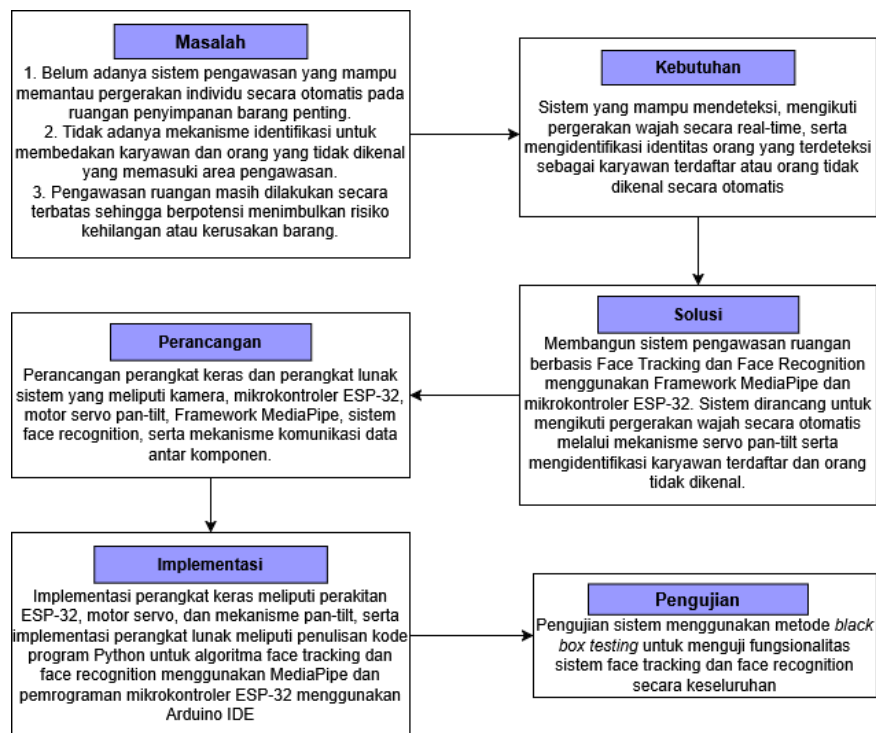


BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Kerangka Berpikir

Kerangka berpikir dalam penelitian ini digunakan untuk memberikan gambaran alur pemikiran dalam rancang bangun sistem *face tracking* menggunakan *framework MediaPipe* dan mikrokontroler *ESP-32*. Adapun langkah-langkah dalam kerangka berpikir ini sebagai berikut:



Gambar 3. 1 Kerangka Berpikir

Penelitian ini dilatarbelakangi oleh kebutuhan akan pengenalan wajah otomatis yang mampu memantau pergerakan individu secara otomatis serta mengidentifikasi identitas orang yang berada pada area pengawasan. Kerangka berpikir penelitian ini disusun berdasarkan tahapan yang dimulai dari identifikasi masalah hingga pengujian sistem yang dikembangkan.

1. Masalah

Permasalahan yang ditemukan adalah belum adanya sistem yang mampu melakukan pengenalan wajah secara otomatis pada area penyimpanan

barang penting di PT Radar Telekomunikasi Indonesia. Proses identifikasi identitas seseorang masih dilakukan secara manual sehingga informasi mengenai identitas orang yang berada di area tersebut belum dapat diperoleh secara otomatis. Selain itu, belum tersedia mekanisme yang mampu mengikuti pergerakan wajah selama proses identifikasi berlangsung sehingga proses pengenalan wajah berpotensi terganggu apabila posisi wajah berubah.

2. Kebutuhan

Berdasarkan permasalahan tersebut, dibutuhkan suatu sistem yang mampu mendeteksi wajah, mengikuti pergerakan wajah secara *real-time*, serta mengenali identitas seseorang secara otomatis berdasarkan data referensi yang telah terdaftar. Selain itu, sistem diharapkan mampu membedakan antara wajah karyawan yang telah terdaftar dengan Orang Tidak Dikenal (OTK).

3. Solusi

Solusi yang diusulkan adalah membangun sistem pengenalan wajah berbasis *face tracking* menggunakan *Framework* MediaPipe, *Framework* DeepFace dengan model ArcFace, dan mikrokontroler ESP-32. *Framework* MediaPipe digunakan untuk mendeteksi serta mengikuti pergerakan wajah, sedangkan *Framework* DeepFace dengan model ArcFace digunakan untuk melakukan proses pengenalan wajah. Selanjutnya, ESP-32 digunakan untuk mengendalikan mekanisme kamera *pan-tilt* sehingga kamera dapat mengikuti pergerakan wajah secara otomatis selama proses pengenalan berlangsung.

4. Perancangan

Tahap perancangan meliputi perancangan perangkat keras dan perangkat lunak sistem. Perancangan perangkat keras mencakup webcam, mikrokontroler ESP-32, motor servo *pan-tilt*, serta komponen pendukung lainnya. Sementara itu, perancangan perangkat lunak meliputi perancangan arsitektur sistem, antarmuka pengguna, implementasi *face tracking* menggunakan *Framework* MediaPipe, implementasi *face recognition*

menggunakan *Framework* DeepFace dengan model ArcFace, serta mekanisme komunikasi data antara aplikasi dan mikrokontroler ESP-32.

5. Implementasi

Tahap implementasi meliputi perakitan perangkat keras dan pembangunan perangkat lunak berdasarkan hasil perancangan yang telah dilakukan. Implementasi perangkat lunak dilakukan menggunakan bahasa pemrograman Python dengan mengintegrasikan *Framework* MediaPipe, *Framework* DeepFace, dan komunikasi serial, sedangkan implementasi perangkat keras dilakukan melalui pemrograman ESP-32 menggunakan Arduino IDE untuk mengendalikan motor servo *pan-tilt*.

6. Pengujian

Tahap akhir adalah pengujian sistem untuk mengevaluasi fungsionalitas dan kinerja sistem secara keseluruhan. Pengujian dilakukan terhadap kemampuan sistem dalam mendeteksi wajah (*face detection*), mengikuti pergerakan wajah (*face tracking*), mengenali identitas karyawan (*face recognition*), komunikasi antara aplikasi dengan ESP-32, serta kemampuan sistem dalam membedakan karyawan yang telah terdaftar dengan Orang Tidak Dikenal (OTK).

3.2 Metode Penelitian

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode Prototype. Prototype merupakan pendekatan pengembangan sistem yang berfokus pada pembuatan model awal (*prototype*) yang dapat dilihat, diuji, dan dievaluasi secara langsung oleh pengguna untuk mengidentifikasi kebutuhan sistem secara cepat dan tepat (Adi Kurniyanti & Murdiani, 2022). Menurut Pressman (2005), *prototyping* merupakan metode rekayasa perangkat lunak yang digunakan ketika kebutuhan pengguna belum sepenuhnya terdefinisi, sehingga pengembang dan pengguna dapat bekerja sama dalam menyempurnakan kebutuhan sistem melalui proses pembangunan dan evaluasi *prototype*. Dalam metode ini, *prototype* berfungsi sebagai media komunikasi antara pengembang dan pengguna selama proses pengembangan berlangsung. Agar pembangunan *prototype* berjalan dengan

baik, diperlukan aturan dan kesepakatan sejak tahap awal sehingga *prototype* yang dihasilkan sesuai dengan kebutuhan pengguna yang telah diidentifikasi sebelumnya (Kustanto et al., 2024).

Metode Prototype memiliki beberapa karakteristik, yaitu melibatkan pengguna secara aktif dalam proses pengembangan, menghasilkan *prototype* awal dalam waktu yang relatif singkat, serta menerapkan proses evaluasi dan penyempurnaan secara berulang (*iteratif*). Hasil evaluasi terhadap *prototype* digunakan sebagai dasar untuk memperbaiki dan menyempurnakan sistem hingga sesuai dengan kebutuhan pengguna. Dengan adanya keterlibatan pengguna pada setiap tahapan, sistem yang dikembangkan diharapkan mampu memberikan solusi yang lebih sesuai terhadap permasalahan yang dihadapi (Adi Kurniyanti & Murdiani, 2022). Oleh karena itu, metode Prototype bersifat *iteratif*, yaitu proses evaluasi dan perbaikan dilakukan secara berulang hingga diperoleh sistem yang memenuhi kebutuhan pengguna (Pressman, 2005).

Metode Prototype dipilih pada penelitian ini karena sesuai dengan karakteristik pengembangan sistem pengenalan wajah berbasis *face tracking* yang mengintegrasikan perangkat lunak dan perangkat keras. Sistem yang dikembangkan tidak hanya melibatkan implementasi algoritma *face tracking* menggunakan *Framework* MediaPipe dan *face recognition* menggunakan *Framework* DeepFace dengan model ArcFace, tetapi juga mengintegrasikan mikrokontroler ESP-32, mekanisme kamera *pan-tilt*, serta antarmuka pengguna. Proses pengembangan dilakukan secara bertahap sehingga setiap komponen dapat diuji, dievaluasi, dan disempurnakan sebelum diintegrasikan menjadi satu sistem yang utuh.

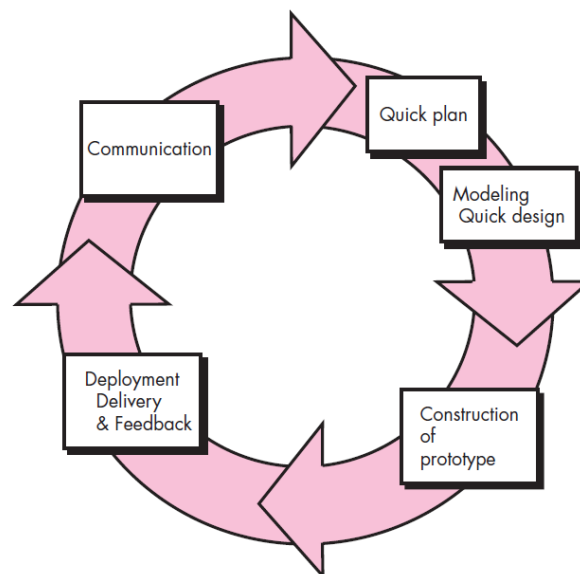
Selain itu, pengembangan sistem dilakukan melalui proses evaluasi terhadap *prototype* yang telah dibangun. Hasil evaluasi digunakan sebagai dasar dalam melakukan penyempurnaan sistem, baik pada aspek perangkat lunak maupun perangkat keras. Selama proses pengembangan dilakukan beberapa penyempurnaan terhadap *prototype* berdasarkan hasil evaluasi, seperti penyempurnaan antarmuka pengguna (*GUI*) dan penyesuaian parameter sistem agar diperoleh kinerja yang lebih optimal. Proses tersebut menunjukkan bahwa

pengembangan sistem tidak dilakukan secara langsung hingga tahap akhir, melainkan melalui beberapa kali evaluasi dan perbaikan sesuai dengan prinsip metode Prototype.

Berdasarkan karakteristik tersebut, metode Prototype dinilai sesuai untuk diterapkan pada penelitian ini karena memungkinkan pengembangan sistem dilakukan secara bertahap melalui proses perencanaan, pembangunan *prototype*, pengujian, evaluasi, dan penyempurnaan hingga diperoleh sistem pengenalan wajah berbasis *face tracking* yang sesuai dengan kebutuhan penelitian.

3.3 Prosedur Penelitian

Pada penelitian ini, prosedur penelitian dilakukan dengan menerapkan tahapan metode Prototype yang terdiri dari Communication, Quick Plan and Modelling Quick Design, Construction of Prototype, serta Deployment, Delivery and Feedback. Alur prosedur penelitian dapat dilihat pada Gambar 3.2



Gambar 3. 2 Alur Prosedur Penelitian

(Sumber: (Pressman, 2005))

Prosedur penelitian ini disusun secara sistematis untuk menggambarkan tahapan yang dilakukan dalam proses perancangan dan pengujian sistem *face tracking*. Adapun tahapan penelitian yang dilakukan adalah sebagai berikut:

1. *Communication*

Tahap ini dilakukan untuk mengidentifikasi kebutuhan sistem dan permasalahan yang menjadi dasar penelitian melalui observasi serta komunikasi dengan pihak terkait di PT Radar Telekomunikasi Indonesia. Proses observasi dilakukan pada ruang penyimpanan barang penting untuk mengetahui kondisi proses identifikasi yang sedang diterapkan, kendala yang dihadapi, serta kebutuhan terhadap sistem pengenalan wajah yang akan dikembangkan.

Selain observasi, dilakukan komunikasi dengan pihak perusahaan untuk memperoleh informasi mengenai kebutuhan pengguna terhadap sistem, seperti kemampuan sistem dalam mendeteksi wajah, mengikuti pergerakan wajah secara otomatis, serta mengenali identitas karyawan yang berada di depan kamera. Informasi yang diperoleh pada tahap ini menjadi dasar dalam menentukan kebutuhan fungsional sistem yang akan dikembangkan.

Pada tahap ini juga dilakukan pengumpulan data referensi wajah karyawan yang akan digunakan dalam proses *face recognition*. Data diperoleh dari dokumentasi kegiatan perusahaan serta pengambilan citra wajah secara langsung terhadap karyawan untuk melengkapi data yang belum mencukupi. Selanjutnya dilakukan studi literatur dengan mempelajari berbagai referensi yang berkaitan dengan *computer vision*, *face detection*, *face tracking*, *face recognition*, *Framework MediaPipe*, *Framework DeepFace*, model ArcFace, mikrokontroler ESP-32, serta metode Prototype sebagai landasan dalam perancangan dan pengembangan sistem.

Output tahap Communication adalah identifikasi permasalahan, kebutuhan pengguna, kebutuhan sistem, serta data referensi.

2. *Quick Plan*

Tahap ini merupakan tahap perencanaan yang dilakukan berdasarkan kebutuhan sistem yang telah diperoleh pada tahap *Communication*. Pada tahap ini dilakukan identifikasi kebutuhan perangkat keras (*hardware*) dan perangkat

lunak (*software*) yang akan digunakan sebagai dasar dalam pengembangan sistem pengawasan ruangan berbasis *face tracking* dan *face recognition*.

Perencanaan perangkat keras meliputi penentuan komponen yang digunakan, yaitu webcam USB sebagai perangkat akuisisi citra, mikrokontroler ESP-32 sebagai pengendali sistem, motor servo *pan-tilt* sebagai aktuator pergerakan kamera, serta komputer yang digunakan sebagai unit pemrosesan utama. Selain itu, direncanakan pula mekanisme komunikasi serial USB antara komputer dan ESP-32 sebagai media pertukaran data untuk mengendalikan pergerakan servo berdasarkan hasil deteksi wajah.

Sementara itu, perencanaan perangkat lunak meliputi penentuan bahasa pemrograman Python sebagai bahasa utama pengembangan sistem, *Framework* MediaPipe untuk proses *face detection* dan *face tracking*, *Framework* DeepFace dengan model ArcFace untuk proses *face recognition*, serta Arduino IDE sebagai perangkat lunak untuk pemrograman mikrokontroler ESP-32. Seluruh kebutuhan tersebut direncanakan agar setiap komponen dapat saling terintegrasi dalam satu sistem pengawasan ruangan.

3. *Modeling Quick Design*

Tahap *Modeling Quick Design* merupakan proses perancangan sistem berdasarkan hasil perencanaan yang telah disusun pada tahap *Quick Plan*. Perancangan dilakukan untuk memberikan gambaran mengenai arsitektur sistem, alur kerja sistem, antarmuka pengguna, serta hubungan antar komponen sebelum sistem diimplementasikan ke dalam bentuk *prototype*. Tahap ini menjadi acuan dalam proses pembangunan sistem sehingga implementasi yang dilakukan sesuai dengan kebutuhan yang telah diidentifikasi sebelumnya.

Perancangan perangkat keras meliputi penyusunan arsitektur sistem, rancangan mekanisme kamera *pan-tilt*, serta hubungan antara webcam, komputer, mikrokontroler ESP-32, dan motor servo. Sementara itu, perancangan perangkat lunak meliputi penyusunan alur proses sistem pengenalan wajah yang dimulai dari proses *face detection*, *face tracking*, *face recognition*, hingga pengiriman data posisi wajah ke mikrokontroler ESP-32 melalui komunikasi serial. Selain itu, pada tahap ini juga dilakukan

perancangan antarmuka sebagai media interaksi antara pengguna dengan sistem.

Selain itu, pada tahap ini dilakukan pemodelan sistem menggunakan Unified Modeling Language (UML) yang terdiri atas *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Deployment Diagram*. Pemodelan tersebut digunakan untuk menggambarkan interaksi antara pengguna dengan sistem, alur proses pengenalan wajah, komunikasi antar komponen, serta konfigurasi perangkat keras dan perangkat lunak yang digunakan sebagai acuan pada tahap pembangunan *prototype*.

4. *Construction of Prototype*

Tahap ini merupakan proses pembangunan *prototype* berdasarkan rancangan sistem yang telah dihasilkan pada tahap *Modeling Quick Design*. Pada tahap ini dilakukan implementasi perangkat lunak dan perangkat keras secara bertahap, kemudian setiap komponen diuji untuk memastikan fungsinya berjalan sesuai dengan rancangan sebelum diintegrasikan menjadi satu kesatuan sistem.

Implementasi perangkat lunak meliputi pengembangan aplikasi menggunakan bahasa pemrograman Python dengan mengintegrasikan *Framework* MediaPipe untuk proses *face detection* dan *face tracking*, *Framework* DeepFace dengan model ArcFace untuk proses *face recognition*, serta komunikasi serial untuk mengirimkan data posisi wajah ke mikrokontroler ESP-32. Selain itu, dilakukan implementasi antarmuka pengguna (*Graphical User Interface/GUI*) sebagai media interaksi antara operator dengan sistem.

Implementasi perangkat keras meliputi perakitan webcam sebagai perangkat akuisisi citra, mikrokontroler ESP-32 sebagai pengendali sistem, motor servo *pan-tilt* sebagai aktuator, serta integrasi seluruh komponen agar dapat bekerja sebagai satu kesatuan sistem pengawasan ruangan.

Selama proses pembangunan *prototype*, dilakukan pengujian terhadap setiap komponen yang telah diimplementasikan. Apabila masih ditemukan kekurangan atau fungsi yang belum berjalan sesuai kebutuhan, maka dilakukan

penyempurnaan terhadap *prototype* sebelum melanjutkan ke tahap berikutnya. Proses ini dilakukan agar sistem yang dibangun dapat berfungsi secara optimal sebelum dilakukan pengujian secara menyeluruh.

5. *Deployment, Delivery and Feedback*

Tahap Construction of Prototype merupakan proses pembangunan *prototype* berdasarkan rancangan sistem yang telah dihasilkan pada tahap *Modeling Quick Design*. Pada tahap ini dilakukan implementasi perangkat lunak dan perangkat keras secara bertahap sehingga *prototype* awal dapat dibangun sesuai dengan kebutuhan sistem. Setiap komponen yang telah diimplementasikan kemudian diuji untuk memastikan fungsinya berjalan sesuai dengan rancangan sebelum diintegrasikan menjadi satu kesatuan sistem.

Implementasi perangkat lunak meliputi pengembangan aplikasi menggunakan bahasa pemrograman Python dengan mengintegrasikan *Framework* MediaPipe untuk proses *face detection* dan *face tracking*, *Framework* DeepFace dengan model ArcFace untuk proses *face recognition*, serta komunikasi serial sebagai media pertukaran data antara aplikasi dan mikrokontroler ESP-32. Selain itu, dilakukan pengembangan antarmuka pengguna sebagai media interaksi antara pengguna dengan sistem dalam melakukan proses pengenalan wajah.

Implementasi perangkat keras meliputi perakitan webcam sebagai perangkat akuisisi citra, mikrokontroler ESP-32 sebagai pengendali mekanisme *pan-tilt*, motor servo sebagai aktuator pergerakan kamera, serta integrasi seluruh komponen perangkat keras dan perangkat lunak sehingga membentuk sistem pengenalan wajah berbasis *face tracking* yang dapat bekerja secara terpadu.

Selama proses pembangunan *prototype*, dilakukan pengujian terhadap setiap komponen yang telah diimplementasikan. Apabila masih ditemukan kekurangan atau fungsi yang belum berjalan sesuai dengan kebutuhan, maka dilakukan penyempurnaan terhadap *prototype* berdasarkan hasil evaluasi sebelum melanjutkan ke tahap berikutnya. Proses tersebut dilakukan secara bertahap agar setiap komponen dapat bekerja secara optimal serta

menghasilkan *prototype* yang siap memasuki tahap pengujian dan evaluasi secara menyeluruh.

3.4 Perbandingan Metode Pengembangan

Tabel 3. 1 Perbandingan Metode Pengembangan

Tahapan Pengembangan Perangkat Lunak	Prototype	Waterfall	<i>RAD (Rapid Application Development)</i>
Perencanaan Sistem (<i>Systems Planning</i>)	Tahap perencanaan dilakukan secara sederhana dengan fokus pada identifikasi kebutuhan awal sistem. Kebutuhan yang belum jelas dapat disempurnakan melalui proses pembuatan dan evaluasi prototype sehingga memungkinkan penyesuaian kebutuhan selama proses pengembangan berlangsung.	Tahap perencanaan dilakukan secara menyeluruh pada awal pengembangan sistem. Seluruh kebutuhan sistem harus didefinisikan dengan jelas karena perubahan kebutuhan pada tahap selanjutnya relatif sulit untuk dilakukan.	Tahap perencanaan difokuskan pada identifikasi kebutuhan utama sistem dalam waktu yang relatif singkat. Perencanaan dilakukan secara cepat agar proses pengembangan dapat segera dimulai.
Analisis Sistem (<i>Systems Analysis</i>)	Analisis kebutuhan dilakukan secara bertahap dan berulang berdasarkan hasil evaluasi prototype. Pendekatan ini memungkinkan kebutuhan sistem yang belum teridentifikasi pada tahap awal dapat ditemukan dan disempurnakan selama proses pengembangan.	Analisis dilakukan secara rinci pada awal pengembangan untuk menghasilkan spesifikasi kebutuhan sistem yang lengkap sebelum memasuki tahap perancangan.	Analisis dilakukan secara cepat dengan melibatkan pengguna secara aktif agar kebutuhan sistem dapat segera diterjemahkan ke dalam proses pengembangan.

Tahapan Pengembangan Perangkat Lunak	Prototype	Waterfall	<i>RAD (Rapid Application Development)</i>
Perancangan Sistem (<i>Systems Design</i>)	Perancangan dilakukan secara fleksibel melalui pembuatan desain awal yang dapat dimodifikasi berdasarkan masukan pengguna. Dengan demikian, desain sistem dapat terus disempurnakan hingga sesuai dengan kebutuhan yang diharapkan.	Perancangan sistem dilakukan secara lengkap berdasarkan hasil analisis kebutuhan. Desain yang telah dibuat menjadi acuan utama pada tahap implementasi.	Perancangan dilakukan secara cepat dengan menitikberatkan pada pembuatan model yang dapat segera diimplementasikan untuk mempercepat pengembangan sistem.
Implementasi Sistem (<i>System Implement</i>)	Implementasi dilakukan melalui pembangunan prototype yang dapat langsung diuji dan dievaluasi. Umpan balik yang diperoleh digunakan untuk melakukan perbaikan dan penyempurnaan sistem secara berulang hingga diperoleh sistem yang sesuai dengan kebutuhan pengguna.	Implementasi dilakukan setelah seluruh tahapan analisis dan perancangan selesai. Pengujian umumnya dilakukan setelah sistem berhasil dibangun secara keseluruhan.	Implementasi dilakukan secara bertahap dan cepat dengan fokus pada penyelesaian fungsi-fungsi utama sistem dalam waktu yang relatif singkat.
Pemeliharaan Sistem (<i>Systems Maintenance</i>)	Pemeliharaan dilakukan berdasarkan hasil evaluasi dan umpan balik yang diperoleh selama proses pengembangan maupun setelah sistem	Pemeliharaan dilakukan setelah sistem selesai dikembangkan dan diimplementasikan. Perubahan atau perbaikan yang dilakukan pada	Pemeliharaan dilakukan untuk memperbaiki kesalahan, menyesuaikan kebutuhan baru, serta meningkatkan kinerja sistem.

Tahapan Pengembangan Perangkat Lunak	Prototype	Waterfall	<i>RAD (Rapid Application Development)</i>
	diimplementasikan. Pendekatan ini memungkinkan perbaikan dan penyempurnaan sistem dilakukan secara berkelanjutan sehingga sistem dapat terus disesuaikan dengan kebutuhan pengguna.	tahap ini umumnya memerlukan proses yang cukup kompleks karena sistem telah dibangun berdasarkan spesifikasi yang telah ditetapkan sebelumnya.	Karena pengembangan dilakukan secara cepat, pemeliharaan diperlukan untuk memastikan sistem tetap berjalan sesuai kebutuhan pengguna.

3.5 Identifikasi Kebutuhan Sistem

Berdasarkan tahapan Quick Plan pada metode Prototype, dilakukan identifikasi kebutuhan sistem yang akan digunakan dalam pengembangan sistem pengenalan wajah berbasis *face tracking*. Identifikasi kebutuhan bertujuan untuk menentukan perangkat keras (*hardware*) dan perangkat lunak (*software*) yang diperlukan agar seluruh komponen sistem dapat saling terintegrasi sesuai dengan kebutuhan yang telah diperoleh pada tahap *Communication*. Hasil identifikasi kebutuhan ini menjadi dasar dalam proses perancangan sistem pada tahap Modeling Quick Design.

3.5.1 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras pada penelitian ini merupakan komponen yang digunakan untuk mendukung proses akuisisi citra, pemrosesan data, komunikasi antarperangkat, serta pengendalian mekanisme kamera *pan-tilt*. Adapun perangkat keras yang digunakan dalam penelitian ini dapat dilihat pada Tabel 3.2.

Tabel 3. 2 Kebutuhan Perangkat Keras

No	Perangkat	Fungsi
1	PC/Laptop	Pemrosesan Sistem <i>face recognition</i> dan <i>face tracking</i>
2	Webcam USB	Akuisisi citra wajah
3	Mikrokontroler ESP-32	Pengendali servo

4	Servo SG90	Pan-Tilt kamera
5	Bracket Pan Tilt	Dudukan servo dan kamera
6	Kabel USB	Komunikasi Serial

Berdasarkan Tabel 3.2, perangkat keras yang dipilih memiliki fungsi yang saling mendukung dalam proses pengembangan sistem. Webcam digunakan sebagai perangkat akuisisi citra, komputer sebagai pusat pemrosesan sistem, ESP-32 sebagai pengendali mekanisme *pan-tilt*, sedangkan motor servo berfungsi menggerakkan kamera agar dapat mengikuti pergerakan wajah secara *real-time*.

3.5.2 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak meliputi perangkat lunak dan *framework* yang digunakan dalam proses pengembangan sistem, mulai dari implementasi *face detection*, *face tracking*, *face recognition*, hingga pemrograman mikrokontroler ESP-32. Adapun perangkat lunak yang digunakan dapat dilihat pada Tabel 3.x.

Tabel 3. 3 Kebutuhan Perangkat Lunak

No	Perangkat	Fungsi
1	Python	Bahasa pemrograman yang dipakai sistem
2	MediaPipe	<i>Face Detection</i> dan <i>Face Tracking</i>
3	DeepFace model ArcFace	<i>Face Recognition</i>
4	Arduino IDE	Program ESP-32
5	VS Code	Kode Editor Program

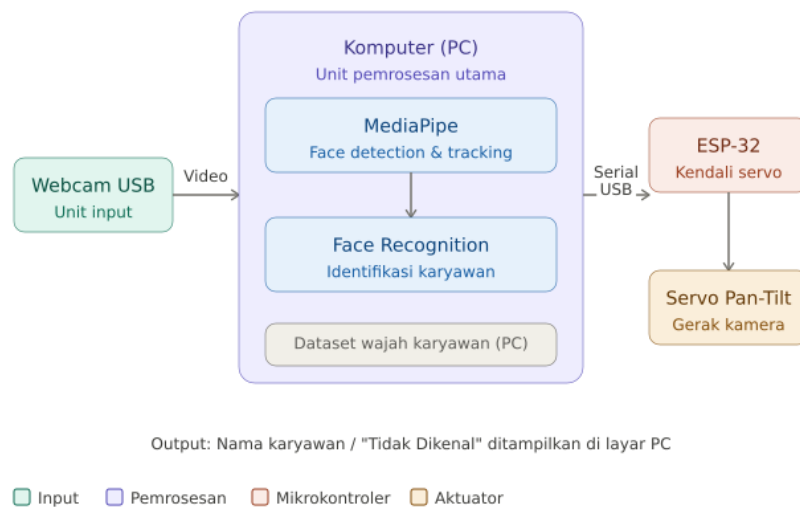
3.6 Rancangan Sistem

Berdasarkan tahapan Modeling Quick Design pada metode Prototype, dilakukan proses perancangan sistem sebagai acuan dalam pembangunan *prototype*. Pada tahap ini disusun berbagai rancangan yang menggambarkan cara kerja sistem sebelum diimplementasikan, meliputi arsitektur sistem, diagram blok sistem, pemodelan menggunakan Unified Modeling Language (UML), perancangan perangkat keras, perancangan perangkat lunak, serta perancangan antarmuka pengguna. Seluruh hasil perancangan tersebut menjadi dasar pada tahap

Construction of Prototype dalam membangun sistem pengenalan wajah berbasis *face tracking*.

3.6.1 Arsitektur Sistem

Arsitektur sistem digunakan untuk menggambarkan hubungan antar komponen serta alur kerja sistem yang dikembangkan. Pada penelitian ini, sistem pengawasan ruangan terdiri atas kamera sebagai perangkat *input*, *framework* MediaPipe yang digunakan untuk proses *Face Tracking* dan *Face Recognition*, mikrokontroler ESP-32 sebagai pengendali, serta motor servo pan-tilt sebagai aktuator.



Gambar 3.3 Arsitektur Sistem

Berdasarkan Gambar 3.3, sistem terdiri dari empat bagian utama yaitu perangkat *input*, unit pemrosesan, mikrokontroler, dan aktuator. Webcam USB berfungsi sebagai perangkat input yang digunakan untuk menangkap citra secara *real-time* dan mengirimkannya ke komputer untuk diproses.

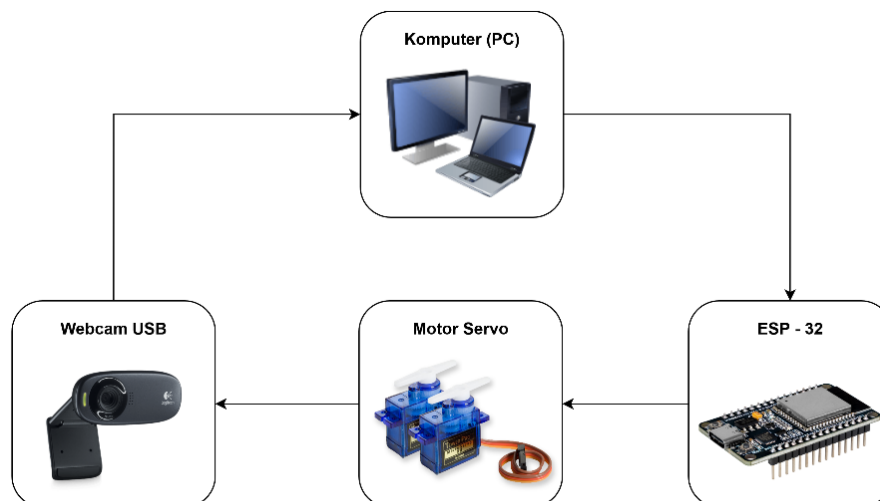
Pada unit pemrosesan, *framework* MediaPipe digunakan untuk melakukan proses *face detection* dan *face tracking* guna mendeteksi keberadaan wajah serta menentukan posisi wajah yang terdeteksi pada setiap *frame*. Selanjutnya, hasil deteksi wajah diproses oleh modul *Face Recognition* dengan membandingkan wajah yang terdeteksi terhadap dataset wajah karyawan yang telah tersimpan pada komputer. Hasil proses identifikasi kemudian digunakan untuk menentukan apakah

individu yang terdeteksi merupakan karyawan yang terdaftar atau orang yang tidak dikenal.

Selain melakukan identifikasi, sistem juga mengirimkan informasi posisi wajah ke mikrokontroler ESP-32 melalui komunikasi serial USB. Data tersebut digunakan oleh ESP-32 untuk mengendalikan pergerakan motor servo pan-tilt sehingga kamera dapat mengikuti pergerakan wajah secara otomatis. Dengan demikian, sistem tidak hanya mampu melakukan pelacakan wajah secara real-time, tetapi juga dapat mengidentifikasi individu yang berada pada area pengawasan. Hasil identifikasi berupa nama karyawan atau status "Tidak Dikenal" ditampilkan pada layar komputer sebagai informasi bagi pengguna sistem.

3.6.2 Perancangan Perangkat Keras

Perancangan perangkat keras dilakukan untuk menentukan komponen yang digunakan serta hubungan antar perangkat dalam sistem pengenalan wajah berbasis *face tracking*. Perangkat keras yang digunakan terdiri dari webcam USB sebagai perangkat input, mikrokontroler ESP-32 sebagai pengendali sistem, dan motor servo pan-tilt sebagai aktuator yang berfungsi menggerakkan kamera mengikuti pergerakan wajah yang terdeteksi.



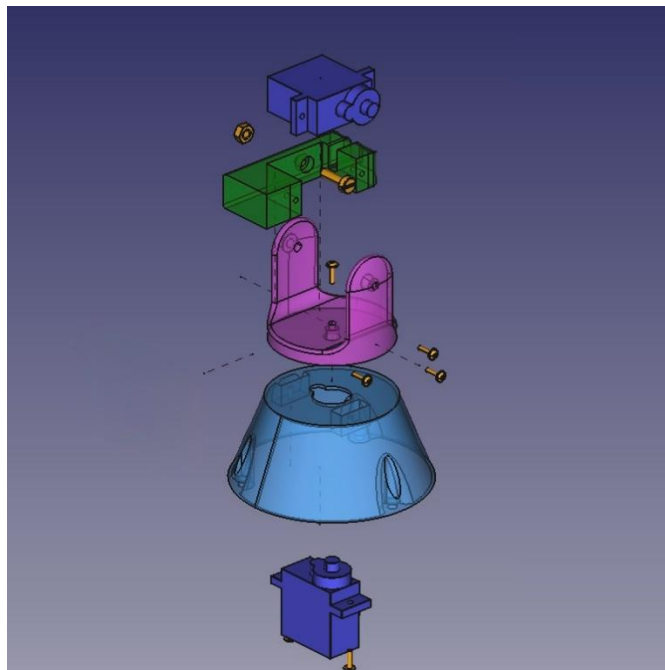
Gambar 3. 4 Rancangan Perangkat Keras

Gambar 3.4 menunjukkan rancangan perangkat keras yang digunakan pada sistem. Webcam USB berfungsi sebagai perangkat input yang digunakan untuk

menangkap citra wajah secara *real-time* dan mengirimkannya ke komputer sebagai unit pemrosesan utama. Komputer menjalankan proses *Face Tracking* dan *Face Recognition*, kemudian mengirimkan data posisi wajah ke mikrokontroler ESP-32 melalui komunikasi serial USB. Selanjutnya, ESP-32 mengendalikan motor servo pan-tilt berdasarkan data yang diterima sehingga kamera dapat bergerak mengikuti pergerakan wajah yang terdeteksi.

a. Perancangan Mekanisme *Pan-Tilt Bracket*

Mekanisme *pan-tilt bracket* dirancang sebagaiudukan kamera yang dapat bergerak ke arah horizontal (*pan*) dan vertikal (*tilt*) secara bersamaan. Desain *pan-tilt bracket* yang digunakan dalam penelitian ini diadaptasi dari desain yang tersedia secara terbuka di platform Thingiverse (eltitomanolo, 2017) dan dilakukan penyesuaian sesuai dengan kebutuhan sistem. Komponen bracket dicetak menggunakan teknologi *3D printing*.



Gambar 3. 5 *Pan-Tilt Bracket*

(Sumber: <https://www.thingiverse.com/thing:2467743>)

Berdasarkan Gambar 3.5, *pan-tilt bracket* terdiri dari beberapa komponen utama yaituudukan atas (*tilt mount*) yang berfungsi sebagai tempat pemasangan kamera dan motor servo *tilt*, penghubung tengah yang menyatukan mekanisme *pan*

dan *tilt*, serta alas (*base*) berbentuk kerucut yang menopang keseluruhan struktur sekaligus menjadi tempat pemasangan motor servo *pan*. Seluruh komponen dirakit menggunakan baut pengikat untuk memastikan kestabilan struktur saat sistem beroperasi.

3.6.3 Persiapan Dataset

Persiapan dataset dilakukan untuk menyediakan data referensi yang digunakan pada proses *face recognition* dalam mengenali identitas karyawan. Dataset terdiri atas kumpulan citra wajah karyawan PT Radar Telekomunikasi Indonesia yang diperoleh dari dokumentasi kegiatan perusahaan, serta pemotretan langsung terhadap karyawan untuk melengkapi data yang belum mencukupi. Setiap citra kemudian dikelompokkan berdasarkan identitas masing-masing karyawan sehingga membentuk dataset yang terstruktur.



Gambar 3. 6 Folder Wajah Karyawan

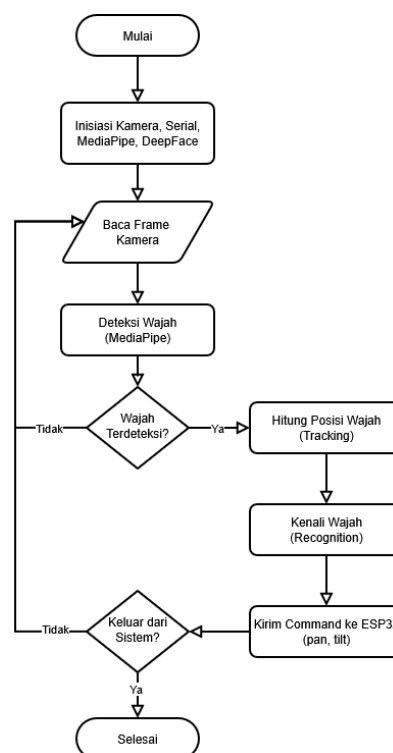
Dataset yang telah disusun selanjutnya diproses menggunakan model ArcFace untuk menghasilkan *embedding* atau representasi numerik dari setiap wajah. Hasil proses tersebut disimpan sebagai data referensi yang akan dibandingkan dengan wajah yang terdeteksi oleh sistem. Melalui proses ini, sistem dapat mengidentifikasi apakah wajah yang terdeteksi merupakan karyawan yang telah terdaftar atau individu yang tidak dikenal.

Pada penelitian ini, jumlah citra wajah yang digunakan untuk setiap individu diupayakan minimal 20 citra agar setiap wajah memiliki variasi representasi, seperti perubahan pose, arah pandangan, dan ekspresi. Penggunaan beberapa citra referensi untuk setiap individu bertujuan menghasilkan *embedding* wajah yang lebih representatif sehingga proses pencocokan pada tahap face recognition dapat dilakukan dengan lebih stabil. Apabila jumlah citra yang digunakan lebih banyak dan memiliki variasi yang memadai, maka representasi karakteristik wajah setiap

individu akan semakin lengkap sehingga berpotensi meningkatkan performa pengenalan wajah.

3.6.4 Perancangan Perangkat Lunak

Perancangan perangkat lunak menggambarkan alur kerja sistem dalam bentuk *flowchart*, mulai dari proses inisialisasi, deteksi dan *face tracking* menggunakan MediaPipe, *face recognition* menggunakan DeepFace dengan model ArcFace, hingga pengiriman perintah ke ESP-32 untuk mengendalikan motor servo. Flowchart perancangan perangkat lunak ditunjukkan pada Gambar 3.7.



Gambar 3. 7 Flowchart

Berdasarkan Gambar 3.7, proses diawali dengan inisialisasi kamera, komunikasi serial, MediaPipe, dan DeepFace. Selanjutnya, sistem membaca *frame* dari kamera dan melakukan deteksi wajah menggunakan MediaPipe. Jika wajah tidak terdeteksi, sistem akan kembali membaca *frame* berikutnya. Apabila wajah terdeteksi, sistem menghitung posisi wajah untuk proses *face tracking*, kemudian melakukan *face recognition* guna mengidentifikasi wajah yang terdeteksi. Hasil perhitungan posisi wajah selanjutnya dikirim ke ESP-32 melalui komunikasi serial

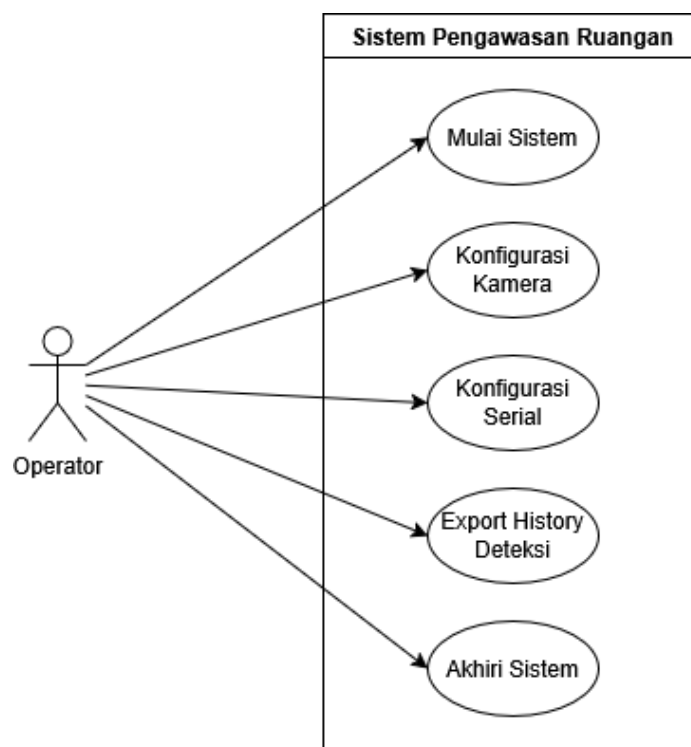
untuk menggerakkan servo pan-tilt sehingga kamera dapat mengikuti pergerakan wajah. Proses ini berlangsung secara berulang selama sistem dijalankan.

a. Pemodelan Sistem (UML)

Pemodelan sistem dilakukan menggunakan *Unified Modeling Language* (UML) untuk menggambarkan struktur dan perilaku sistem secara visual. *UML* digunakan sebagai alat bantu dalam memahami alur kerja sistem, interaksi antar komponen, serta hubungan antara aktor dan sistem yang dikembangkan. Dalam penelitian ini, pemodelan sistem menggunakan empat jenis diagram *UML*, yaitu *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Deployment Diagram*.

1. Use Case Diagram

Use Case Diagram digunakan untuk menggambarkan interaksi antara operator dengan sistem pengawasan ruangan. Diagram ini menunjukkan fungsi-fungsi utama yang dapat dilakukan oleh operator selama menggunakan sistem, mulai dari menjalankan sistem, melakukan konfigurasi, hingga mengakhiri sistem.



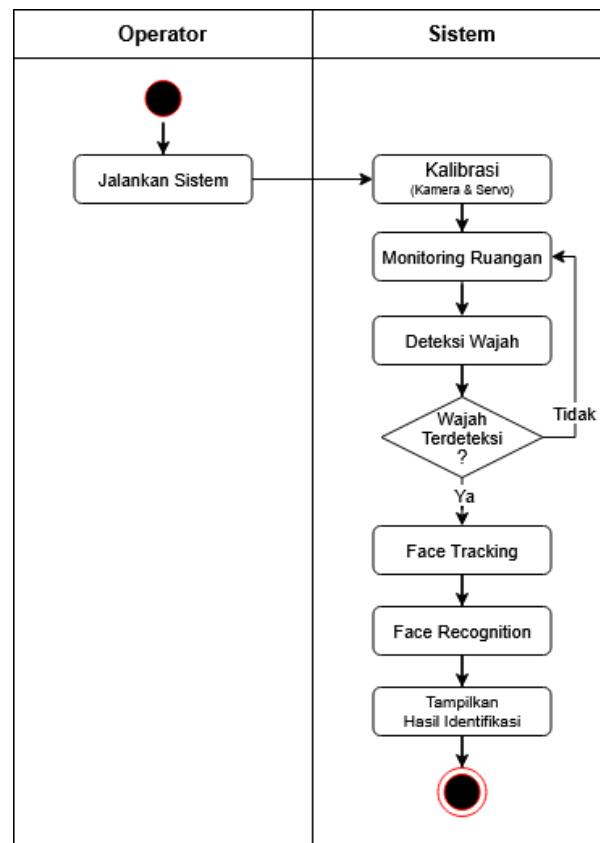
Gambar 3. 8 *Use Case Diagram*

Berdasarkan Gambar 3.8, Use Case Diagram menunjukkan bahwa sistem memiliki satu aktor, yaitu operator, yang berinteraksi secara langsung dengan sistem pengawasan ruangan. Operator memiliki hak akses untuk menjalankan sistem, melakukan konfigurasi kamera, mengatur komunikasi serial yang terhubung, mengekspor riwayat deteksi, serta mengakhiri sistem. Seluruh fungsi tersebut merupakan fitur utama yang disediakan oleh sistem untuk mendukung proses pengawasan ruangan berbasis *Face Tracking* dan *Face Recognition*.

Fungsi Mulai Sistem digunakan untuk mengaktifkan proses pengawasan ruangan. Konfigurasi Kamera berfungsi untuk memilih perangkat kamera yang akan digunakan, sedangkan Konfigurasi Serial digunakan untuk mengatur komunikasi antara aplikasi dengan mikrokontroler ESP-32 melalui port serial. Selanjutnya, Export History Deteksi memungkinkan operator menyimpan riwayat hasil deteksi untuk kebutuhan dokumentasi atau evaluasi. Adapun fungsi Akhiri Sistem digunakan untuk menghentikan proses pengawasan dan menutup aplikasi secara aman.

2. Activity Diagram

Activity Diagram digunakan untuk menggambarkan alur aktivitas yang terjadi pada sistem pengawasan ruangan berbasis *Face Tracking* dan *Face Recognition*. Diagram ini menunjukkan urutan proses yang dilakukan sistem mulai dari inisialisasi sistem, deteksi wajah, pelacakan wajah, identifikasi individu, hingga menampilkan hasil identifikasi berdasarkan wajah yang terdeteksi.



Gambar 3. 9 Activity Diagram

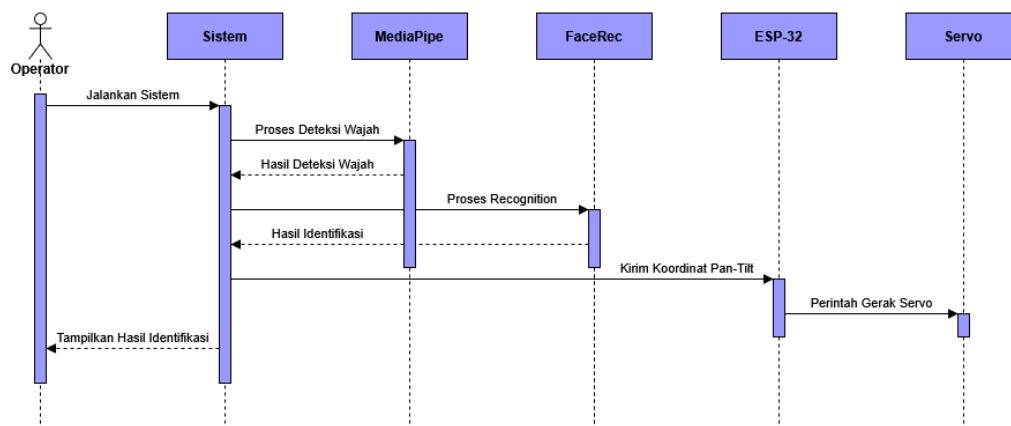
Berdasarkan Gambar 3.9, proses dimulai ketika operator menjalankan sistem. Setelah sistem aktif, dilakukan kalibrasi kamera dan *servo* untuk memastikan seluruh komponen dapat berfungsi dengan baik. Selanjutnya sistem melakukan monitoring ruangan secara *real-time* melalui *webcam* yang terhubung ke komputer.

Pada tahap berikutnya, sistem melakukan deteksi wajah terhadap citra yang diperoleh dari kamera. Jika tidak ditemukan wajah, sistem akan kembali melakukan monitoring ruangan dan proses deteksi secara berulang. Namun apabila wajah berhasil terdeteksi, sistem akan melanjutkan proses *Face Tracking* untuk menentukan posisi wajah serta mengikuti pergerakannya.

3. Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan interaksi dan pertukaran informasi antar komponen yang terlibat dalam sistem pengawasan ruangan. Melalui diagram ini, dapat diketahui urutan komunikasi yang terjadi mulai

dari saat sistem dijalankan, proses deteksi dan identifikasi wajah, hingga pengendalian pergerakan kamera menggunakan mikrokontroler *ESP-32*.



Gambar 3. 10 *Sequence Diagram*

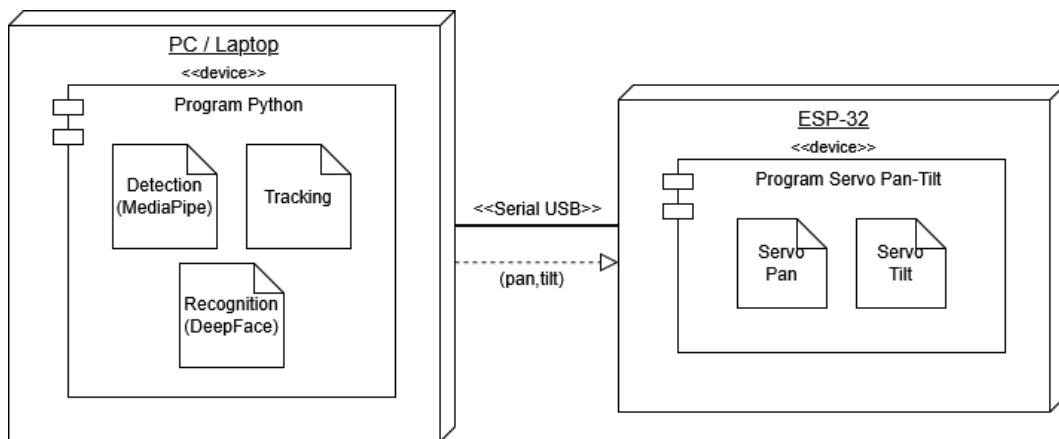
Berdasarkan Gambar 3.10, alur dimulai ketika operator menjalankan sistem. Setelah aktif, citra yang ditangkap oleh kamera diproses menggunakan *framework MediaPipe* untuk mendeteksi keberadaan wajah pada area pengawasan. Hasil deteksi yang diperoleh berupa informasi posisi wajah yang selanjutnya digunakan sebagai dasar dalam proses pelacakan dan identifikasi.

Data wajah yang berhasil terdeteksi kemudian diteruskan ke modul *Face Recognition* untuk dilakukan pencocokan dengan dataset wajah karyawan yang telah tersimpan. Dari proses tersebut diperoleh hasil identifikasi yang menunjukkan apakah wajah yang terdeteksi merupakan karyawan yang terdaftar atau individu yang tidak dikenal.

Selain digunakan untuk proses identifikasi, informasi posisi wajah juga dimanfaatkan untuk mengendalikan pergerakan kamera. Koordinat wajah dikirimkan ke mikrokontroler *ESP-32* melalui komunikasi serial, kemudian diterjemahkan menjadi perintah untuk menggerakkan *motor servo pan-tilt*. Mekanisme ini memungkinkan kamera mengikuti pergerakan wajah secara otomatis sehingga objek tetap berada dalam area pengamatan. Hasil identifikasi yang diperoleh selanjutnya ditampilkan kepada operator sebagai informasi pemantauan ruangan.

4. Deployment Diagram

Deployment Diagram digunakan untuk menggambarkan hubungan antara perangkat keras (*hardware*) dan perangkat lunak (*software*) yang terlibat dalam sistem *face tracking*. Melalui diagram ini, dapat diketahui bagaimana artefak perangkat lunak didistribusikan ke dalam *node* perangkat keras, serta bagaimana komunikasi antar *node* berlangsung dalam sistem yang dirancang.



Gambar 3. 11 *Deployment Diagram*

Berdasarkan Gambar 3.11, sistem terdiri dari dua *node* utama yaitu PC/Laptop dan mikrokontroler *ESP-32*. Pada *node* PC/Laptop, terdapat *package* Program Python yang memuat tiga artefak utama, yaitu *Detection* yang memanfaatkan *framework* *MediaPipe* untuk mendeteksi wajah secara *real-time* melalui webcam, *Tracking* yang bertugas menghitung posisi wajah terhadap pusat frame, serta *Recognition* yang menggunakan *framework* *DeepFace* dengan model *ArcFace* untuk melakukan pengenalan wajah.

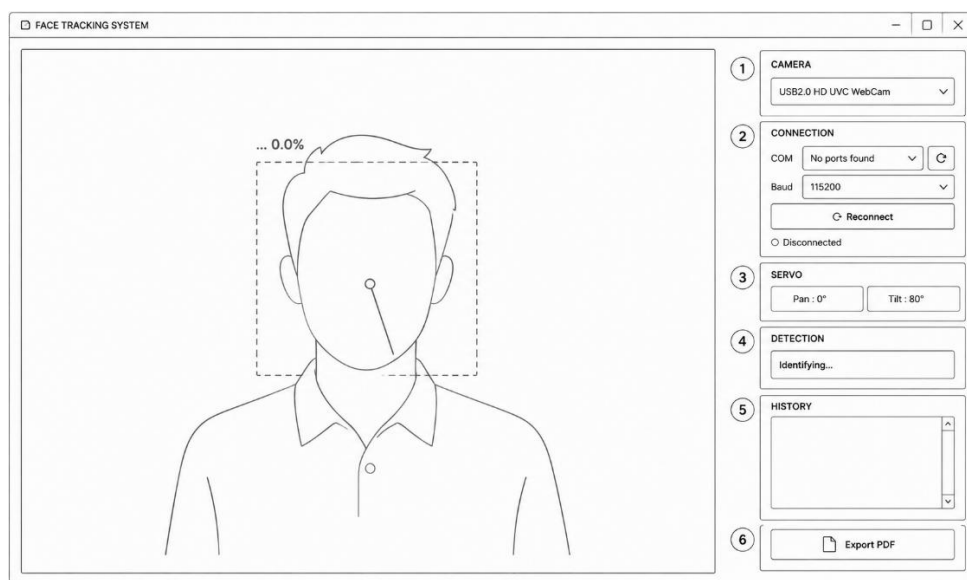
Hasil perhitungan posisi wajah dari proses *tracking* kemudian dikirimkan ke *node* *ESP-32* melalui koneksi *Serial USB* dalam format perintah (*pan, tilt*) yang merepresentasikan sudut pergerakan kamera. Pada *node* *ESP-32*, *package* Program *Servo Pan-Tilt* menerima perintah tersebut dan mengolahnya untuk mengendalikan dua artefak aktuator yaitu *Servo Pan* dan *Tilt*, sehingga kamera yang terpasang pada mekanisme *pan-tilt bracket* dapat bergerak mengikuti posisi wajah yang terdeteksi secara otomatis.

b. Perancangan Antarmuka

Perancangan antarmuka dilakukan untuk memberikan gambaran mengenai tampilan aplikasi yang digunakan sebagai media interaksi antara pengguna dengan sistem pengenalan wajah berbasis *face tracking*. Antarmuka dirancang agar pengguna dapat mengoperasikan sistem dengan mudah, mulai dari memilih perangkat kamera, menghubungkan aplikasi dengan mikrokontroler ESP-32, memantau proses *face detection*, *face tracking*, dan *face recognition*, hingga melihat riwayat hasil identifikasi yang dilakukan oleh sistem. Selain itu, antarmuka juga dirancang untuk menampilkan informasi sistem secara *real-time* sehingga memudahkan pengguna dalam melakukan pemantauan terhadap kinerja sistem selama proses pengenalan wajah berlangsung.

1. Wireframe

Wireframe antarmuka merupakan rancangan awal tampilan aplikasi yang digunakan sebagai acuan dalam proses pengembangan antarmuka sistem. Perancangan ini bertujuan untuk menggambarkan tata letak setiap komponen antarmuka sebelum diimplementasikan ke dalam aplikasi. Dengan adanya wireframe, proses pengembangan antarmuka menjadi lebih terstruktur karena fungsi dan posisi setiap komponen telah direncanakan terlebih dahulu sesuai dengan kebutuhan sistem.



Gambar 3. 12 Wireframe Antarmuka Sistem

Berdasarkan Gambar 3.12, *wireframe* antarmuka sistem terdiri atas beberapa komponen utama yang dirancang untuk mendukung proses pengoperasian sistem pengenalan wajah berbasis *face tracking*. Area utama antarmuka berupa tampilan kamera digunakan untuk menampilkan hasil tangkapan webcam secara *real-time*, yang dilengkapi dengan informasi hasil *face detection*, *face tracking*, dan *face recognition*. Di sampingnya terdapat panel kamera untuk memilih perangkat webcam yang digunakan serta panel koneksi yang berfungsi menghubungkan aplikasi dengan mikrokontroler ESP-32 melalui komunikasi serial USB dengan menyediakan pilihan COM Port, baud rate, tombol Reconnect, serta indikator status koneksi.

Selain itu, antarmuka juga dilengkapi dengan panel servo yang menampilkan informasi posisi sudut Pan dan Tilt secara *real-time*, panel Detection yang memberikan informasi status proses pengenalan wajah, serta panel History yang menampilkan riwayat hasil identifikasi selama aplikasi berjalan. Pada bagian bawah antarmuka disediakan tombol Export PDF yang digunakan untuk mengeksport data riwayat hasil pengenalan wajah ke dalam dokumen PDF sebagai dokumentasi atau laporan hasil identifikasi. Seluruh komponen tersebut dirancang agar pengguna dapat mengoperasikan dan memantau kinerja sistem melalui satu tampilan antarmuka yang sederhana dan mudah digunakan.