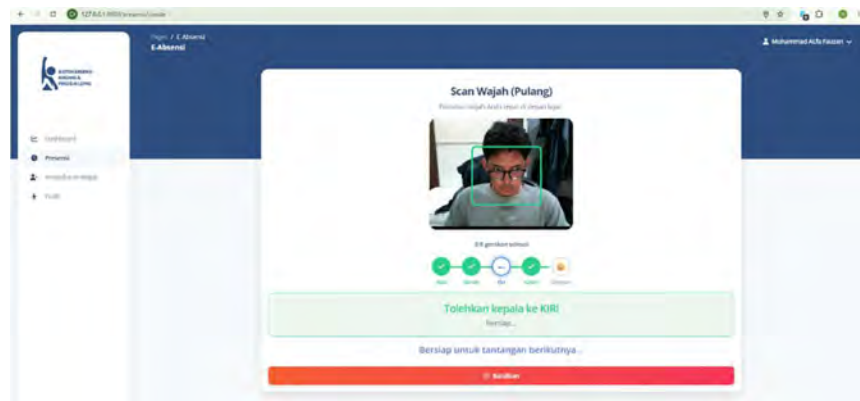
Setelah identitas peserta terverifikasi pada lima bingkai secara berurutan, antarmuka presensi otomatis berpindah menampilkan komponen animasi stepper lima langkah. Komponen ini terdiri dari instruksi visual berupa ikon panah yang digunakan untuk membimbing arah gerakan kepala pengguna, seperti atas, bawah, kiri, dan kanan, serta satu ikon khusus yang digunakan untuk memberi petunjuk agar pengguna tersenyum. Sistem memberikan perintah gerakan yang spesifik secara acak, dilengkapi dengan hitung mundur 5 detik untuk setiap tantangannya. Selama proses berlangsung, antarmuka menampilkan indikator progres yang akan terisi perlahan setiap kali satu gerakan diketahui valid oleh sistem.



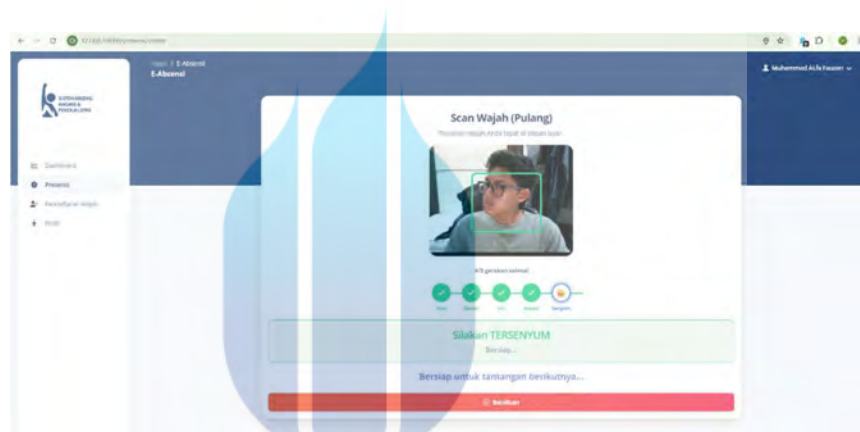
Gambar 4.30 Tantangan wajah ke kanan



Gambar 4.31 Tantangan wajah ke atas



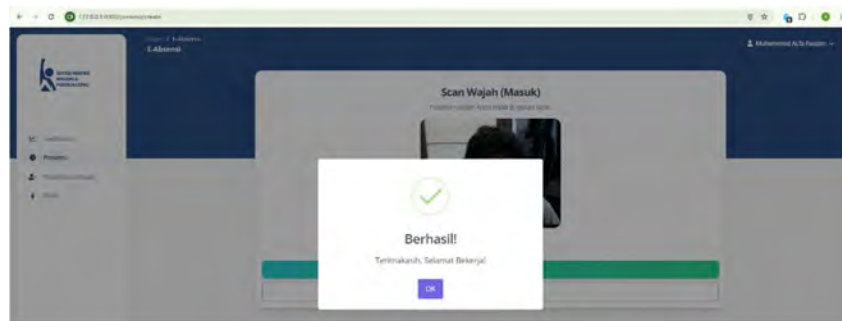
Gambar 4.32 Tantangan wajah ke bawah



Gambar 4.33 Tantangan wajah ke kiri



Gambar 4.34 Tantangan wajah Tersenyum



Gambar 4. 35 Presensi Berhasil

4.4 Pengujian Sistem (*Testing*)

Tahap pengujian adalah proses evaluasi apakah sistem tersebut layak digunakan secara luas. Uji coba dilakukan langsung di lingkungan kerja server Universitas Mercu Buana untuk mengukur tingkat keakuratan kecerdasan buatan, kemampuan bekerja dengan logika bisnis antarmuka, serta ketahanan dalam mengirim data.

4.4.1 Pengujian Kinerja Model AI

Kinerja klasifikasi arsitektur YOLO11n diperiksa secara mendalam dengan menggunakan 10% dari *dataset* uji. Matriks evaluasi ini menunjukkan seberapa baik model bisa mengenali dan menolak gambar yang dimanipulasi, baik dari kertas maupun layar ponsel, di dunia nyata.

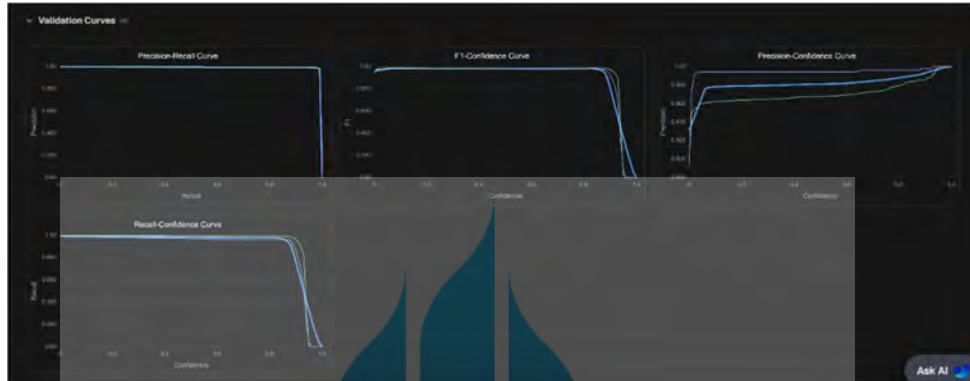


Gambar 4.36 *Confusion Matrix*

Berdasarkan matriks pada Gambar 4.36, terdapat fakta penting yang sangat krusial mengenai keandalan dan keamanan sistem. Pada kasus *False Positive* wajah palsu yang salah diprediksi sebagai asli, model mencatatkan angka 0% yang sempurna. Ini membuktikan bahwa tidak ada serangan pemalsuan visual yang

berhasil melewati perlindungan model AI. Meskipun ada sedikit angka (2,9%) yang menunjukkan hasil *False Negative*, wajah asli dianggap palsu, kondisi ini tetap bisa diterima dalam standar keamanan biometrik, karena pengguna yang sah hanya perlu memindai ulang wajahnya.

Selain menggunakan *confusion matrix*, kemampuan model juga dilihat melalui *Validation Curves*.



Gambar 4.37 *Validation Curves*

Berdasarkan Gambar 4.37, terutama pada kurva F1-Confidence, dapat dilihat bahwa model mampu mencapai skor F1 tertinggi, keseimbangan sempurna antara Presisi dan Recall dalam rentang tingkat kepercayaan confidence threshold yang cukup lebar. Kurva yang datar di atas dan baru turun di ambang batas keyakinan ekstrem menunjukkan bahwa model AI sangat yakin saat menyatakan suatu item sebagai "Asli" atau "Palsu", tanpa adanya keraguan komputasi yang signifikan.

Pengujian yang dilakukan menunjukkan bahwa model YOLO11n sangat cocok digunakan sebagai pelindung pertama dalam sistem keamanan yang handal untuk menghalangi anomali pada tekstur wajah dua dimensi.

4.4.2 Pengujian Fungsionalitas (*Black Box Testing*)

Pengujian fungsionalitas sistem dilakukan menggunakan metode Black Box Testing. Pengujian ini bertujuan memastikan bahwa semua fitur antarmuka yang berinteraksi langsung dengan pengguna, mulai dari proses login, pengelolaan data oleh SDM dan Kabag, sampai verifikasi biometrik oleh peserta, berjalan sesuai

dengan kebutuhan yang sudah ditentukan. Rincian skenario, masukan, serta hasil pengujian disajikan pada Tabel 4.4 berikut.

Tabel 4.4 Tabel Blackbox Testing

Tabel Blackbox Testing						
No	Fungsi yang diuji	Skenario Pengujian	Input	Output yang diharapkan	Output yang didapat	Status
1.	Login	Login dengan kredensial valid	Email dan password benar sesuai role	Sistem mengarahkan ke halaman dashboard sesuai role	Sistem berhasil mengarahkan ke dashboard sesuai role	Berhasil
2.	Login Peserta	Login dengan kredensial valid	Email dan password benar sesuai role	Sistem mengarahkan ke halaman dashboard peserta	Sistem berhasil mengarahkan ke dashboard peserta	Berhasil
3.	Login	Login dengan password salah	Email benar, password salah	Sistem menampilkan pesan kesalahan autentikasi	Sistem menampilkan pesan kesalahan autentikasi	Berhasil
4.	Login	Login dengan kolom kosong	Email dan password tidak diisi	Sistem menampilkan pesan validasi	Sistem menampilkan validasi	Berhasil
5.	Logout	Logout dari sistem	Klik logout	Sistem mengarahkan ke halaman login	Redirect ke halaman login	Berhasil
SDM						
6.	Dashboard SDM	Menampilkan tabel rekomendasi kelayakan	Login sebagai SDM	Dashboard menampilkan tabel rekomendasi beserta nilai 4 kriteria dan label layak/tidak layak	Tabel rekomendasi berhasil ditampilkan dengan data yang sesuai	Berhasil
7.	Master Data	Menambahkan data	Data lengkap valid	Sistem menyimpan data dan memperbarui daftar	Data peserta berhasil disimpan dan tampil di daftar	Berhasil
8.	Master Data	Mengubah data yang sudah ada	Data yang diubah valid dan lengkap	Sistem memperbarui data Dan menampilkan perubahan	Data berhasil diperbarui sesuai perubahan	Berhasil
9.	Master Data	Menghapus data yang sudah ada	Pilih data lalu konfirmasi hapus	Sistem menghapus data dan memperbarui daftar	Data berhasil dihapus dari sistem	Berhasil
10.	Cetak Presensi	Mencetak laporan presensi	Klik tombol cetak setelah filter diterapkan	Sistem menghasilkan laporan presensi siap cetak	Laporan presensi berhasil dicetak	Berhasil

Kabag						
11.	Dashboard Kabag	Menampilkan data hanya dari bagiannya	Login sebagai kabag	Dashboard hanya menampilkan data peserta dari bagian kabag yang bersangkutan	Data yang tampil hanya dari bagian yang login	Berhasil
12.	Penilaian	Input nilai valid pada 4 kriteria	Nilai 1-10 untuk kinerja, tanggung jawab, kedisiplinan, komunikasi	Sistem menyimpan nilai dan menampilkan data di tabel bawah	Nilai berhasil disimpan dan ditampilkan	Berhasil
13.	Paraf Digital	Paraf logbook	Klik paraf dan simpan	Sistem menyimpan paraf	Paraf berhasil disimpan	Berhasil
Peserta						
14.	Pendaftaran Wajah	Registrasi biometrik awal	Wajah asli menghadap kamera dalam pencahayaan normal	Sistem menangkap 15 bingkai wajah secara kontinu, mengekstrak fitur, dan menyimpannya ke basis data	15 sampel wajah berhasil diekstrak dan tersimpan di PostgreSQL	Berhasil
15.	Presensi Lapis 1	Pendeteksian <i>Passive Liveness</i>	Wajah asli menghadap kamera dalam pencahayaan normal	Sistem mendeteksi asli secara <i>real time</i>	Sistem mendeteksi asli secara <i>real time</i>	Berhasil
16.	Presensi Lapis 2	Validasi <i>Active Liveness Challenge</i>	Wajah asli menoleh sesuai arahan dan tersenyum mengikuti instruksi sistem	Modul MediaPipe mendeteksi gerakan ruang, <i>stepper</i> animasi terisi penuh, dan memicu pengiriman citra ke server	Indikator progres penuh, lolos ke tahap evaluasi berikutnya	Berhasil
17.	Presensi serangan spoofing	Deteksi serangan manipulasi statis dan dinamis (<i>Spoofing</i>)	Foto cetak dua dimensi dan layar device dihadapkan pada kamera	Tantangan <i>active liveness</i> gagal (waktu habis) ATAU model YOLO11n mendeteksi citra sebagai <i>fake</i>	Akses diblokir otomatis dengan peringatan indikasi wajah palsu	Berhasil
18.	Presensi mismatch	Pengujian ketidakcocokan identitas silang	User A memindai wajah aslinya pada sesi presensi milik akun User B	Algoritma Cosine Similarity menolak pencocokan (skor ≤ 0.5)	Akses ditolak karena identitas biometrik tidak sesuai dengan sesi akun	Berhasil

4.4.3 Daftar Responden

Pengujian fungsionalitas dan penerimaan sistem (User Acceptance Testing) dalam penelitian ini melibatkan partisipasi langsung dari perwakilan berbagai unit

kerja operasional di Universitas Mercu Buana. Responden dipilih secara sengaja agar dapat mewakili pengguna akhir yang dituju, yang meliputi rekan-rekan Developer , IT Support, data analyst, hingga asisten dosen fakultas. Profil dari sepuluh orang responden yang ikut dalam pengujian sistem ini ditampilkan pada Tabel 4.5 di bawah ini.

Tabel 4.5 Daftar Responden

Nama Responden	Biro Responden	Bagian Responden
Faiz Muhammad Ramadhan	Biro Teknologi Informasi	Developer
Elang Pacifico Saquel	Biro Teknologi Informasi	Developer
Dwi nur cahyo	Biro Teknologi Informasi	Developer
Muhammad Ardhan Darmawan	Biro Teknologi Informasi	IT Support
Antonio Simatupang	Biro Karir & Alumni	Data Analyst
Fikri Ramadhan	Biro Karir & Alumni	Data Analyst
Shahibul Amnar	Fakultas Teknik	Asisten Dosen
Nur Ghazali Santoso	Fakultas Teknik	Asisten Dosen
Ahmad Dzulfikri	Fakultas Ekonomi & Bisnis	Developer
Chiryl Malik Izzuddin	Fakultas Ekonomi & Bisnis	Developer

4.4.4 Pengujian Pendaftaran Wajah

Uji coba ini bertujuan agar antarmuka React JS dan layanan Flask API bisa bekerja sama dalam menerima 15 contoh wajah yang benar-benar nyata dan menyimpannya ke dalam database di memori.

Tabel 4.6 Pengujian Pendaftaran Wajah Pada 5 Responden

Akun Peserta	Nama Peserta	Ekspektasi Sistem	Hasil Aktual	Status
Responden 1	Faiz Muhammad Ramadhan	Menangkap 15 frame, ekstraksi vektor, simpan ke database.	Tampil alert "Wajah berhasil didaftarkan!"	Valid

Responden 2	Chiryl Malik Izzuddin	Menangkap 15 frame, ekstraksi vektor, simpan ke database.	Tampil alert "Wajah berhasil didaftarkan!"	Valid
Responden 3	Ahmad Dzulfikri	Menangkap 15 frame, ekstraksi vektor, simpan ke database.	Tampil alert "Wajah berhasil didaftarkan!"	Valid
Responden 4	Elang Pacifico Shaquille	Menangkap 15 frame, ekstraksi vektor, simpan ke database.	Tampil alert "Wajah berhasil didaftarkan!"	Valid
Responden 5	Dwi Nur Cahyo	Menangkap 15 frame, ekstraksi vektor, simpan ke database.	Tampil alert "Wajah berhasil didaftarkan!"	Valid
Responden 6	Antonio Parlindungan Simatupang	Menangkap 15 frame, ekstraksi vektor, simpan ke database.	Tampil alert "Wajah berhasil didaftarkan!"	Valid
Responden 7	Fikri Ramadhan	Menangkap 15 frame, ekstraksi vektor, simpan ke database.	Tampil alert "Wajah berhasil didaftarkan!"	Valid
Responden 8	Shahibul Amnar	Menangkap 15 frame, ekstraksi vektor, simpan ke database.	Tampil alert "Wajah berhasil didaftarkan!"	Valid
Responden 9	Nur Ghazali Santoso	Menangkap 15 frame, ekstraksi vektor, simpan ke database.	Tampil alert "Wajah berhasil didaftarkan!"	Valid

Responden 10	Muhammad Ardhan Darmawan	Menangkap 15 frame, ekstraksi vektor, simpan ke database.	Tampil alert "Wajah berhasil didaftarkan!"	Valid
-----------------	--------------------------------	--	--	-------

Berdasarkan hasil uji coba pada Tabel 4.6, dapat disimpulkan bahwa fitur pendaftaran wajah berjalan dengan sangat lancar dan berhasil mencapai tingkat keberhasilan 100% pada 10 orang responden. Sistem bisa bekerja sesuai harapan yang sudah direncanakan, yaitu dapat menangkap 15 frame wajah peserta, mengambil fitur wajah dan mengubahnya menjadi data berbentuk vektor, lalu menyimpan data tersebut ke dalam database. Proses ini berhasil diverifikasi oleh sistem yang selalu memberi respons nyata berupa peringatan atau notifikasi 'Wajah berhasil didaftarkan!' setiap kali ada sesi pengujian, sehingga semua pengujian dianggap valid.

4.4.5 Pengujian Presensi Wajah Asli Berhasil

Pengujian ini menyimulasikan kondisi normal di mana pemilik akun yang sah melakukan pemindaian wajah. Sistem diharapkan mampu mendeteksi liveness *real* dan mencocokkan identitas dengan tingkat kemiripan tinggi.

Tabel 4.7 Pengujian Presensi

Nama Peserta	Prediksi <i>passive liveness</i>	Hasil Cosine Similarity	Prediksi <i>active liveness</i>	Waktu Penyelesaian Tantangan	Status
Faiz Muhammad Ramadhan	<i>real</i>	0.67	<i>real</i>	18.44 detik	Valid
Chiryl Malik Izzuddin	<i>real</i>	0.65	<i>real</i>	20.13 detik	Valid
Ahmad Dzulfikri	<i>real</i>	0.66	<i>real</i>	16.52 detik	Valid

Elang Pacifico Shaquille	<i>real</i>	0.69	<i>real</i>	19.41 detik	Valid
Dwi Nur Cahyo	<i>real</i>	0.68	<i>real</i>	18.46 detik	Valid
Antonio Parlindungan Simatupang	<i>real</i>	0.72	<i>real</i>	17.44 detik	Valid
Shahibul Amnar	<i>real</i>	0.66	<i>real</i>	18.33 detik	Valid
Nur Ghazali Santoso	<i>real</i>	0.64	<i>real</i>	17.37 detik	Valid
Muhammad Ardhan Darmawan	<i>real</i>	0.70	<i>real</i>	16.51 detik	Valid
Fikri Ramadhan	<i>real</i>	0.69	<i>real</i>	18.51 detik	Valid

Berdasarkan hasil uji coba pada Tabel 4.7, terlihat bahwa sistem berhasil memverifikasi kehadiran pengguna asli dengan sukses 100% pada 10 orang responden. Seluruh peserta terbukti memiliki wajah nyata baik pada prediksi passive liveness maupun active liveness. Tingkat kesesuaian identitas yang dihitung dengan algoritma Cosine Similarity memiliki nilai antara 0.64 sampai 0.72, yang semuanya melebihi batas minimum validasi (≥ 0.5). Selain itu, rata-rata waktu yang dibutuhkan oleh peserta untuk menyelesaikan seluruh rangkaian tantangan gerakan wajah berada dalam rentang 16 hingga 20 detik. Ini menunjukkan bahwa sistem mampu mengenali identitas yang benar dengan tepat dan juga cukup cepat dalam menyelesaikan tugasnya.

4.4.6 Pengujian Serangan Wajah Palsu

Pengujian ini dilakukan untuk mengetahui seberapa tangguh model AI pelindung pertama, yaitu YOLO11n dan MediaPipe sebagai pelindung keduanya.

Responden diberi instruksi untuk mengatur kamera dengan menggunakan foto cetak berbentuk kertas atau tampilan foto/video dari layar ponsel.

Tabel 4.8 Pengujian Serangan Wajah Palsu

Nama Peserta	Media Serangan	Prediksi <i>Passive Liveness</i>	Hasil <i>Active Liveness</i>	Status
Faiz Muhammad Ramadhan	Foto layar ponsel (HP).	<i>fake</i>	-	Valid
Chiryl Malik Izzuddin	Foto layar ponsel (HP).	<i>real</i>	gagal	Valid
Ahmad Dzulfikri	Foto layar ponsel (HP).	<i>fake</i>	-	Valid
Elang Pacifico Shaquille	Foto Kertas	<i>fake</i>	-	Valid
Dwi Nur Cahyo	Foto Kertas	<i>fake</i>	-	Valid
Antonio Parlindungan Simatupang	Video layar ponsel (HP)	<i>fake</i>	-	Valid
Shahibul Amnar	Foto Kertas	<i>real</i>	gagal	Valid
Nur Ghazali Santos	Foto layar ponsel (HP).	<i>fake</i>	-	Valid
Muhammad Ardhan Darmawan	Video layar ponsel (HP)	<i>fake</i>	-	Valid
Fikri Ramadhan	Video layar ponsel (HP)	<i>fake</i>	-	Valid

Berdasarkan hasil uji coba pada Tabel 4.8, bisa disimpulkan bahwa sistem memiliki ketahanan yang sangat baik terhadap berbagai jenis serangan manipulasi visual (spoofing), baik yang menggunakan foto cetak maupun rekaman layar dari perangkat gawai. Model passive liveness YOLO11n mampu mendeteksi dan

memblokir 80% dari serangan yang menggunakan wajah palsu pada tahap awal secara langsung. Sementara itu, pada 20% insiden yang terdeteksi sebagai wajah asli oleh lapisan pertama, sistem berhasil menggagalkan serangan tersebut di tahap *active liveness* karena media penyerang tidak mampu menjawab instruksi tantangan gerak. Dengan demikian, tingkat keberhasilan penolakan serangan mencapai 100%, dan semua skenario pengujian dianggap valid.

4.4.7 Pengujian Serangan Spoofing Berupa Video

Pengujian ini untuk menguji *passive liveness* terhadap 2 skenario serangan, yaitu berupa serangan *deepfake*, dan *video injection*.

Tabel 4.9 Pengujian Spoofing

No	Skenario Serangan	Passive Liveness	Active Liveness	Status Akhir
1	Video injection (rekaman video wajah asli via virtual camera)	Real (lolos)	Gagal	Ditolak sistem
2	Deepfake berbentuk video	Real (lolos)	Gagal	Ditolak sistem

Hasil uji menunjukkan bahwa *passive liveness detection* bisa dikelabui oleh serangan *video injection* atau *deepfake*, karena secara tampilan kedua jenis serangan tersebut menunjukkan wajah yang terlihat nyata, sehingga dianggap sebagai "asli" oleh model YOLO11n. Namun, ketika serangan tersebut diuji hingga tahap uji *active liveness*, baik bentuk video injeksi maupun *deepfake* berupa video tidak berhasil melewati pengujian. Hal ini terjadi karena video yang telah direkam sebelumnya, baik itu hasil rekaman asli maupun hasil sintesis *deepfake*, tidak bisa menjawab instruksi tantangan yang diberikan secara acak dan langsung oleh sistem dalam jangka waktu yang ditentukan, yaitu 5 detik.

4.4.8 Pengujian Robustness Kondisi Nyata

Pengujian ini menguji model *passive liveness* dalam beberapa kondisi nyata, berupa pencayahan dan aksesoris.

Tabel 4.10 Pengujian Kondisi Nyata

No	Kondisi Pengujian	Hasil Passive Liveness	Rentang Cosine Similarity	Status Akhir
1	Pencahayaan Terang	Samar, sempat terdeteksi tidak dikenali/fake	0.57 – 0.68	Lolos
2	Pencahayaan Normal	Real, konsisten	0.65 – 0.75	Lolos
3	Pencahayaan Gelap	Samar, sempat terdeteksi tidak dikenali/fake	0.57 – 0.68	Lolos
4	Sudut Frontal	Real, konsisten	0.65 – 0.75	Lolos
5	Sudut Miring (kiri/kanan/atas/bawah)	Samar, sempat terdeteksi tidak dikenali/fake	0.57 – 0.68	Lolos
6	Masker	Samar, sempat terdeteksi tidak dikenali/fake	0.57 – 0.68	Lolos
7	Kacamata	Real, konsisten	0.65 – 0.75	Lolos

Berdasarkan hasil uji pada Tabel 4.10, proses pencocokan wajah menunjukkan hasil yang tetap konsisten dalam berbagai kondisi uji. Nilai cosine similarity selalu di atas ambang batas 0.5 yang ditetapkan sistem, meskipun terjadi penurunan nilai di beberapa kondisi seperti pencahayaan terang, wajah miring, dan penggunaan masker (0.57–0.68), dibandingkan kondisi normal (0.65–0.75). Ini menunjukkan bahwa modul pengenalan wajah masih tahan terhadap berbagai kondisi yang berbeda.

Namun, terdeteksi pola yang berbeda pada modul passive liveness detection. Dalam kondisi cahaya terang atau gelap, sudut wajah miring, dan saat menggunakan masker, passive liveness sempat memberikan indikasi samar atau tidak konsisten terhadap wajah orang sebenarnya. Meskipun begitu, sistem tetap mampu memverifikasi pengguna dengan benar. Temuan ini menunjukkan bahwa

sistem dua lapis (passive + active liveness) memiliki peran penting sebagai cara mengimbangi ketika salah satu lapisan mengalami penurunan keakuratan karena perubahan kondisi lingkungan. Selain itu, temuan juga menunjukkan bahwa modul passive liveness lebih rentan terhadap perubahan kondisi nyata dibandingkan modul active liveness maupun face matching.

4.4.9 Pengujian Silang Akun (*Mismatch Identity*)

Pengujian pamungkas ini menguji akurasi model InsightFace. Pengujian dilakukan dengan menggunakan wajah asli, tetapi wajah tersebut digunakan untuk memindai pada akun orang lain yang sedang login di browser web.

Tabel 4.11 *Mismatch Identity*

Wajah Aktual di Kamera	Akun yang Sedang Login	Prediksi Liveness	Respons Sistem	Status
Faiz Muhammad Ramadhan	Muhammad Ardhan Darmawan	<i>real</i>	Ditolak, alert "Wajah terdeteksi Faiz Muhammad Ramadhan, bukan Muhammad Ardhan Darmawan!"	Valid
Muhammad Ardhan Darmawan	Faiz Muhammad Ramadhan	<i>real</i>	Ditolak, alert "Wajah terdeteksi Muhammad Ardhan Darmawan, bukan Faiz Muhammad Ramadhan!"	Valid
Chiryl Malik Izzuddin	Ahmad Dzulfikri	<i>real</i>	Ditolak, alert "Wajah terdeteksi Chiryl Malik Izzuddin, bukan"	Valid

			Ahmad Dzulfikri!"	
Ahmad Dzulfikri	Chiryl Malik Izzuddin	<i>real</i>	Ditolak, alert "Wajah terdeteksi Ahmad Dzulfikri, bukan Chiryl Malik Izzuddin!"	Valid
Elang Pacifico Shaquille	Dwi Nur Cahyo	<i>real</i>	Ditolak, alert "Wajah terdeteksi Elang Pacifico Shaquille, bukan Dwi Nur Cahyo!"	Valid
Dwi Nur Cahyo	Elang Pacifico Shaquille	<i>real</i>	Ditolak, alert "Wajah terdeteksi Dwi Nur Cahyo, bukan Elang Pacifico Shaquille!"	Valid
Antonio Parlindungan Simatupang	Fikri Ramadhan	<i>real</i>	Ditolak, alert "Wajah terdeteksi Antonio Parlindungan Simatupang, bukan Fikri Ramadhan!"	Valid
Fikri Ramadhan	Antonio Parlindungan Simatupang	<i>real</i>	Ditolak, alert "Wajah terdeteksi Fikri Ramadhan, bukan Antonio Parlindungan Simatupang!"	Valid

Shahibul Amnar	Nur Ghazali Santoso	<i>real</i>	Ditolak, alert "Wajah terdeteksi Shahibul Amnar, bukan Nur Ghazali Santoso!"	Valid
Nur Ghazali Santoso	Shahibul Amnar	<i>real</i>	Ditolak, alert "Wajah terdeteksi Nur Ghazali Santoso, bukan Shahibul Amnar!"	Valid

Berdasarkan hasil uji silang akun (mismatch identity) pada Tabel 4.11, dapat disimpulkan bahwa model ekstraksi fitur wajah (InsightFace) dan algoritma pencocokan identitas (Cosine Similarity) berjalan dengan akurasi 100%. Semua skenario pengujian menunjukkan bahwa meskipun wajah yang muncul di depan kamera berhasil dideteksi sebagai wajah manusia asli (prediksi liveness menunjukkan nilai *real*), sistem tetap dengan tegas menolak kehadiran tersebut karena wajah yang sebenarnya tidak sesuai dengan data biometrik pemilik akun. Sistem juga terus menampilkan notifikasi penolakan yang jelas beserta nama pengguna yang terlibat, sehingga semua pengujian dianggap valid. Ini menunjukkan bahwa sistem tersebut aman dari upaya menipu dengan menggunakan akun orang lain atau rekan kerja untuk absen.

4.4.10 Pengujian Latensi Sistem dan Stabilitas

Pengujian ini mengukur secara kuantitatif seberapa efisien arsitektur microservices dalam memproses pertukaran data gambar berformat Base64 ke model AI, dengan menghitung seluruh waktu yang dibutuhkan dalam satuan milidetik (ms).

1. Pengujian Latensi Inferensi Wajah Asli

Pengukuran ini dilakukan saat presensi berjalan lancar. Sistem mencatat waktu yang dibutuhkan untuk mengenali satu *frame* citra asli pengguna hingga diizinkan masuk.

Tabel 4.12 Latensi Wajah Asli

Nama Responden	Hasil Prediksi Identitas	Prediksi Liveness	Catatan Log Waktu Respons (Live)	Status
Faiz Muhammad Ramadhan	Valid (Faiz Muhammad Ramadhan)	<i>real</i>	365 ms	Valid
Chiryl Malik Izzuddin	Valid (Chiryl Malik Izzuddin)	<i>real</i>	292 ms	Valid
Ahmad Dzulfikri	Valid (Ahmad Dzulfikri)	<i>real</i>	424 ms	Valid
Elang Pacifico Shaquille	Valid (Elang Pacifico Shaquille)	<i>real</i>	318 ms	Valid
Dwi Nur Cahyo	Valid (Dwi Nur Cahyo)	<i>real</i>	288 ms	Valid
Antonio Parlindungan Simatupang	Valid (Antonio Parlindungan Simatupang)	<i>real</i>	293 ms	Valid
Fikri Ramadhan	Valid (Fikri Ramadhan)	<i>real</i>	283 ms	Valid
Shahibul Amnar	Valid (Shahibul Amnar)	<i>real</i>	277 ms	Valid
Nur Ghazali Santoso	Valid (Nur Ghazali Santoso)	<i>real</i>	299 ms	Valid
Muhammad Ardhan Darmawan	Valid (Muhammad Ardhan Darmawan)	<i>real</i>	265 ms	Valid

2. Pengujian Stabilitas Jaringan pada Siklus Serangan (*Stress Testing*)

Mengevaluasi daya tahan server Flask API saat terus-menerus menerima permintaan pengiriman gambar yang berulang karena kamera klien terblokir oleh objek palsu.

Tabel 4.13 Latensi Serangan Spoofing

Putaran Frame	Sampel Partisipan Uji	Klasifikasi Liveness AI	Status Otorisasi Jaringan	Durasi Waktu Respons
Frame 1	Muhammad Alfa Fauzan	<i>Fake</i>	Diblokir (Scan Ulang)	227 ms
Frame 2	Muhammad Alfa Fauzan	<i>Fake</i>	Diblokir (Scan Ulang)	179 ms
Frame 3	Muhammad Alfa Fauzan	<i>Fake</i>	Diblokir (Scan Ulang)	188 ms
Frame 4	Muhammad Alfa Fauzan	<i>Fake</i>	Diblokir (Scan Ulang)	251 ms
Frame 5	Muhammad Alfa Fauzan	<i>Fake</i>	Diblokir (Scan Ulang)	178 ms
Frame 6	Muhammad Alfa Fauzan	<i>Fake</i>	Diblokir (Scan Ulang)	189 ms
Frame 7	Muhammad Alfa Fauzan	<i>Fake</i>	Diblokir (Scan Ulang)	180 ms
Frame 8	Muhammad Alfa Fauzan	<i>Fake</i>	Diblokir (Scan Ulang)	190 ms

Frame 9	Muhammad Alfa Fauzan	<i>Fake</i>	Diblokir (Scan Ulang)	192 ms
Frame 10	Muhammad Alfa Fauzan	<i>Fake</i>	Diblokir (Scan Ulang)	226 ms

Berdasarkan data kuantitatif pada Tabel 4.12 dan Tabel 4.13, dapat dilihat bahwa rata-rata waktu respons sistem secara keseluruhan tetap stabil dalam rentang 240 ms hingga 300 ms. Hasil ini menunjukkan bahwa metode pengambilan data langsung dari RAM (caching lookups) berhasil mengurangi waktu pemrosesan di bawah batas toleransi 500 ms, sehingga mampu memberikan pengalaman presensi *real-time* yang halus dan tidak ada gangguan.

4.4.11 Pengujian Integritas Transaksi Data

Tahap ini menguji kemampuan baris kode DB::beginTransaction() dalam pengontrol Laravel untuk mengumpulkan data presensi akhir, yang mencakup

```
[2026-06-24 07:52:52] production.INFO: [INTEGRITAS DATA AMAN] Presensi ID Peserta: berhasil disimpan utuh.
[2026-06-24 08:42:53] production.INFO: [INTEGRITAS DATA AMAN] Presensi ID Peserta: berhasil disimpan utuh.
[2026-06-25 02:26:18] production.INFO: [INTEGRITAS DATA AMAN] Presensi ID Peserta: berhasil disimpan utuh.
[2026-06-25 02:44:23] production.INFO: [INTEGRITAS DATA AMAN] Presensi ID Peserta: berhasil disimpan utuh.
[2026-06-25 02:55:55] production.INFO: [INTEGRITAS DATA AMAN] Presensi ID Peserta: berhasil disimpan utuh.
```

gambar terakhir, koordinat lokasi, serta waktu, milik responden yang sudah berhasil diverifikasi oleh AI.

Gambar 4.38 Catatan Log Bukti Keberhasilan Transaksi Atomik PostgreSQL

Pencatatan log aktivitas pada Gambar 4.38 menunjukkan bahwa basis data relasional PostgreSQL berjalan dengan metode atomik mutlak. Jika terjadi gangguan tak terduga, seperti hilangnya sinyal internet pada perangkat klien saat mengirim formulir akhir, maka seluruh perintah INSERT akan dibatalkan secara

otomatis (rollBack), sehingga struktur database tidak terkena kerusakan data (data corruption).

4.4.12 User Acceptance Testing (UAT)

UAT diuji pada 10 orang responden. Setiap orang yang mengikuti uji coba mengevaluasi fitur dan memberikan penilaian berdasarkan pengalaman langsung dalam menggunakan sistem tersebut. Hasil rekapitulasi dari jawaban responden seperti berikut.

Tabel 4.14 Tabel Rekapitulasi *UAT*

No	Pernyataan	STS	TS	N	S	SS	Skor
Aspek 1							
1.	Tampilan antar muka mudah dipahami tanpa memerlukan pelatihan khusus	0	0	1	2	7	46
2.	Alur navigasi jelas dan tidak membingungkan	0	0	0	2	8	48
3.	Proses login kedalam sistem mudah dilakukan	0	0	0	2	8	48
4.	Fitur-fitur yang tersedia mudah ditemukan dan digunakan	0	0	0	1	9	49
Aspek 2							
1.	Dashboard peserta menampilkan rekap kehadiran yang imformatif dan mudah dipahami	0	0	0	6	4	44
2.	Proses presensi harian melalui sistem lebih mudah dibandingkan metode manua	0	0	2	4	4	42

3.	Fitur pengisian logbook pada halaman presensi sudah sesuai dengan kebutuhan dokumentasi aktivitas harian	0	0	1	3	6	45
4.	Informasi rekap kehadiran yang ditampilkan di dashboard sudah mencerminkan aktivitas presensi saya dengan benar	0	0	0	5	5	45
5.	Sistem presensi digital ini lebih praktis dan efisien dibandingkan metode pencatatan manual	0	0	1	3	6	45
Aspek 3							
1.	Secara keseluruhan saya merasa puas dengan sistem yang dibangun	0	0	1	3	6	45
2.	Sistem ini layak untuk digunakan dalam pengelolaan peserta magang dan pekerja lepas	0	0	0	4	6	46
3.	Saya bersedia menggunakan sistem ini jika diterapkan secara nyata	0	0	0	2	8	48

Hasil dari kuisioner diperoleh skor sebesar 551 dari skor maksimum 600, sehingga diperoleh presentase UAT sebesar 91,83%.

$$P = \frac{551}{600} \times 100\% = 91,83\%$$

Tabel 4.15 *User Acceptance Testing*

Role	Responden	Total Skor	Skor Maksimal	Presentasei
------	-----------	------------	---------------	-------------

Peserta	10	551	600	91,83%.
---------	----	-----	-----	---------

Berdasarkan hasil pengujian User Acceptance Testing (UAT) yang melibatkan 10 orang peserta, penilaian terhadap antarmuka dan fungsi sistem memberikan skor total sebesar 551 dari skor maksimal yang sebesar 600. Dengan menghitung persentase kelayakan, sistem presensi ini mencapai angka 91,83%. Angka tersebut termasuk dalam kategori 'Sangat Layak' atau 'Sangat Baik' berdasarkan skala penilaian kelayakan sistem informasi.

Dari hasil tersebut, bisa disimpulkan bahwa sistem presensi berbasis pengenalan wajah dengan dua lapisan yang diusulkan memiliki tingkat kemudahan penggunaan yang sangat tinggi. Antarmuka pengguna dianggap jelas, cepat merespons, dan mudah digunakan tanpa menyebabkan kebingungan, sehingga sistem ini dianggap sangat cocok untuk langsung diterapkan secara penuh demi memudahkan pengelolaan administrasi peserta magang dan pekerja lepas di Universitas Mercu Buana.

4.4.13 Pemeliharaan Sistem (*Maintenance*)

Tahap terakhir dalam metode Waterfall adalah pemeliharaan sistem yang bertujuan untuk memastikan kelancaran operasional di lingkungan server produksi Universitas Mercu Buana. Fokus utama dalam pemeliharaan mencakup pengawasan terhadap kapasitas memori (RAM) pada mesin virtual untuk mencegah terjadinya kebocoran memori yang disebabkan oleh aktivitas komputasi berbasis AI, yang selanjutnya ditindaklanjuti dengan menjadwalkan pemulihan layanan yang dilakukan di luar masa operasional. Selain itu, dilakukan pemeriksaan berkala terhadap log sistem (laravel.log) untuk memastikan keamanan dan integritas transaksi database PostgreSQL tetap terjaga, serta sinkronisasi zona waktu (timezone) server secara rutin agar catatan waktu kehadiran tetap akurat. Rangkaian langkah pencegahan ini diterapkan secara terus-menerus untuk memastikan layanan tetap tersedia secara stabil dan aplikasi tetap tangguh dalam jangka waktu yang lama.