

## BAB III

### PELAKSANAAN KERJA MAGANG

#### **3.1 Kedudukan dan Koordinasi**

Posisi *Software Engineering Intern* di bawah naungan *Divisi Research and Development* (RnD) dengan fokus pengembangan sistem berbasis kecerdasan buatan menjadi ranah pelaksanaan kerja praktik di PT Satya Solusindo Indonesia. Tanggung jawab utama mencakup keterlibatan teknis dalam proyek *AI Visitor Management System with People Counting System*. Sistem tersebut dirancang sebagai solusi integratif untuk absensi otomatis, pencatatan kunjungan, *monitoring* kapasitas gedung, serta visualisasi data kehadiran secara *real-time*. Lingkup pengembangan proyek meliputi sisi *frontend*, *backend*, integrasi basis data, hingga pemrosesan AI. Implementasi teknologi cerdas mencakup fitur *face recognition* menggunakan *library* InsightFace serta *people detection and counting* berbasis algoritma YOLO. Peran ini mewajibkan penguasaan konsep teoretis sekaligus kemampuan praktis dalam *problem solving*, *debugging*, dan integrasi lintas teknologi. Stabilitas, keamanan, dan efektivitas operasional sistem menjadi parameter utama keberhasilan integrasi tersebut.

Alur kerja fungsional di perusahaan menuntut koordinasi langsung dengan *Supervisor RnD*. Pelaksanaan ini mencakup penerimaan arahan teknis terstruktur serta partisipasi wajib dalam rapat internal mingguan. Forum tersebut digunakan untuk mempresentasikan progres, kendala teknis, dan rencana pengembangan sistem berikutnya. Tugas juga melibatkan interaksi lintas tim, seperti tim *User Interface/User Experience* (UI/UX) untuk penetapan desain antarmuka, tim infrastruktur terkait kebutuhan *server* dan *deployment*, serta tim *Quality Assurance* (QA) untuk pengujian performa sistem. Koordinasi dilakukan melalui penyusunan dokumentasi teknis, penyerahan *source code* via *repository* Git perusahaan, dan penjelasan detail setiap fitur atau modul sistem yang mencapai tahap finalisasi atau siap uji. Kedudukan *intern* tersebut merupakan bagian integral dari alur koordinasi pengembangan sistem yang menuntut komunikasi efektif lintas divisi dan keahlian.

Inisiasi proyek bermula dengan pembuatan *mock-up* sistem dan desain antarmuka sebagai langkah awal penyelarasan konsep. Perancangan visual UI/UX menggunakan *tools prototyping* dilaksanakan sesuai identitas kebutuhan perusahaan. Presentasi konsep dilakukan secara langsung kepada CEO guna memperoleh persetujuan arah pengembangan, fungsi utama sistem, serta definisi skenario penggunaan. Skenario tersebut mencakup dua kategori pengguna, yaitu *employee* dan *visitor*, serta satu peran admin yang bertugas memantau aktivitas pengguna pada setiap lantai gedung secara *real-time*. Fase *mock-up* ini menjadi tahapan krusial dalam penentuan gambaran menyeluruh alur aplikasi, desain *dashboard monitoring*, halaman absensi, tampilan verifikasi wajah, serta kontrol akses data.

Eksplorasi dan eksperimen teknis dilakukan terhadap model InsightFace untuk *face recognition* serta algoritma YOLO untuk *people counting* dan *people detection*. Fokus kegiatan pada fase ini meliputi analisis *dataset*, pelaksanaan *inference test*, pemahaman parameter model, serta pengujian berbagai konfigurasi yang memengaruhi akurasi dan kecepatan inferensi. Pengembangan sistem berlanjut ke tahap *backend* menggunakan *framework* FastAPI pasca-evaluasi performa kedua model *machine learning* tersebut. Tahapan ini mencakup pembuatan *endpoint API* untuk memfasilitasi komunikasi antara sistem AI, basis data, dan *dashboard website*. Arsitektur *backend* dirancang secara modular guna menangani permintaan login, absensi otomatis, penyimpanan histori pengunjung dan karyawan, data statistik, serta manajemen admin.

Realisasi antarmuka pengguna dilaksanakan melalui pengembangan *frontend* berbasis HTML, CSS, dan JavaScript, mengacu pada desain *mock-up* yang telah disetujui, menyusul berjalannya sistem *backend*. Perancangan dan implementasi basis data MySQL dilakukan secara simultan untuk memfasilitasi penyimpanan data absensi, identifikasi wajah, log kunjungan, serta pemantauan keberadaan pengguna di setiap lantai. Konsolidasi seluruh komponen independen bermuara pada proses integrasi sistem secara menyeluruh yang menghubungkan modul AI, *backend*, *frontend*, dan basis data. Rangkaian kegiatan ini dilanjutkan dengan tahap

*debugging*, optimasi alur data, serta pengujian *user experience* guna menjamin kemudahan, akurasi, keamanan, dan stabilitas sistem.

### **3.2 Tugas dan Uraian Kerja Magang**

Pelaksanaan magang di PT Satya Solusindo Indonesia (SATSINDO) mencakup rangkaian tugas yang mengintegrasikan bidang otomasi, teknologi sensor, serta pengembangan perangkat lunak berbasis sistem AI. Tahap awal kegiatan difokuskan pada eksplorasi dan percobaan teknis menggunakan sensor *vibration* industrial QM30VT2. Perangkat ini difungsikan untuk akuisisi data kondisi mesin melalui protokol komunikasi UART. Proses pembacaan sinyal digital menggunakan mikrokontroler *Raspberry Pi* menjadi materi pembelajaran teknis sebagai bagian dari sistem pemantauan kesehatan mesin (*condition monitoring*). Pengujian sensor *barcode reader* turut dilakukan guna memahami mekanisme input sistem untuk validasi identitas produk dan otomatisasi pencatatan proses.

Lingkup tanggung jawab diperluas hingga mencakup proyek *Financial Technology* (FinTech) yang berfokus pada analisis dan rekomendasi produk keuangan berbasis kecerdasan buatan, melengkapi aktivitas teknis di bidang sensor. Implementasi teknis melibatkan penggunaan model AI berbasis DeepSeek serta penerapan algoritma rekomendasi, meliputi *content-based filtering*, *collaborative filtering*, dan *hybrid recommendation model*. Pendekatan hibrida tersebut bertujuan menghasilkan akurasi rekomendasi produk yang presisi sesuai profil pengguna. Rangkaian proses pengembangan ini mencakup pengolahan *dataset*, pembuatan skema evaluasi, serta analisis performa algoritma. Penugasan ini memberikan pemahaman mendalam mengenai urgensi *Big Data* dan AI dalam transformasi industri keuangan, khususnya terkait penyediaan pengalaman layanan terpersonalisasi.

Kontribusi juga diberikan pada proyek industri manufaktur aki, yang merupakan sistem deteksi dan penghitungan jumlah *plate* guna menjamin kesesuaian dengan standar mutu produksi. Keterlibatan teknis mencakup pengembangan fungsi tambahan dan *endpoint API* sebagai upaya peningkatan

performa sistem. Tujuan dari peningkatan tersebut adalah efisiensi identifikasi produk cacat serta percepatan proses pemeriksaan kualitas. Tanggung jawab ini memberikan pengalaman observasi nyata terhadap integrasi perangkat otomatis dan perangkat lunak. Kolaborasi teknologi tersebut menjadi solusi efektif dalam meningkatkan produktivitas serta meminimalkan *human error* dalam proses manufaktur.

Agenda pembelajaran teknis diperluas hingga mencakup proses pengembangan *frontend* dan *backend*, melengkapi penugasan berbasis AI dan sensor. Akuisisi keahlian pemrograman *website* menggunakan HTML, CSS, JavaScript, serta pengelolaan *backend* berbasis *framework* FastAPI dan basis data MySQL, menjadi tantangan akademis sekaligus wawasan baru di luar kurikulum inti Teknik Elektro. Fokus materi meliputi perancangan antarmuka pengguna, manajemen API, *routing*, komunikasi *client-server*, serta integrasi basis data demi terciptanya sistem yang komprehensif. Pengalaman ini mengonfirmasi paradigma bahwa sistem industri modern tidak lagi bergantung parsial pada aspek perangkat keras (*hardware*), melainkan menuntut sinergi antara perangkat lunak (*software*), sensor, kecerdasan buatan, dan pengolahan data guna menghasilkan solusi operasional yang cerdas, efisien, dan adaptif.

Keterlibatan pada beberapa projek dan pengembangan produk, projek utama yang menjadi fokus selama masa magang adalah mengerjakan *AI Visitor Management System with People Counting System* sebagai solusi otomatisasi absensi dan pemantauan pergerakan orang di lingkungan perusahaan. Proyek *Visitor Management System* dipilih sebagai proyek utama magang karena memiliki keterkaitan yang kuat dengan bidang teknik elektro dan otomasi industri, khususnya pada integrasi antara perangkat keras dan perangkat lunak. Sistem ini mengotomatisasi proses pencatatan kehadiran, identifikasi pengunjung, serta pemantauan jumlah orang yang sebelumnya dilakukan secara manual. Kamera berperan sebagai sensor visual, jaringan lokal berfungsi sebagai media transmisi data, dan server bertindak sebagai unit pemrosesan yang terintegrasi dengan perangkat lunak berbasis kecerdasan buatan. Integrasi tersebut mencerminkan

prinsip otomasi industri yang mencakup proses sensing, processing, dan pengendalian sistem secara *real-time*. Pendekatan ini memberikan pengalaman aplikatif dalam merancang sistem terintegrasi yang efisien, andal, dan relevan dengan penerapan teknologi otomasi modern dalam bidang Teknik Elektro.

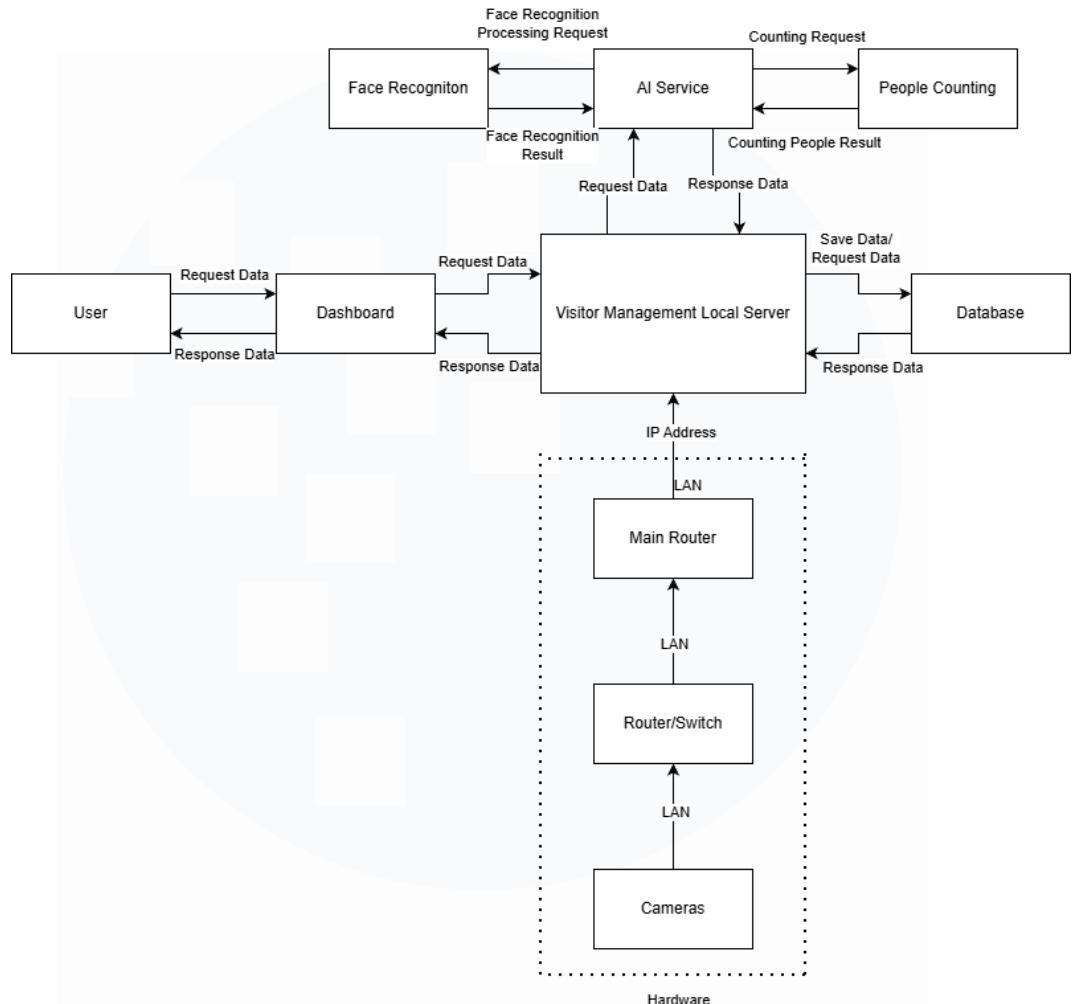
Tahap awal, melakukan analisis kebutuhan dan merancang struktur alur sistem baik dari sisi operasional pengguna maupun dari alur teknis program. Sistem ini dibangun sebagai platform berbasis *website* yang memiliki dua kategori pengguna yakni *employee* dan *visitor*, serta dilengkapi dengan halaman admin yang berfungsi untuk memantau aktivitas kedatangan, kepulangan, dan distribusi pengguna pada setiap lantai atau area. Proses pengembangan, dengan mengimplementasikan MySQL sebagai pusat basis data untuk melakukan proses penyimpanan data absensi, informasi pengguna, log aktivitas, dan data pendukung lainnya secara terstruktur dan efisien.

Konstruksi antarmuka pengguna (*frontend*) direalisasikan menggunakan HTML, CSS, dan JavaScript melalui pendekatan desain responsif guna menjamin aksesibilitas sistem lintas perangkat. Pengembangan sisi backend mengandalkan *framework* FastAPI berbasis Python yang memiliki karakteristik ringan, modern, dan akseleratif dalam penanganan permintaan serta pembangunan *endpoint* layanan. Fase pengembangan dan pengujian awal memanfaatkan XAMPP sebagai server lokal yang menyediakan paket layanan berbasis Apache *web server*. Lingkungan ini memfasilitasi konfigurasi basis data, uji konektivitas, manajemen *request*, serta simulasi *deployment* aplikasi secara terintegrasi, stabil, dan efisien. Mekanisme integrasi *backend* dan *frontend* diterapkan melalui konsep REST API guna menjamin pertukaran data antara pengguna, server, dan basis data berjalan secara *real-time*, stabil, serta adaptif terhadap pengembangan fitur lanjutan[1].

Proses identifikasi dan validasi kehadiran berbasis wajah, dengan mengintegrasikan model InsightFace[2], sebuah *library open-source* yang secara khusus dirancang untuk analisis wajah dan *face recognition* dengan *pipeline* canggih yang mencakup deteksi wajah, *landmark detection*, *face alignment*, hingga

ekstraksi fitur menjadi *embedding* berdimensi tinggi[3]. InsightFace didukung oleh mekanisme pelatihan berbasis ArcFace, sebuah fungsi *loss* yang meningkatkan kemampuan model dalam membedakan wajah yang berbeda secara akurat dengan menciptakan ruang fitur yang lebih diskriminatif, sehingga sangat efektif digunakan dalam proses verifikasi dan identifikasi identitas pengguna[4], [5]. Sebagai bagian dari sistem pemantauan jumlah orang, dengan mengimplementasikan YOLO (*You Only Look Once*) yaitu model deteksi objek *real-time* yang memproses gambar dalam satu kali *forward* secara cepat dan efisien. Model tersebut diterapkan untuk proses *people detection* dan *people counting*[6] dengan memanfaatkan pergerakan *centroid* dari *bounding box* setiap objek yang melintasi zona tertentu sehingga sistem mampu mendekripsi dan mencatat jumlah orang secara otomatis tanpa perangkat tambahan lainnya. Pendekatan ini mendukung kebutuhan pemantauan distribusi pengguna di setiap lantai, mendekripsi kepadatan, serta menjadi bagian dari sistem manajemen fasilitas berbasis kecerdasan buatan.

Visualisasi alur proyek utama disajikan melalui diagram arsitektur digunakan untuk menggambarkan struktur teknis sistem serta hubungan antar komponen secara menyeluruh. Diagram ini berfungsi untuk menjelaskan bagaimana komponen perangkat lunak, perangkat keras, jaringan, dan layanan pendukung saling terhubung dan berkomunikasi. Representasi visual dalam diagram arsitektur memudahkan pemahaman terhadap pembagian peran setiap komponen, alur pertukaran data, serta posisi layanan utama seperti server, basis data, dan kecerdasan buatan dalam sistem. Pendekatan ini membantu perancang sistem dalam memastikan bahwa arsitektur yang dibangun memenuhi kebutuhan fungsional, kinerja, Berikut arsitektur sistemnya:



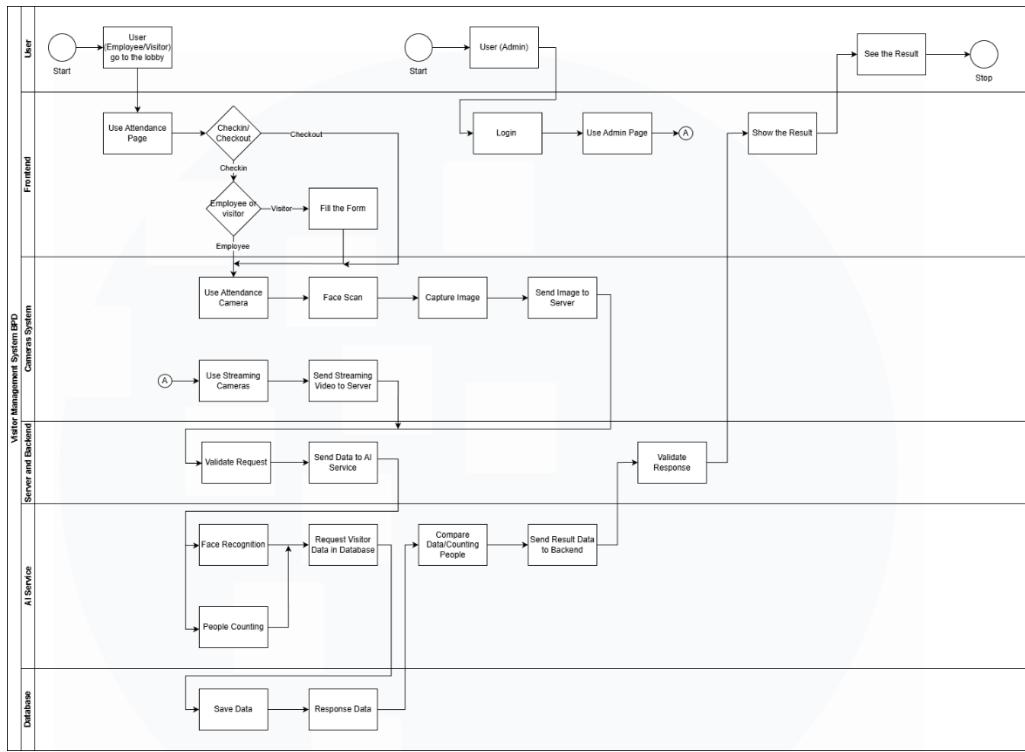
Gambar 3. 1 Diagram Arsitektur Sistem

Arsitektur *Visitor Management System* pada gambar 3.1 menunjukkan sistem terintegrasi yang menggabungkan pengguna, *dashboard*, server lokal, layanan kecerdasan buatan, basis data, serta infrastruktur jaringan lokal. Pengguna berinteraksi melalui *dashboard* sebagai antarmuka utama untuk mengirimkan permintaan data dan menerima hasil pemrosesan. *Dashboard* berkomunikasi langsung dengan *Visitor Management Local Server* yang berfungsi sebagai pusat kendali sistem. Server menerima data dari kamera yang terhubung melalui jaringan *Local Area Network*, kemudian meneruskan permintaan pemrosesan ke layanan kecerdasan buatan yang terdiri atas modul *Face Recognition* dan *People Counting*. Modul *Face Recognition* melakukan identifikasi wajah berdasarkan citra yang diterima, sedangkan modul *People Counting* menghitung jumlah orang dalam area

pemantauan. Hasil pemrosesan dikembalikan ke server dalam bentuk data terstruktur untuk diproses lebih lanjut.

*Visitor Management Local Server* mengelola seluruh alur data dengan menyimpan hasil analisis ke dalam basis data serta mengirimkan respons akhir ke dashboard untuk ditampilkan kepada pengguna. Basis data berfungsi sebagai penyimpanan terpusat yang mencakup data pengenalan wajah dan jumlah pengunjung. Infrastruktur jaringan lokal yang terdiri atas *main router*, *router* atau *switch*, dan kamera memastikan komunikasi data berjalan stabil dan aman. Pendekatan pemrosesan terpusat pada server lokal memungkinkan sistem bekerja secara *real-time*, mengurangi latensi, serta menjaga keamanan data pengunjung sehingga sesuai untuk penerapan sistem manajemen pengunjung di lingkungan institusi atau perusahaan.

Business Process Diagram merupakan alat pemodelan yang digunakan untuk menggambarkan alur proses bisnis secara sistematis dan terstruktur. Diagram ini berguna untuk menjelaskan bagaimana suatu sistem bekerja dari awal hingga akhir dengan memperlihatkan urutan aktivitas, pihak yang terlibat, serta hubungan antarproses. Business Process Diagram membantu pengembang, analis sistem, dan pemangku kepentingan dalam memahami mekanisme operasional sistem, mengidentifikasi kebutuhan fungsional, serta memastikan bahwa setiap proses berjalan sesuai dengan tujuan sistem yang dirancang.



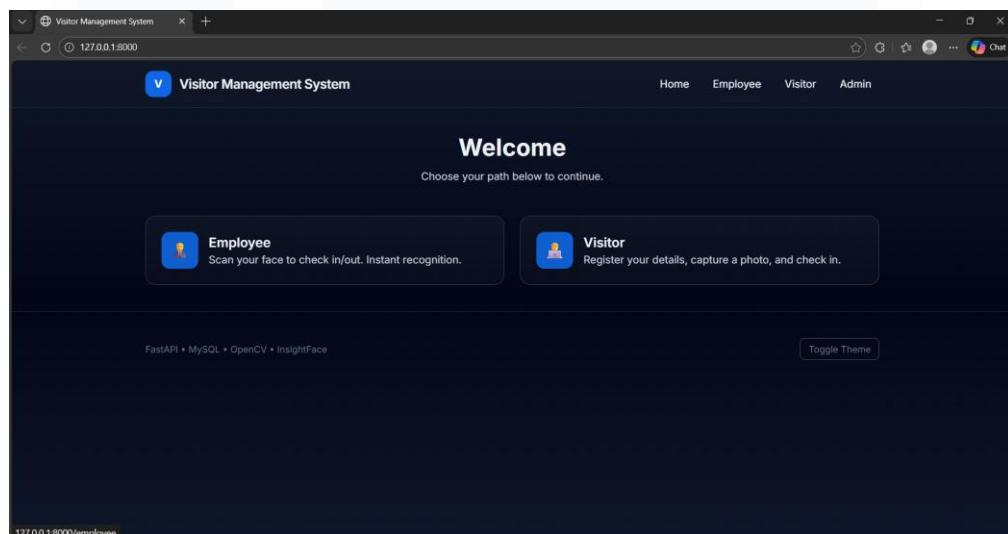
Gambar 3. 2 Business Process Diagram

*Business Process Diagram Visitor Management System* pada gambar 3.2 menggambarkan alur proses bisnis sistem secara menyeluruh yang melibatkan pengguna, antarmuka *frontend*, sistem kamera, *server* dan *backend*, layanan kecerdasan buatan, serta basis data. Proses diawali ketika karyawan atau pengunjung memasuki area lobi dan mengakses halaman absensi melalui antarmuka *frontend*. Sistem menyediakan opsi *check-in* dan *check-out* yang disesuaikan dengan peran pengguna sebagai karyawan atau pengunjung. Pengunjung diwajibkan mengisi formulir data, sedangkan karyawan langsung melanjutkan ke proses pemindaian wajah menggunakan kamera absensi. Kamera melakukan pengambilan citra wajah dan mengirimkan data gambar ke server untuk diproses lebih lanjut.

Server melakukan validasi permintaan sebelum meneruskan data ke layanan kecerdasan buatan. Layanan *Face Recognition* mencocokkan citra wajah dengan data yang tersimpan dalam basis data untuk proses identifikasi, sedangkan layanan *People Counting* memproses data video *streaming* dari kamera pemantauan untuk

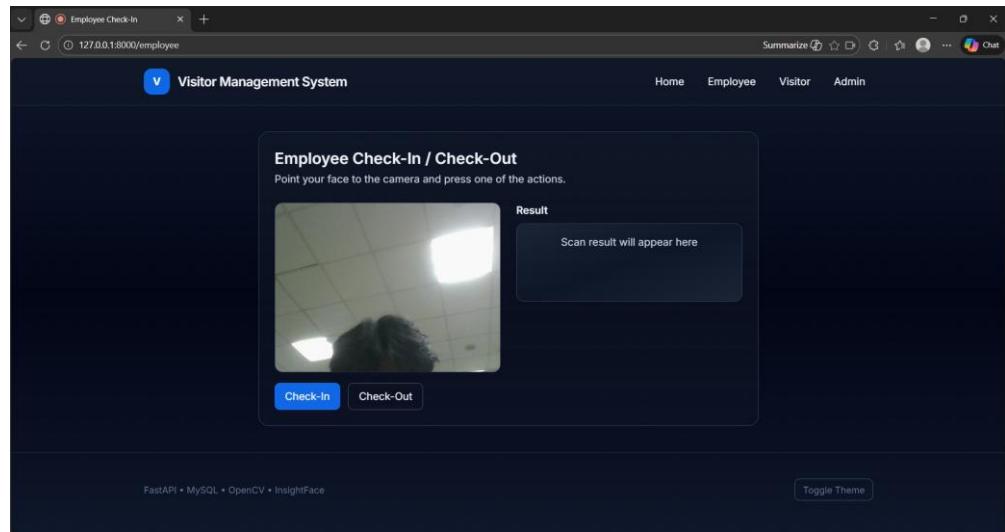
menghitung jumlah orang di area tertentu. Hasil pemrosesan dari layanan kecerdasan buatan dikirim kembali ke backend untuk dilakukan validasi respons, kemudian disimpan ke dalam basis data sebagai rekam jejak aktivitas pengunjung. Admin mengakses halaman administrasi melalui proses login untuk melihat hasil pemrosesan sistem, termasuk data kehadiran, identifikasi pengunjung, dan jumlah orang yang terdeteksi. Alur proses ini menunjukkan integrasi antar komponen sistem yang mendukung pengelolaan pengunjung secara otomatis, terstruktur, dan real-time.

Bagian dashboard website terbagi menjadi 4 bagian yaitu *home page*, *employee page*, *visitor page*, dan *admin page* seperti gambar dibawah ini.



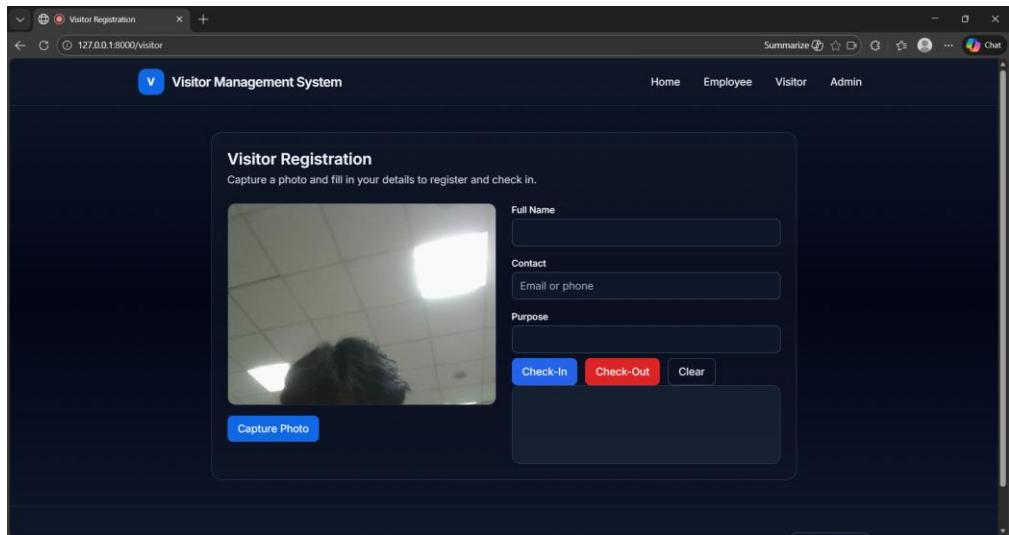
Gambar 3. 3 *Home Page*

Tampilan Home Page (gambar 3.3), pengguna diarahkan pada tampilan awal sistem yang menampilkan pilihan kategori sesuai aktivitas dan identitas pengguna, yakni Employee atau Visitor. Halaman ini dirancang sebagai gerbang interaksi utama yang menghadirkan antarmuka sederhana dan informatif agar pengguna memahami langkah awal sebelum masuk ke proses absensi.



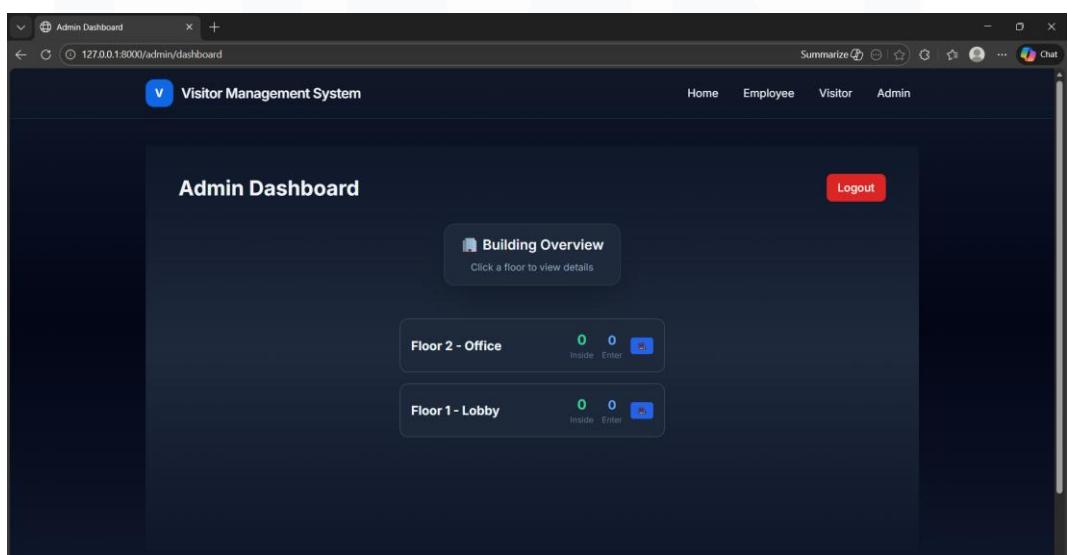
Gambar 3. 4 *Employee Page*

Selanjutnya, *Employee Page* (gambar 3.4) diperuntukkan khusus bagi karyawan atau pekerja internal perusahaan. Pada halaman ini, proses *check-in* dan *check-out* berlangsung secara otomatis melalui *Face Recognition* tanpa harus melakukan input manual seperti penulisan identitas atau pengambilan foto ulang. Hal ini dimungkinkan karena data identitas karyawan—termasuk *embedding* wajah—sudah tersimpan pada *database MySQL* sejak proses registrasi awal. Sistem melakukan akuisisi citra wajah secara *real-time* melalui kamera saat prosedur absensi karyawan berlangsung, lalu melakukan pencocokan *embedding* menggunakan model InsightFace dengan metode perhitungan *cosine similarity*.



Gambar 3. 5 *Visitor Page*

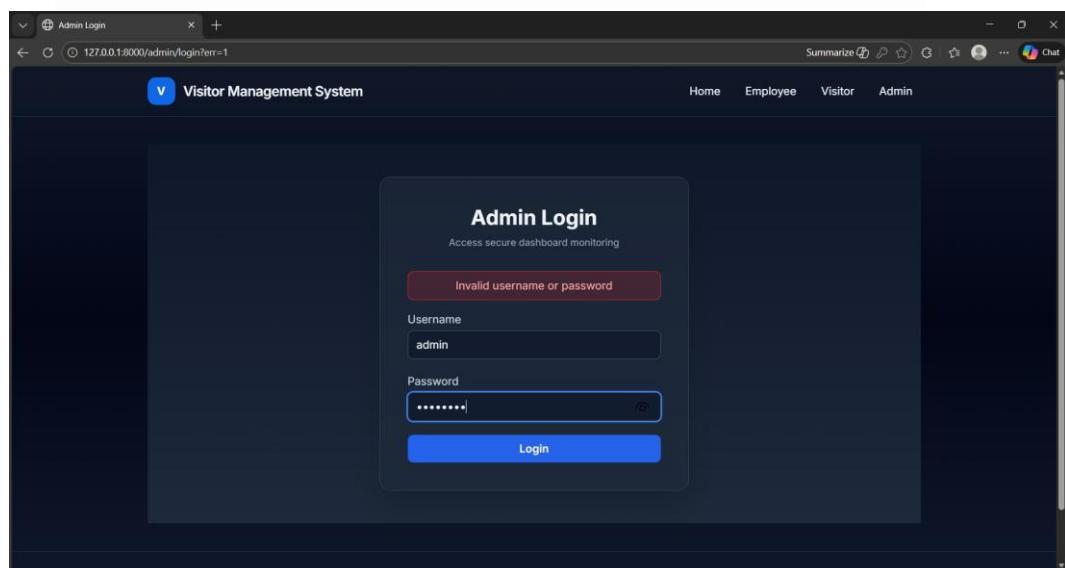
Berbeda dengan *employee page*, *Visitor Page* (gambar 3.5) dirancang untuk tamu yang datang ke perusahaan tanpa memiliki data wajah yang tersimpan sebelumnya. Pada halaman ini, pengunjung diwajibkan mengisi data diri seperti nama, kontak, dan tujuan kunjungan serta mengambil foto secara langsung melalui kamera sistem. Foto tersebut kemudian diproses menjadi *embedding* wajah yang tersimpan sementara pada database. *Embedding* inilah yang digunakan untuk proses *check-out* tamu menggunakan *face recognition* saat mereka meninggalkan area perusahaan.



Gambar 3. 6 *Admin Page*

Bagian terpenting dalam pengelolaan keseluruhan sistem berada pada *Admin Dashboard* (gambar 3.6) yang berfungsi sebagai pusat pemantauan dan kontrol terpadu. *Admin Dashboard* menyajikan tampilan *monitoring* jumlah orang di setiap lantai dengan memanfaatkan data *people counting* berbasis YOLO dan perhitungan *centroid crossing*. Informasi jumlah orang diperoleh secara otomatis setiap kali objek manusia terdeteksi melintas pada area atau garis yang telah ditentukan sistem. Masing-masing memiliki fungsi dan alur operasional yang berbeda namun saling terintegrasi dalam satu sistem manajemen absensi dan monitoring pergerakan orang. Pembagian halaman ini bertujuan untuk menciptakan pengalaman penggunaan yang tersegmentasi sesuai kebutuhan pengguna (*role-based access*), meningkatkan keamanan *access control*, serta mempermudah alur pelayanan absensi berbasis kecerdasan buatan.

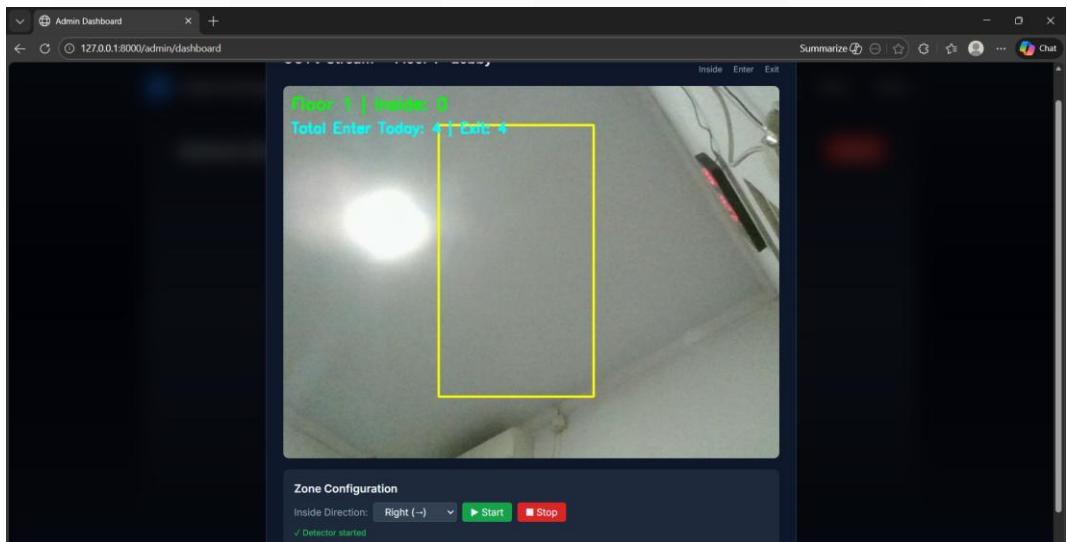
*Dashboard* admin digunakan untuk menampilkan *live CCTV streaming* seperti gambar 3.8, sebagai bagian dari pendukung keamanan gedung, memberi admin kemampuan untuk melihat kepadatan area, aktivitas pengunjung, serta pergerakan dalam konteks keselamatan dan operasional fasilitas.



Gambar 3. 7 *Admin Login Page*

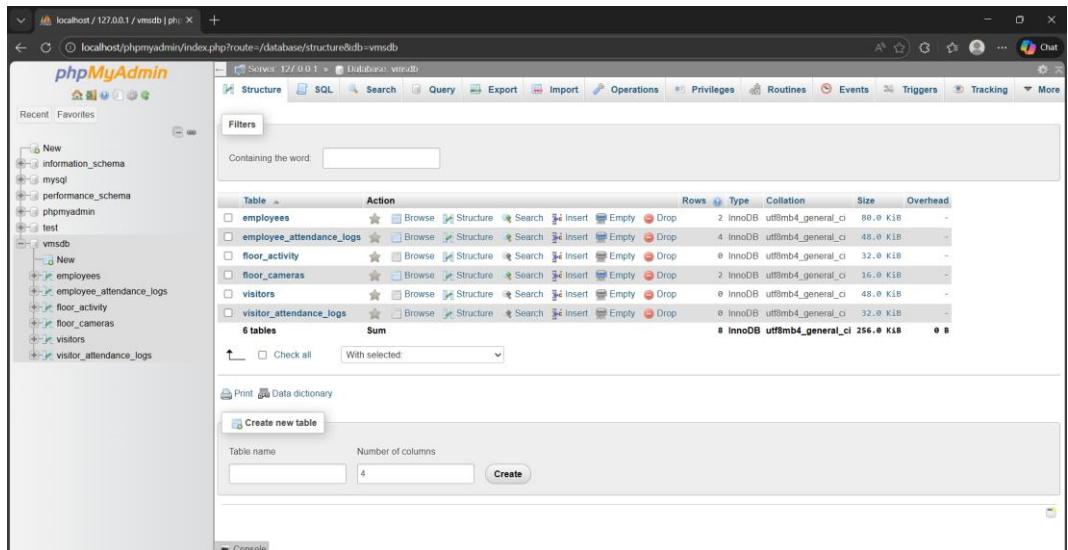
Admin Dashboard juga menyediakan *Login Page* (gambar 3.7) sebagai lapisan autentikasi untuk memastikan hanya pengguna berwenang yang dapat mengakses

data operasional, terutama data absensi, jumlah penghuni, dan rekaman aktivitas. Admin dapat mengakses fitur CCTV Streaming yang terintegrasi dengan modul deteksi saat sudah masuk untuk menampilkan area pengawasan secara *real-time*.

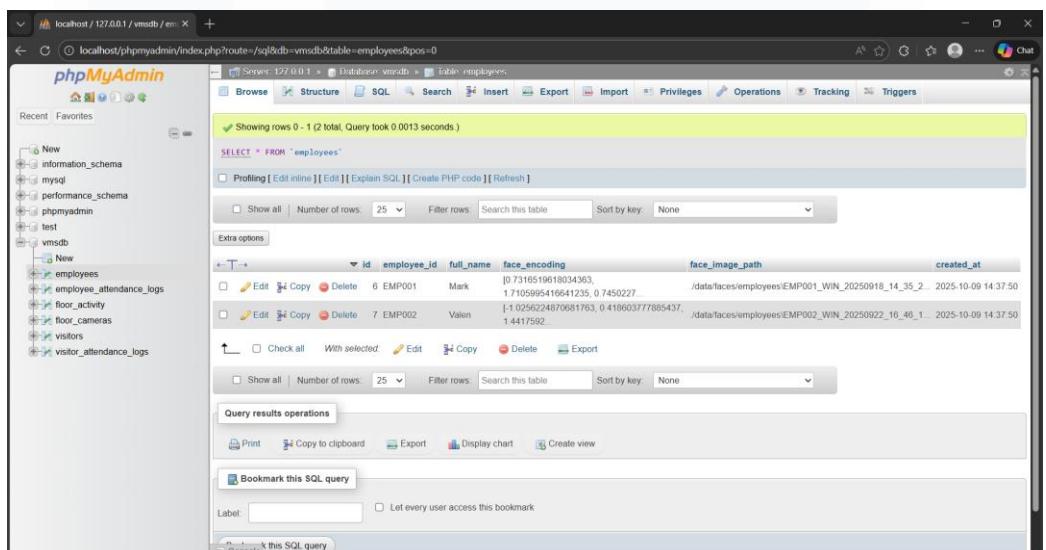


Gambar 3. 8 Admin Streaming

Tampilan streaming pada gambar 3.8 dilengkapi dengan *zone*, yaitu area yang ditetapkan secara virtual sebagai batas analisis dan keamanan yang berperan sebagai *trigger* utama dalam proses *people counting* dan identifikasi. Ketika *centroid* manusia terdeteksi berada di dalam *zone*, sistem secara otomatis menentukan arah pergerakan untuk membedakan aktivitas masuk dan keluar ruangan serta melakukan proses *cropping* pada area wajah. Proses *cropping* ini membantu sistem dalam kondisi keramaian karena setiap wajah dapat dipotong dan diproses secara terpisah sehingga mengurangi tumpang tindih antar objek. Data wajah hasil *cropping* selanjutnya diproses melalui *face detection* dan *face recognition* untuk memastikan identitas orang tersebut. Mekanisme ini memungkinkan sistem menghitung jumlah orang yang memasuki ruangan pada hari berjalan, jumlah orang yang keluar, serta jumlah orang yang berada di dalam ruangan secara *real-time*, sehingga memudahkan admin dalam melakukan pengawasan, analisis kepadatan ruangan, dan pengambilan keputusan secara cepat dan akurat.



(a)



(b)

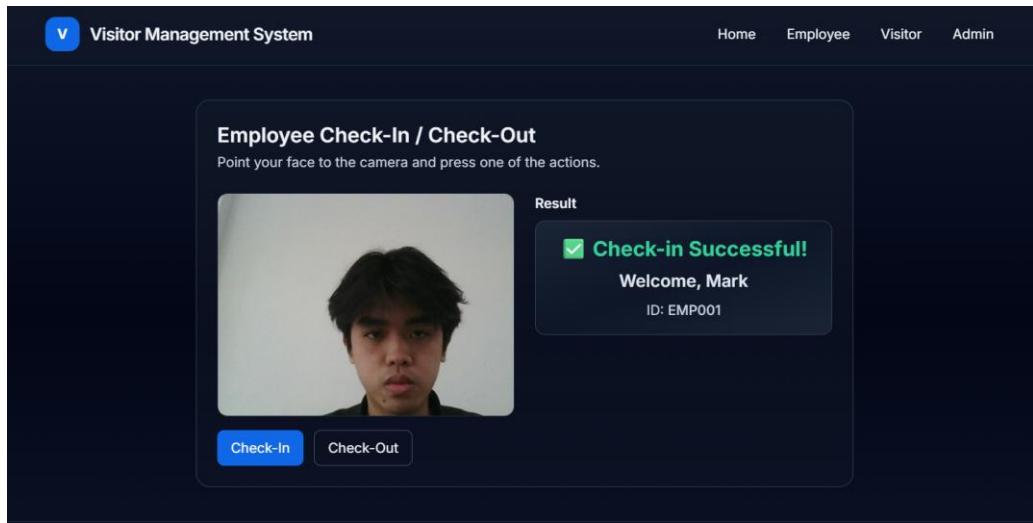
Gambar 3. 9 Tampilan Database: (a) Visitor Management System; (b) Employee

Sistem ini menggunakan database MySQL sebagai pusat pengelolaan data yang berkaitan dengan absensi, identitas pengguna, aktivitas lantai, serta sumber data pendukung pemantauan keamanan. MySQL dipilih karena memiliki kemampuan manajemen data yang stabil, terstruktur, mudah diintegrasikan dengan aplikasi berbasis web, serta mampu menangani transaksi data *real-time* dengan performa yang baik. Pada gambar 3.9, ditampilkan struktur *database* yang digunakan dalam

proyek ini, yang terdiri dari beberapa tabel utama: *employees*, *visitors*, *employee\_attendance\_logs*, *visitor\_attendance\_logs*, *floor\_activity*, dan *floor\_cameras*. Setiap tabel memiliki perannya masing-masing dalam menunjang operasional sistem *AI Visitor Management and People Counting*.

*Database* bagian *Employees* berfungsi menyimpan data identitas karyawan yang meliputi ID, nama lengkap, *embedding* wajah (*face encoding*), dan *image path* sebagai lokasi penyimpanan gambar wajah yang direkam saat registrasi. *Embedding* wajah merupakan representasi numerik hasil ekstraksi dari model InsightFace, berfungsi sebagai "sidik jari digital" yang digunakan untuk proses verifikasi wajah secara akurat. *Visitors* menyimpan data tamu yang datang ke perusahaan, mencakup informasi umum serta data wajah yang juga diproses menjadi *embedding* untuk kebutuhan *check-out*. *Employee\_attendance\_logs* dan *visitor\_attendance\_logs* mencatat aktivitas absensi masuk dan pulang secara otomatis berdasarkan *face recognition*. *Floor\_activity* digunakan untuk menyimpan data hasil *people counting* dari setiap lantai, termasuk jumlah orang yang melintas pada *safe zone* tertentu dalam rentang waktu tertentu. *Floor\_cameras* menyimpan daftar kamera beserta lokasi penempatannya untuk kebutuhan pemetaan area pemantauan.

Contoh bagian *database*, pada tabel *employees* (gambar 3.9 (b)), terdapat kolom *employee\_id*, *full\_name*, *face\_encoding*, dan *image\_path*. Kolom *id* digunakan sebagai *primary key* agar sistem dapat mengidentifikasi setiap karyawan secara unik, *name* menyimpan nama karyawan, *face\_encoding* menyimpan vektor numerik hasil perhitungan *embedding* wajah yang menjadi dasar proses kecocokan oleh model InsightFace, sementara *image\_path* menyimpan lokasi file gambar wajah yang berfungsi sebagai data pendukung proses verifikasi atau kebutuhan audit.



Gambar 3. 10 Hasil Tampilan Ketika Berhasil *Checkin*

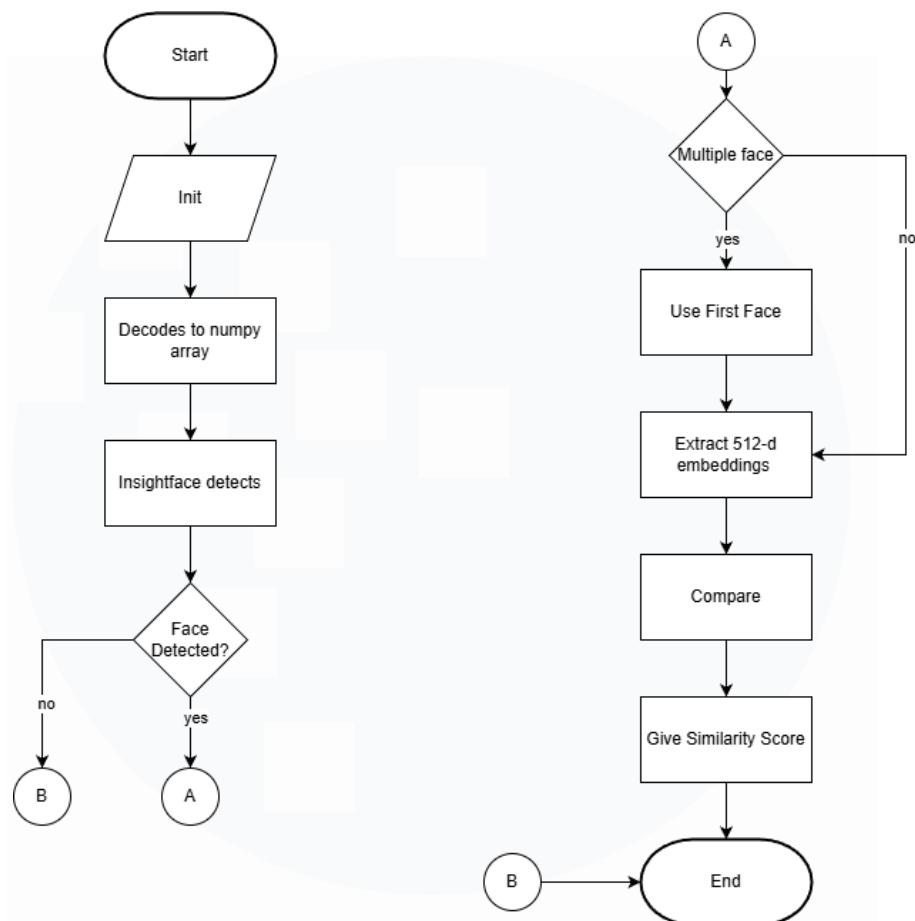
```
<class 'str'>
Mark 0.8013199776877771
INFO:    127.0.0.1:57841 - "POST /api/employee/checkin HTTP/1.1" 200 OK
INFO:    127.0.0.1:62018 - "GET / HTTP/1.1" 200 OK
□
```

Gambar 3. 11 Hasil Tampilan *Terminal* Ketika Berhasil *Checkin*

Tahap implementasi dan pengujian fitur absensi pada *Visitor Management System*, sistem menghasilkan dua jenis umpan balik yang menunjukkan bahwa proses verifikasi dan pencatatan absensi berjalan dengan benar, yaitu tampilan antarmuka pada *dashboard* pada gambar 3.10 serta log proses yang terlihat pada *terminal backend* seperti gambar 3.11. Bagian *dashboard*, pengguna—dalam hal ini karyawan—dapat melihat tampilan kamera secara *real-time* ketika melakukan proses *Check-In* atau *Check-Out*. Setelah wajah pengguna terdeteksi oleh kamera, gambar tersebut dikirimkan ke *backend* untuk diproses oleh model InsightFace. Model ini bekerja dengan mengekstraksi ciri-ciri unik pada wajah dan mengubahnya menjadi sebuah *embedding vector*, yaitu representasi numerik wajah yang memungkinkan sistem untuk melakukan proses pembandingan. *Embedding* hasil *capture* tersebut dibandingkan dengan *embedding* wajah karyawan yang telah tersimpan dalam *database MySQL*. Perbandingan dilakukan menggunakan perhitungan jarak atau *similarity score*, sehingga sistem dapat menentukan tingkat kemiripan wajah secara objektif.

Pencapaian nilai *similarity score* di atas ambang batas (*threshold*) yang telah ditentukan menjadi indikator validasi kecocokan identitas wajah dengan basis data karyawan. Mekanisme respon balik ke sisi *frontend* menampilkan status keberhasilan, semisal “Check-in Successful”, yang dilengkapi atribut nama dan ID unik karyawan (contoh: “EMP001”), sebagaimana tervisualisasi pada Gambar 3.10. Proses validasi ini turut memicu pencatatan waktu absensi secara otomatis ke dalam *database* sebagai rekam jejak aktivitas kehadiran karyawan.

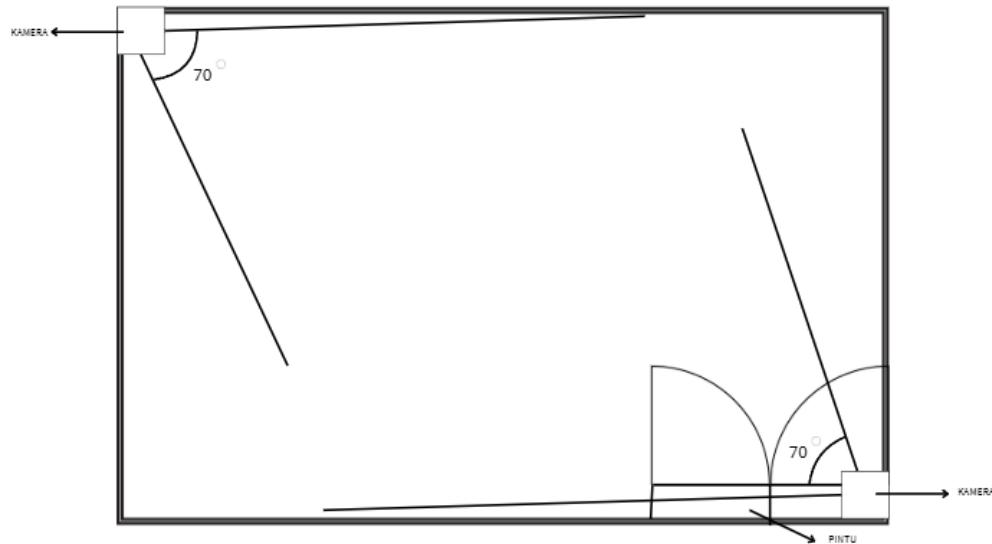
Pada sisi *backend*, terminal mencatat seluruh proses yang terjadi selama verifikasi berlangsung. Informasi seperti nilai *similarity score* yang dihitung oleh InsightFace akan ditampilkan untuk keperluan debugging maupun analisis performa model. Log seperti ”*POST /api/employee/checkin* *HTTP/1.1 200 OK*” menunjukkan bahwa permintaan dari *frontend* berhasil diterima dan diproses tanpa *error*. Terminal juga menampilkan *request GET* dari halaman utama yang menandakan bahwa *frontend* aktif berkomunikasi dengan *backend*. Dua bukti *output* tersebut—tampilan *dashboard* dan log *terminal*—dapat disimpulkan bahwa sistem absensi berbasis pengenalan wajah telah berfungsi dengan baik, mampu melakukan verifikasi identitas secara otomatis, akurat, serta memberikan pengalaman penggunaan yang intuitif dan responsif.



Gambar 3. 12 Flowchart Face Recognition

Diagram alur (*flowchart*) pada Gambar 3.12 menyajikan visualisasi proses verifikasi wajah secara detail di sisi *backend*. Fungsi utama *flowchart* ini adalah sebagai alat bantu visual tambahan untuk menjelaskan alur pemrosesan data, bukan merepresentasikan hasil tampilan antarmuka (seperti dashboard atau terminal). Diagram menunjukkan proses diawali dengan inisialisasi model dan *decoding* citra yang diterima dari *frontend* menjadi *numpy array*. Data tersebut kemudian dideteksi oleh InsightFace. Jika wajah tidak terdeteksi, proses akan dihentikan, dan sistem memberikan respons kegagalan yang ditampilkan pada *frontend*. Namun, apabila wajah terdeteksi, *flowchart* mengilustrasikan penanganan kondisi *multiple faces* dengan hanya memproses wajah pertama untuk mencegah kesalahan identifikasi. Selanjutnya, InsightFace mengekstraksi *embedding* wajah dalam vektor 512-d dan membandingkannya dengan *embedding* yang tersimpan di *database*. Hasil

perbandingan ini menghasilkan *similarity score* yang dapat dilihat pada terminal (Gambar 3.11). Dengan demikian, *flowchart* tersebut melengkapi deskripsi dengan menggambarkan proses *unseen*—mulai dari deteksi wajah hingga perhitungan *similarity*—yang menjadi dasar tampilan hasil akhir pada *dashboard* dan *backend log*.



Gambar 3. 13 Gambaran Ruangan Tampak Atas

Ilustrasi pada gambar 3.13 menunjukkan konfigurasi penempatan kamera pada ruangan dengan luas sekitar  $6 \text{ m}^2$  atau lebih untuk kebutuhan pemantauan dan deteksi wajah. Misalkan kamera yang digunakan memiliki *field of view* sebesar 70 derajat, yang merupakan sudut pandang umum pada kamera CCTV konvensional. Sudut pandang ini memungkinkan kamera memantau area tertentu secara efektif, namun berpotensi menimbulkan *blind spot* apabila hanya menggunakan satu kamera. Kondisi tersebut dapat diatasi dengan penggunaan dua kamera yang ditempatkan pada sudut ruangan yang berbeda atau dengan satu kamera 360 derajat yang dipasang di tengah ruangan sehingga seluruh area dapat terpantau secara menyeluruh. Sistem deteksi wajah bekerja optimal pada jarak 1 hingga 3 meter dari kamera karena detail wajah dapat ditangkap dengan baik, sehingga menghasilkan akurasi *face detection* dan *face recognition* yang lebih stabil dan andal. Penggunaan lebih dari satu kamera memungkinkan sistem memperoleh sudut pandang wajah

yang berbeda, sehingga meningkatkan peluang deteksi wajah yang jelas dan akurasi identifikasi.

Komunikasi antar dua kamera pada sistem ini dilakukan secara terpusat melalui server dalam satu jaringan *Local Area Network*, bukan secara langsung antar kamera. Setiap kamera CCTV mengirimkan video streaming secara kontinu ke server melalui koneksi Ethernet menggunakan protokol *streaming* seperti RTSP. Server menerima *stream* dari masing-masing kamera sebagai sumber data terpisah yang telah diberi identitas berdasarkan lokasi dan arah pandang. Data video tersebut kemudian diproses oleh modul *backend* dan layanan kecerdasan buatan untuk melakukan *people counting* dan deteksi wajah berdasarkan zona yang telah ditetapkan. Sinkronisasi antar kamera dilakukan pada tingkat server dengan membandingkan arah pergerakan dan waktu deteksi sehingga sistem dapat membedakan aktivitas masuk dan keluar tanpa terjadi duplikasi perhitungan. Pendekatan komunikasi terpusat ini memastikan kestabilan sistem, kemudahan integrasi banyak kamera, serta konsistensi data dalam proses pemantauan dan identifikasi pengunjung.

Sistem menggunakan Global ID sebagai identitas unik untuk setiap individu agar proses *people counting* tidak menghitung satu orang lebih dari satu kali pada 2 kamera atau lebih. Global ID dibentuk dari hasil akhir metode *Hierarchical Matching* yang mengintegrasikan pencocokan ciri wajah, *person re-identification* (postur tubuh, warna baju, warna celana, dsb), serta konsistensi spasial dan temporal. Setiap individu yang terdeteksi oleh kamera akan dibandingkan dengan basis data identitas yang telah ada. Individu dengan tingkat kemiripan tinggi akan diberikan Global ID yang sama meskipun terdeteksi oleh kamera yang berbeda. Pendekatan ini memastikan satu identitas tetap konsisten di seluruh sistem pemantauan.

Global ID berperan dalam pencatatan peristiwa masuk dan keluar individu secara akurat. Sistem menambah jumlah orang hanya ketika Global ID baru memasuki area pemantauan dan mengurangi jumlah orang ketika Global ID

tersebut terdeteksi keluar dari area. Perhitungan jumlah orang didasarkan pada status aktif Global ID, bukan pada jumlah deteksi kamera. Penerapan mekanisme ini meningkatkan akurasi *people counting* dan mencegah terjadinya duplikasi perhitungan untuk individu yang sama.

Melalui pengerjaan proyek ini, dapat memperoleh pengalaman komprehensif mulai dari merancang *mockup* presentasi kepada CEO, membangun *frontend* dan *backend*, mengintegrasikan model *machine learning*, serta mengoptimalkan struktur *database* hingga sistem bekerja sebagai satu kesatuan yang sesuai kebutuhan perusahaan. Latar belakang sebagai mahasiswa Teknik Elektro menjadikan proses ini sebagai tantangan sekaligus pembelajaran berharga, karena harus mempelajari konsep dan praktik *software engineering* seperti API, *debugging*, integrasi AI, serta arsitektur *database* yang sebelumnya berada di luar fokus bidang studi. Pengalaman lintas bidang yang melibatkan otomasi, sensor, kecerdasan buatan, dan pengembangan aplikasi ini tidak hanya memperluas kompetensi teknis, tetapi juga mengasah kemampuan analisis, inovasi, dan pemecahan masalah, sehingga menjadi modal penting dalam menghadapi tuntutan industri masa depan yang memerlukan perpaduan kemampuan *hardware*, *software*, dan teknologi cerdas.

### 3.3 Kendala yang Ditemukan

Kendala yang dialami selama mengerjakan projek magang visitor management system oleh penulis

1. Pada proses implementasi sistem absensi berbasis pengenalan wajah menggunakan InsightFace, salah satu kendala signifikan muncul ketika kamera mendeteksi lebih dari satu wajah dalam satu *frame*. Hal ini sering terjadi ketika lingkungan sekitar tidak dapat dikontrol sepenuhnya, seperti ketika ada pengunjung lain yang tidak sengaja masuk ke dalam area pandang kamera, atau ketika karyawan berdiri terlalu dekat dengan orang lain. InsightFace secara default akan mendeteksi seluruh wajah yang muncul pada frame dan menghasilkan *embedding* untuk masing-masing

wajah. Karena sistem Anda dirancang untuk melakukan pencocokan *embedding* hanya pada satu wajah utama, kehadiran lebih dari satu wajah menyebabkan kebingungan dalam proses verifikasi, karena model akan mengembalikan *array embedding* lebih dari satu.

2. Seiring bertambahnya jumlah karyawan dan data absensi dalam sistem, performa *query database* MySQL mulai menurun. Hal ini terutama terjadi pada proses pencocokan *embedding* wajah yang membutuhkan pembacaan data *embedding* dari setiap karyawan di *database* sebelum dibandingkan oleh InsightFace. *Embedding* wajah yang memiliki 512 nilai *floating point* per orang menjadikan data semakin berat, sehingga proses *SELECT* yang mengambil seluruh data *embedding* menjadi lambat. Kondisi ini sangat terlihat pada terminal backend yang menunjukkan waktu pemrosesan meningkat dan respons API menjadi lebih lama.
3. Sebagai mahasiswa Teknik Elektro, Anda memiliki latar belakang yang lebih dominan pada perangkat keras, rangkaian elektronika, dan sistem kontrol, sehingga ketika mulai mengembangkan sistem Visitor Management berbasis web, Anda menghadapi tantangan besar dalam memahami aspek *software engineering* yang kompleks. Beberapa kesulitan yang muncul antara lain memahami cara kerja API berbasis FastAPI, penerapan *routing*, pengelolaan *request-response*, *debugging error backend*, serta integrasi model AI InsightFace ke dalam alur *backend*. Selain itu, Anda juga perlu mempelajari konsep penyimpanan *embedding*, struktur *database* MySQL, serta bagaimana frontend JavaScript mengirim data *base64* ke *backend*.

### 3.4 Solusi atas Kendala yang Ditemukan

Berikut solusi atas kendala projek *visitor management system* yang diberikan oleh penulis:

1. Solusi yang diterapkan adalah dengan menambahkan logika penyaringan wajah pada *backend*. Berdasarkan *flowchart* yang pernah Anda lampirkan, sistem hanya akan mengambil wajah pertama yang terdeteksi oleh

InsightFace, sehingga *embedding* yang digunakan untuk proses pencocokan tetap konsisten. Langkah ini juga diperkuat dengan pemberian notifikasi pada *frontend* untuk memberitahu pengguna agar memastikan bahwa hanya satu wajah yang berada di depan kamera saat proses absensi berlangsung, sehingga meminimalkan ambiguitas pada input. Dengan demikian, proses verifikasi menjadi lebih stabil dan risiko kesalahan identifikasi dapat ditekan secara signifikan.

2. Peningkatan performa dan struktur basis data terealisasi melalui penerapan serangkaian optimasi teknis. Langkah prioritas meliputi penambahan index pada kolom dengan intensitas pencarian tinggi, khususnya ID karyawan. Implementasi indexing tersebut terbukti mengakselerasi proses pengambilan data secara signifikan. Modifikasi struktur penyimpanan *embedding* menuju tipe JSON atau LongText turut dilakukan guna meningkatkan efisiensi penyimpanan array 512 *floating point* tanpa membebani proses *parsing*. Strategi optimasi disempurnakan dengan penerapan mekanisme *caching* untuk penyimpanan *embedding* sementara di memori *backend*. Teknik ini bertujuan mengeliminasi redundansi *query* ke basis data saat data dibutuhkan secara berulang.
3. Keterbatasan pemahaman awal mengenai *software* diatasi melalui implementasi pendekatan pembelajaran bertahap. Metode ini mencakup studi dokumentasi resmi, praktik kode secara empiris, dan konsultasi intensif dengan mentor perusahaan. Implementasi *backend* dilakukan secara modular demi kemudahan pemahaman dan pemeliharaan, diikuti oleh *debugging* terstruktur menggunakan *logging* pada setiap tahap proses, serta studi pola pengembangan API modern menggunakan FastAPI. Selain itu, dikuasai pula dasar-dasar integrasi model AI dengan *backend* Python, meliputi mekanisme pembacaan berkas *base64*, konversi ke *numpy array*, hingga pemrosesannya menggunakan InsightFace. Proses tersebut menghasilkan perkembangan signifikan dalam pemahaman *software engineering*, termasuk konsep API, manajemen data, integrasi AI, dan alur

komunikasi *frontend-backend*. Pengembangan sistem utuh secara mandiri berhasil dicapai, membuktikan kapabilitas penyelesaian seluruh komponen proyek secara efektif, meskipun awalnya berada di luar kurikulum studi utama.