

BAB III METODE PENELITIAN

3.1 Desain Penelitian

Penulis menggunakan metodologi Design and Development (DnD) untuk mengembangkan dan menguji sistem kunci pintu digital yang memanfaatkan teknik *encoding* Base64 *Shuffle* dan pengenalan wajah menggunakan LBP. Metodologi ini dimulai dengan analisis kebutuhan sistem pada sistem kunci pintu, kemudian mendefinisikan tujuan spesifik untuk mengatasi masalah tersebut melalui pengembangan sistem yang lebih aman dan efisien. Metodologi ini efektif dalam menentukan area penelitian yang relevan dan mengembangkan argumen yang kuat, sambil menyediakan kerangka yang mendukung inovasi (Ellis & Levy, 2010). Metode DnD ini memungkinkan penulis untuk secara sistematis mengeksplorasi, mendesain, dan mengimplementasikan solusi yang inovatif sambil memastikan bahwa sistem yang dikembangkan responsif terhadap kebutuhan pengguna. Adapun proses mengenai DnD dalam implementasi penelitian ini di ilustrasikan dalam gambar 3.1.



Gambar 3. 1 Alur Penelitian Metode *Design and Development*.

3.1.1 Analisis Kebutuhan Sistem

Tahap Analisis Kebutuhan Sistem merupakan langkah awal yang krusial dalam proses pengembangan sistem, di mana identifikasi kebutuhan fungsional dan non-fungsional dari sistem yang akan dibangun dilakukan. Pada tahap ini, dilakukan pengumpulan informasi dari berbagai pemangku kepentingan untuk memastikan bahwa sistem yang dikembangkan dapat memenuhi tujuan yang telah ditetapkan. Salah satu kebutuhan utama yang diidentifikasi adalah pengembangan antarmuka pengguna grafis (GUI) yang akan dibuat menggunakan bahasa pemrograman Python. GUI ini akan dirancang untuk memudahkan interaksi pengguna dengan sistem, sehingga memberikan pengalaman pengguna yang intuitif dan efisien. Dengan memahami kebutuhan ini sejak awal, proses perancangan dan pengembangan selanjutnya dapat lebih terarah dan sesuai dengan harapan pengguna.

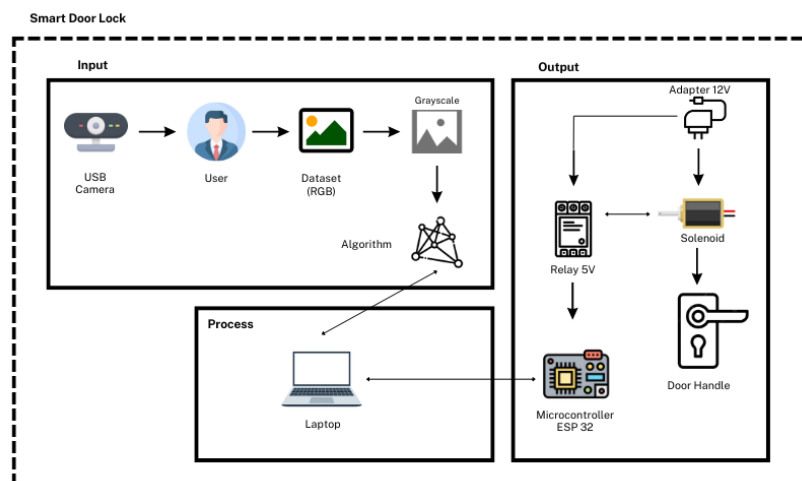
3.1.2 Perancangan Sistem

Tahap perancangan sistem merupakan langkah lanjutan setelah analisis kebutuhan, di mana rancangan arsitektur sistem mulai dibangun. Pada tahap ini, arsitektur sistem dirancang secara menyeluruh untuk memastikan bahwa setiap komponen sistem dapat berfungsi dengan baik dan saling terintegrasi. Pembuatan arsitektur ini mencakup pengembangan desain aplikasi dalam sistem. Selain itu, diagram-diagram penting seperti use case dan aktivitas juga dibuat untuk memodelkan interaksi pengguna dengan sistem serta alur kerja yang terjadi di dalamnya.

Selanjutnya, perancangan sistem juga mencakup pembuatan wiring diagram dan diagram blok yang berfungsi untuk memvisualisasikan hubungan antar komponen perangkat keras dan bagaimana sistem secara keseluruhan bekerja. Wiring diagram akan menunjukkan bagaimana setiap komponen perangkat keras dihubungkan satu sama lain, sementara diagram blok memberikan gambaran yang lebih tinggi tentang interaksi antara modul perangkat keras dan perangkat lunak. Dengan adanya perancangan ini, diharapkan pengembangan dan implementasi sistem dapat berjalan lebih lancar, karena setiap aspek teknis telah dipertimbangkan dan diatur dengan baik dalam tahap perancangan ini.

3.1.2.1 Desain Arsitektur

Desain arsitektur sistem yang ditampilkan dalam gambar ini memberikan gambaran menyeluruh tentang mekanisme kerja kunci pintu, yang dibagi menjadi tiga komponen utama: *input*, *process*, dan *output*. Diilustrasikan pada gambar 3.2.

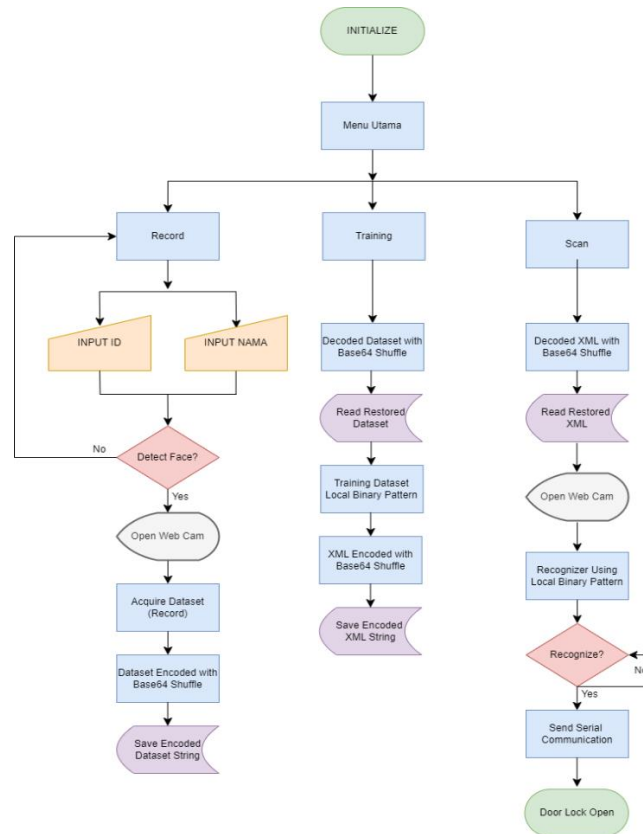


Gambar 3. 2 Diagram Arsitektur

1. **Input:** Tahap ini melibatkan interaksi pengguna, di mana dataset RGB yang mencakup data wajah pengguna diambil melalui input visual. Data ini kemudian dikonversi menjadi format *greyscale* untuk pemrosesan lebih lanjut. Algoritma pengenalan wajah digunakan untuk menganalisis dan memverifikasi identitas pengguna berdasarkan dataset yang diberikan.
2. **Process:** Proses pengolahan data dilakukan oleh sebuah laptop yang terhubung dengan USB camera. Kamera ini bertugas menangkap gambar wajah pengguna secara real-time dan mengirimkannya ke laptop untuk diekstraksi dan dianalisis menggunakan algoritma yang telah ditentukan.
3. **Output:** Hasil dari proses pengolahan data kemudian diteruskan ke mikrokontroler ESP 32 yang terhubung dengan beberapa komponen output. Mikrokontroler ini mengaktifkan relay 5V yang, dengan bantuan adaptor 12V, mengendalikan solenoid yang berfungsi sebagai mekanisme penguncian pintu. Ketika identitas pengguna telah diverifikasi, solenoid akan diaktifkan untuk membuka kunci, memungkinkan akses ke pegangan pintu dan membuka pintu.

3.1.2.2 Design Aplikasi Kunci Pintu

Diagram alir yang ditampilkan pada Gambar 3.3 kunci pintu terdiri dari beberapa komponen penting yang saling berhubungan dan membentuk sistem terintegrasi untuk meningkatkan keamanan. Komponen-komponen ini termasuk modul pengenalan wajah yang menggunakan algoritma *Local binary pattern* (LBP), yang bertugas mengenali dan memverifikasi wajah pengguna. Selain itu, sistem ini juga dilengkapi dengan mekanisme *encoding Base64 Shuffle* untuk mengamankan data wajah yang disimpan dan diolah, memastikan bahwa informasi sensitif terlindungi dari akses yang tidak sah. Proses ini melibatkan beberapa tahapan mulai dari perekaman dataset wajah, pelatihan model, hingga pengenalan dan verifikasi wajah saat pengguna mencoba membuka pintu. Integrasi teknologi ini memungkinkan sistem kunci pintar tidak hanya memberikan keamanan yang lebih baik tetapi juga memastikan efisiensi dan kenyamanan dalam penggunaannya. Diagram blok alir kunci pintu dapat dilihat pada gambar 3.3



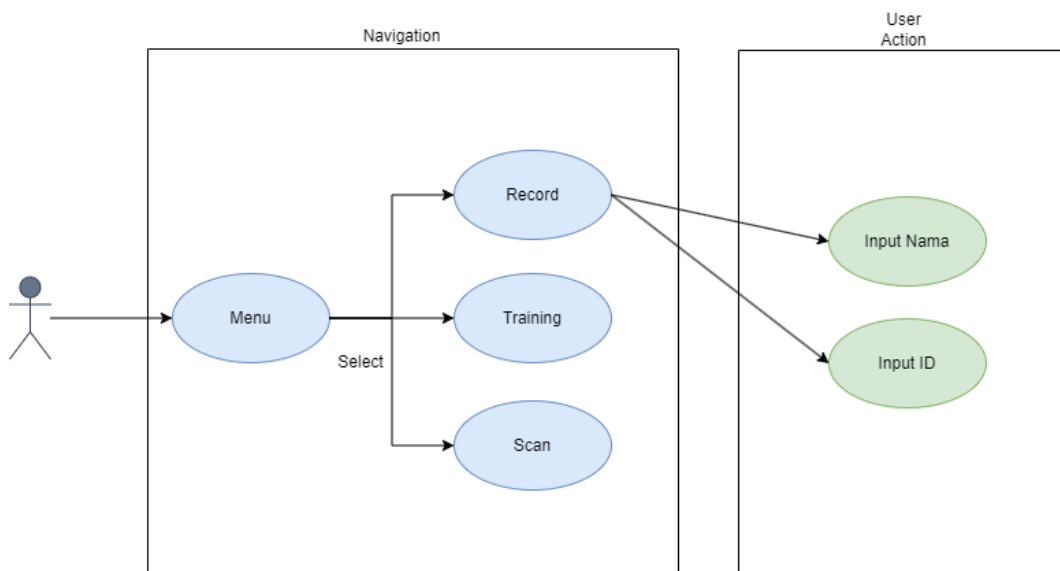
Gambar 3. 3 Diagram Alir Aplikasi

Diagram alir berikut adalah diagram alir dari aplikasi kunci pintu berbasis GUI PyQt5 yang dirancang untuk memanfaatkan teknologi pengenalan wajah dan encode data. penulis menunjukkan proses keseluruhan dari sistem kunci pintu digital berbasis pengenalan wajah dan enkripsi Base64 *Shuffle*. Proses dimulai dengan inialisasi sistem yang kemudian mengarahkan pengguna ke Menu Utama, di mana tiga opsi utama tersedia: Record, Training, dan Scan. Dalam proses "Record", pengguna dapat memasukkan ID dan Nama, yang kemudian diikuti oleh deteksi wajah. Jika wajah terdeteksi, sistem akan membuka webcam untuk merekam dataset wajah, yang kemudian diencode menggunakan Base64 *Shuffle* dan disimpan sebagai string yang diencode. Pada proses "Training", dataset yang telah diencode sebelumnya di-decode menggunakan Base64 *Shuffle* dan dipulihkan. Dataset yang dipulihkan ini kemudian dilatih menggunakan algoritma *Local binary pattern* (LBP) untuk pengenalan wajah, sebelum akhirnya diencode kembali ke dalam format XML yang juga diencode dengan Base64 *Shuffle* dan disimpan. Proses "Scan" bekerja dengan mendecode XML yang telah diencode

menggunakan Base64 *Shuffle*, memulihkan dataset, dan kemudian membuka webcam untuk melakukan pengenalan wajah menggunakan LBP. Jika wajah dikenali, sistem akan mengirimkan komunikasi serial untuk membuka kunci pintu. Diagram ini menggambarkan integrasi antara enkripsi data dan pengenalan wajah dalam sistem, yang memastikan keamanan dan efisiensi dalam pengelolaan akses pintu.

3.1.2.3 Diagram Use Case

Gambar 3.4 menunjukkan gambar diagram use case yang menggambarkan bagaimana interaksi antara pengguna dan sistem aplikasi GUI yang dikembangkan. Penulis membuat diagram use case untuk menunjukkan berbagai fungsi dalam aplikasi yang dapat dilakukan oleh pengguna, seperti fungsi *record*, *training*, dan *scan*.



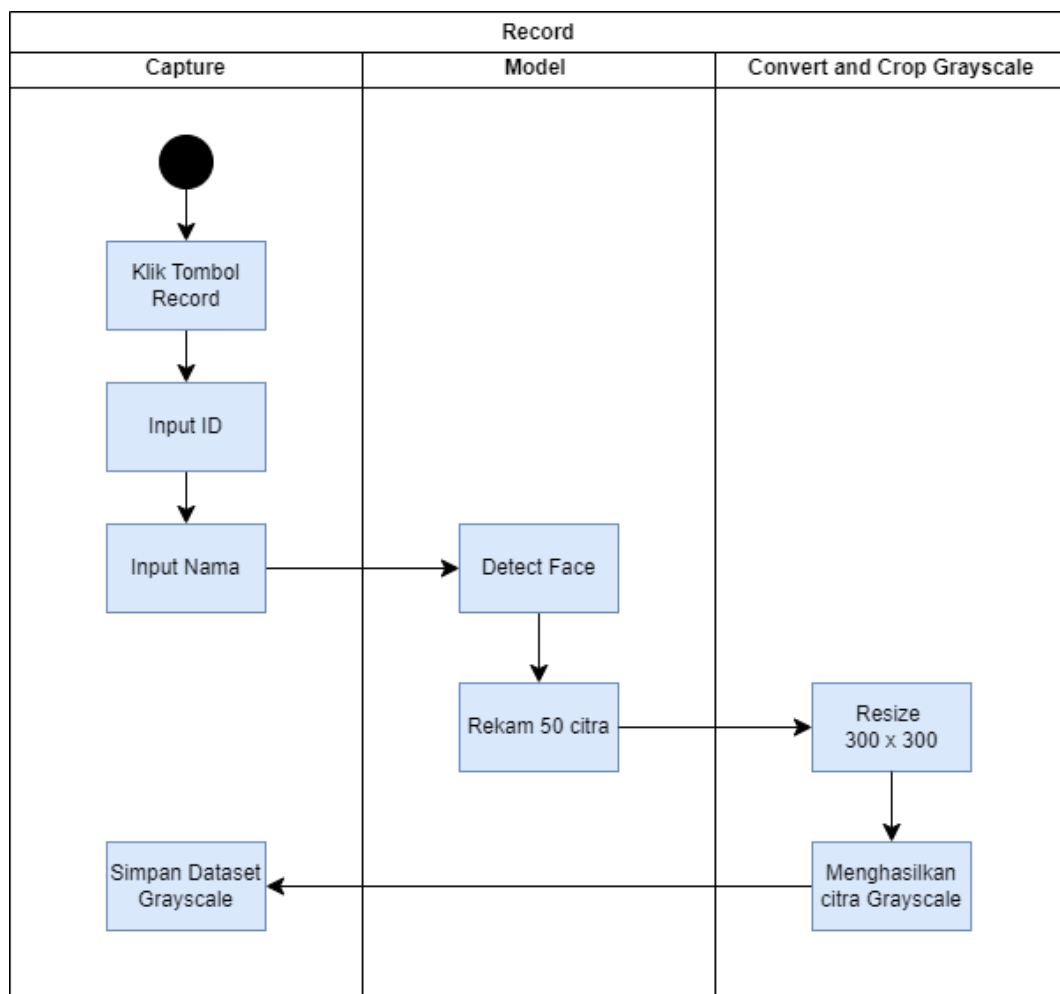
Gambar 3. 4 Diagram Use Case

Diagram use case pada Gambar 3.4, yang ditampilkan menggambarkan interaksi pengguna dengan sistem kunci pintu digital melalui aplikasi GUI. Pengguna memulai dari menu utama, di mana ia memiliki tiga opsi utama: Record, Training, dan Scan. Dalam proses 'Record', pengguna diminta untuk memasukkan ID dan nama sebagai langkah awal dalam pengumpulan data wajah. Fungsi 'Training' digunakan untuk melatih model pengenalan wajah berdasarkan data yang telah dikumpulkan, sedangkan fungsi 'Scan' digunakan untuk memverifikasi

identitas pengguna berdasarkan wajah yang terdeteksi. Diagram ini secara efisien mengilustrasikan alur navigasi dan tindakan pengguna yang diperlukan untuk mengoperasikan sistem kunci pintar ini, mulai dari pengumpulan data hingga proses verifikasi identitas.

3.1.2.4 Diagram Aktivitas

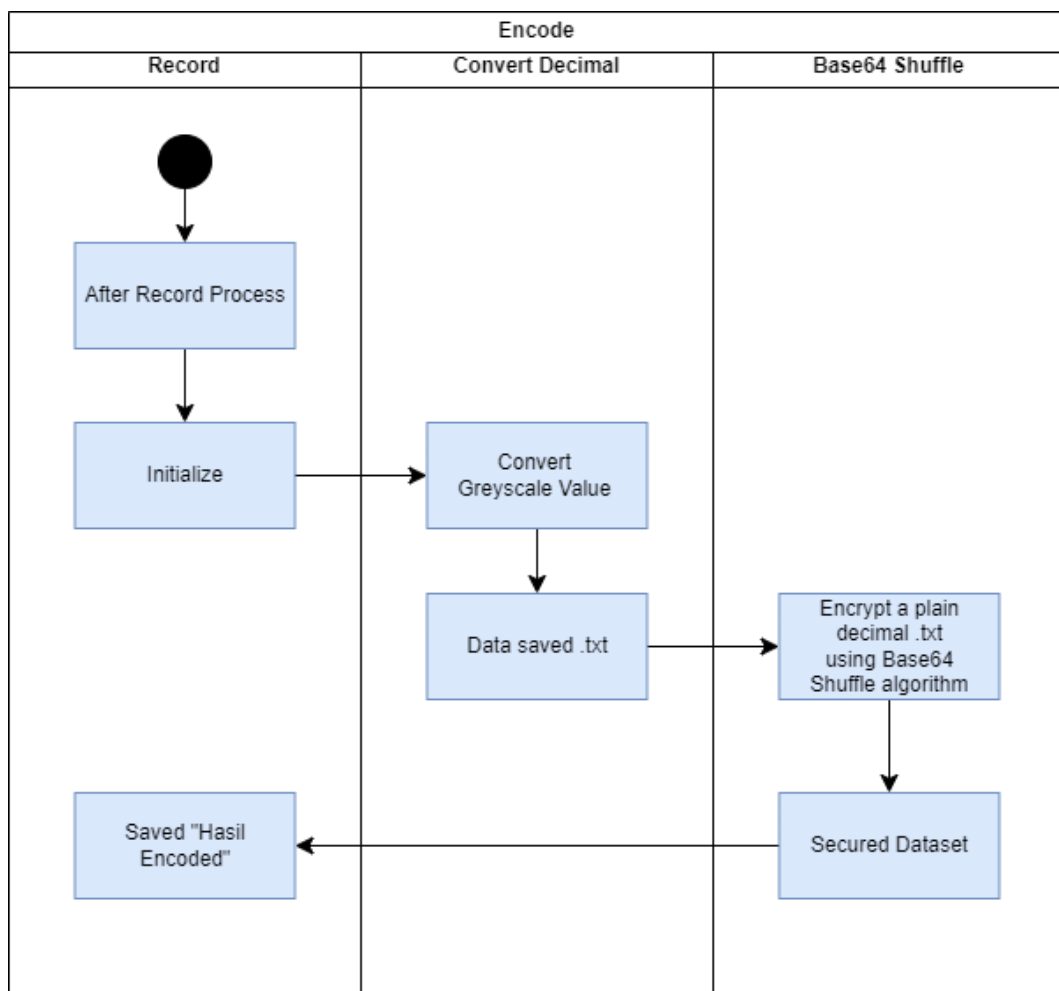
Diagram aktivitas yang disajikan memberikan gambaran yang jelas dan terstruktur mengenai alur operasional dan interaksi dalam aplikasi. Diagram ini dirancang untuk memvisualisasikan langkah demi langkah dalam proses kerja aplikasi, mulai dari interaksi awal pengguna hingga akhir.



Gambar 3. 5 Diagram Aktivitas Record

Gambar 3.5 menunjukkan diagram aktivitas dari proses perekaman data wajah dalam sistem kunci pintu digital. Proses ini dimulai dengan pengguna menekan

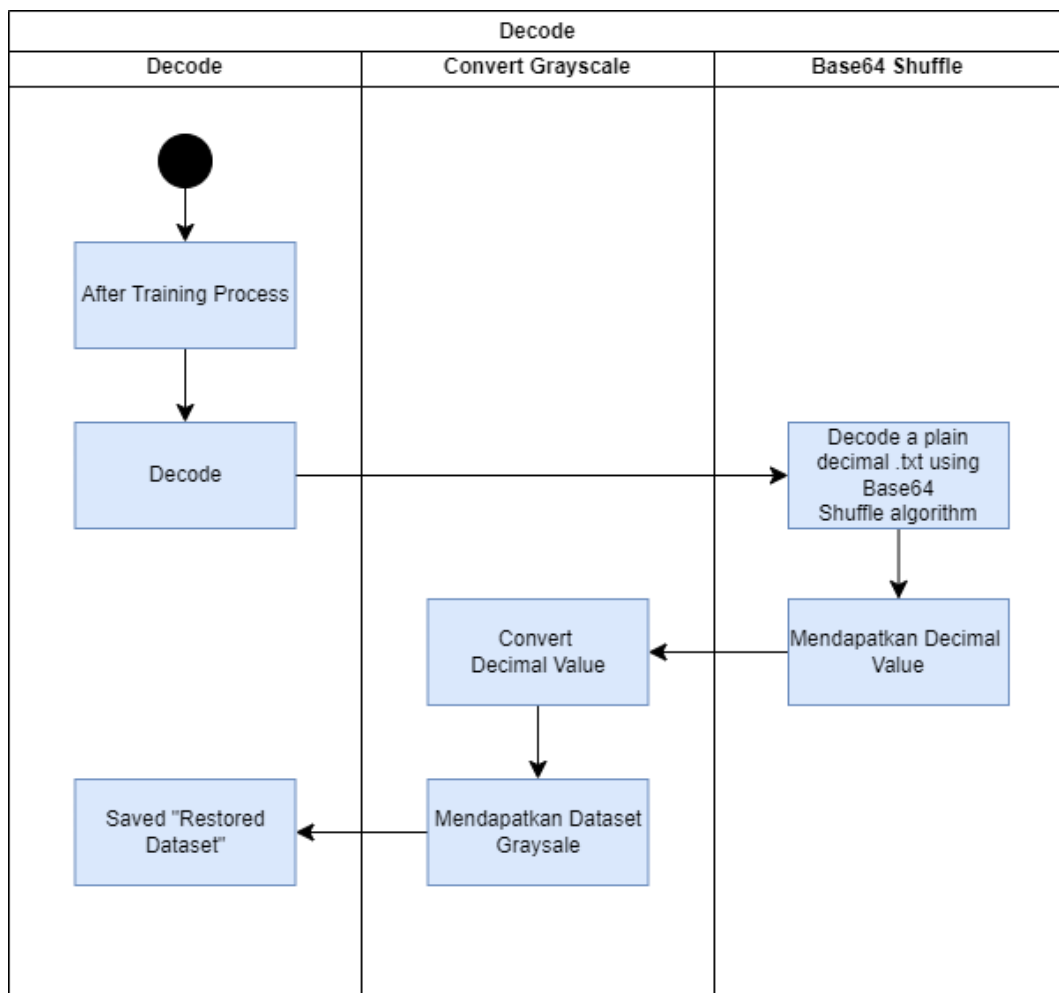
tombol "Record," yang dilanjutkan dengan memasukkan ID dan nama mereka sebagai identitas awal. Setelah itu, sistem akan mendeteksi wajah pengguna menggunakan kamera, dan jika wajah terdeteksi, sistem akan merekam 50 citra wajah. Citra-citra ini kemudian diubah menjadi format grayscale dan diresize menjadi ukuran 300x300 piksel untuk memastikan konsistensi dalam dataset. Langkah akhir dalam proses ini adalah menyimpan dataset yang telah diolah dalam format grayscale, yang kemudian akan digunakan untuk pelatihan dan pengenalan wajah dalam sistem keamanan ini. Diagram ini secara jelas mengilustrasikan alur proses yang diperlukan untuk mengumpulkan dan mempersiapkan data wajah yang berkualitas untuk digunakan dalam sistem pengenalan wajah.



Gambar 3. 6 Diagram Aktivitas Encode

Gambar 3.6 menunjukkan diagram aktivitas dari proses *encoding* data setelah perekaman citra wajah dalam sistem kunci pintu digital. Proses dimulai

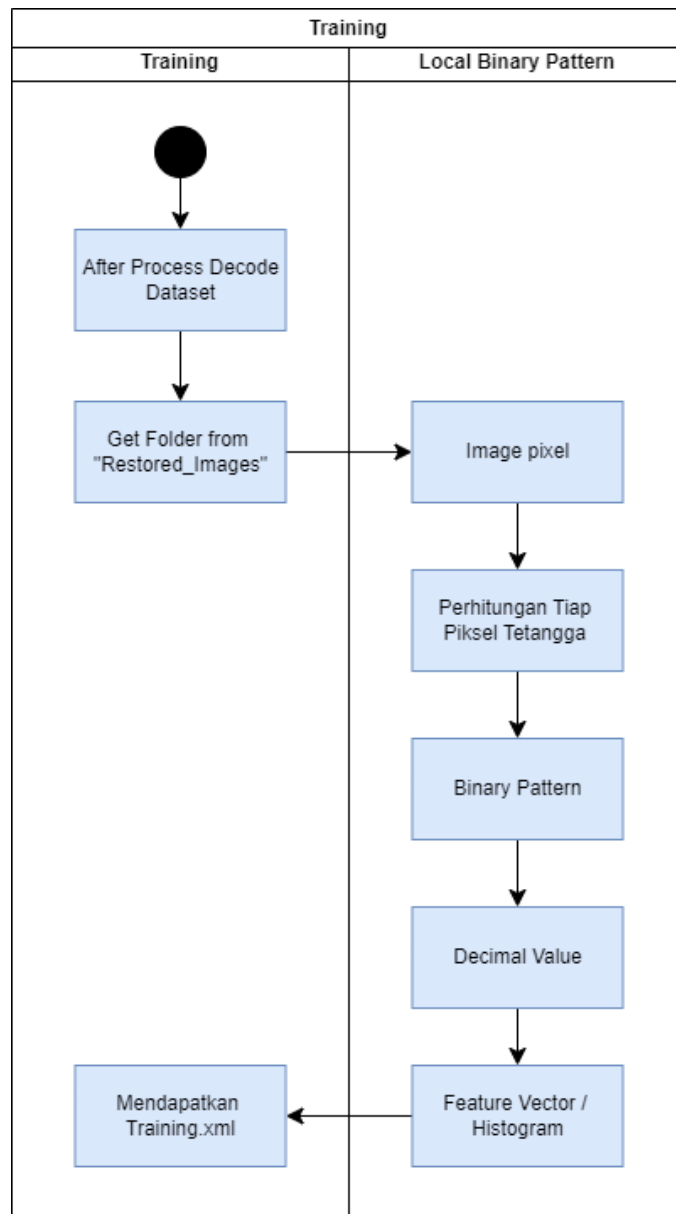
setelah tahap perekaman dengan inialisasi sistem untuk mempersiapkan data yang akan diencode. Nilai grayscale dari citra yang telah direkam kemudian diubah menjadi format desimal dan disimpan dalam *file* .txt. Selanjutnya, *file* desimal ini dienkripsi menggunakan algoritma Base64 *Shuffle*, menghasilkan dataset yang aman. Langkah terakhir dalam proses ini adalah menyimpan hasil yang telah diencode sebagai "Hasil Encoded." Diagram ini mengilustrasikan secara rinci bagaimana data citra yang telah direkam diproses dan diamankan menggunakan teknik enkripsi, memastikan bahwa data tersebut terlindungi dari akses yang tidak sah.



Gambar 3. 7 Diagram Aktivitas Decode

Gambar 3.7 menunjukkan diagram aktivitas proses decode, menggambarkan langkah-langkah yang dilakukan setelah proses pelatihan dalam sistem kunci pintu digital. Proses dimulai dengan tahap decoding, di mana data yang

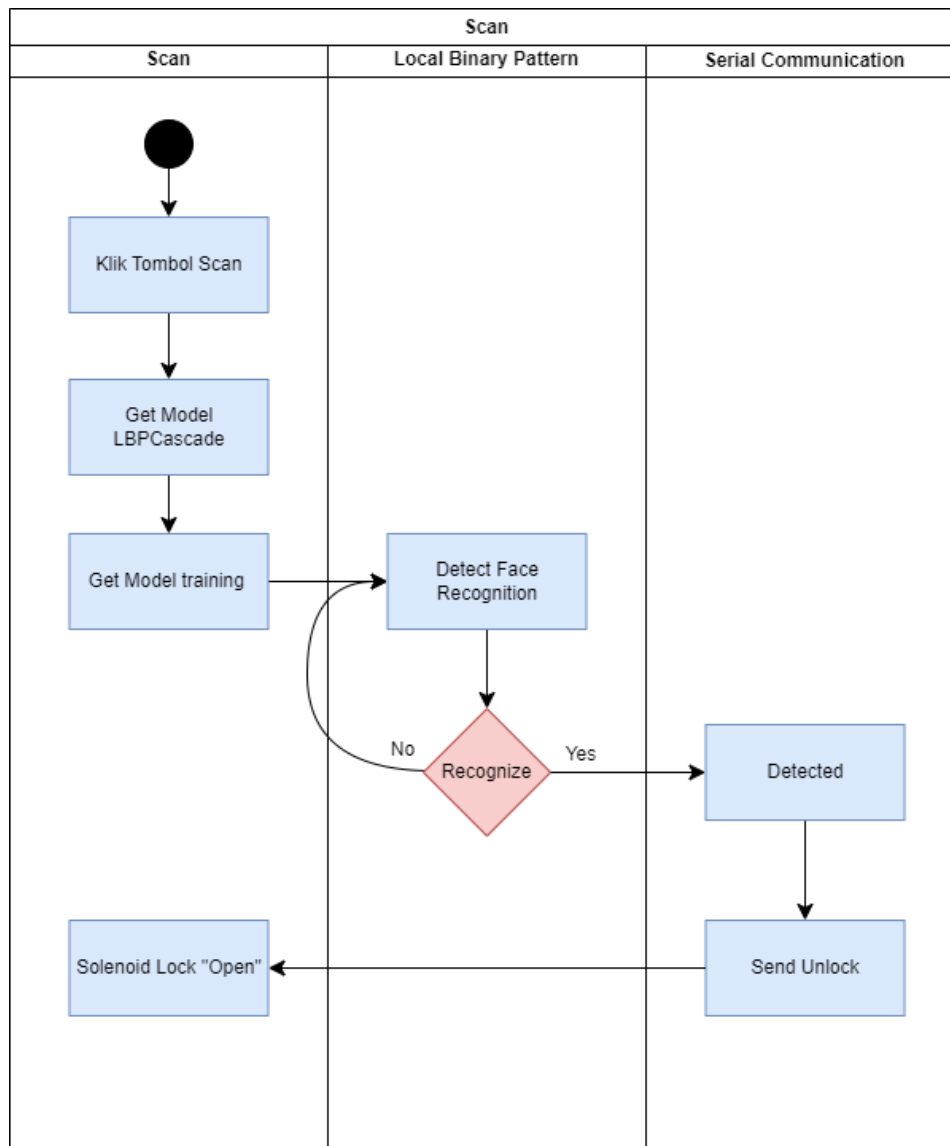
telah diencode sebelumnya di-decode menggunakan algoritma *Base64 Shuffle* untuk mendapatkan nilai desimal asli. Setelah nilai desimal diperoleh, langkah selanjutnya adalah mengonversi nilai desimal tersebut kembali ke dalam format grayscale untuk memulihkan dataset citra asli. Dataset grayscale yang dipulihkan ini kemudian disimpan sebagai "Restored Dataset," yang siap digunakan untuk keperluan lebih lanjut dalam pengenalan wajah.



Gambar 3. 8 Diagram Aktivitas Training

Gambar 3.8 menunjukkan diagram aktivitas proses pelatihan menggunakan algoritma *Local binary pattern* (LBP). Setelah proses decode dataset selesai, sistem

mengambil folder berisi citra yang dipulihkan dari "Restored_Images." Kemudian, setiap piksel dalam citra dianalisis, di mana nilai-nilai tetangga dihitung untuk membentuk pola biner. Pola biner ini dikonversi menjadi nilai desimal, yang kemudian digunakan untuk menghasilkan vektor fitur atau histogram. Hasil akhir dari proses ini adalah *file* "Training.xml" yang berisi model pelatihan yang akan digunakan untuk pengenalan wajah.

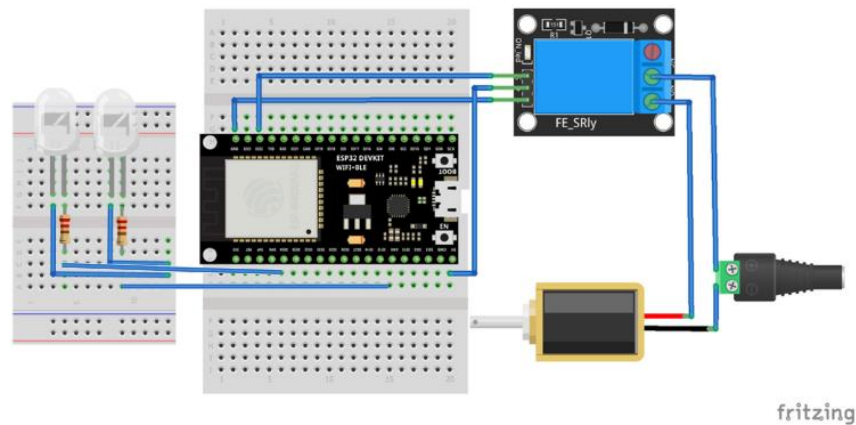


Gambar 3. 9 Diagram Aktivitas Scan

Gambar 3.9 diagram aktivitas scan menggambarkan proses pengenalan wajah yang dilakukan oleh sistem kunci pintu digital. Proses dimulai ketika pengguna menekan tombol "Scan," yang kemudian untuk mendeteksi dan mengenali wajah.

3.1.2.5 Wiring Diagram

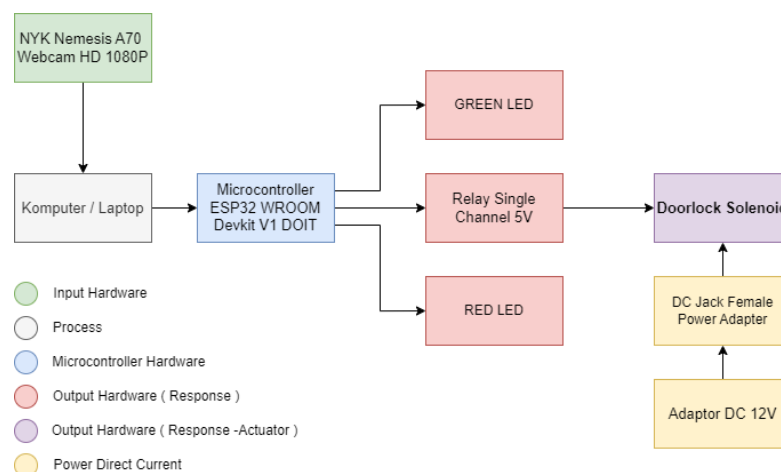
Sebelum melakukan pengembangan pada perangkat keras, diagram ini berfungsi untuk memudahkan wiring component perangkat keras yang disesuaikan dengan fungsi pinout GPIO pada mikrokontroler ESP32 karena telah memiliki gambaran ilustrasi sebelumnya. Adapun detail dari wiring diagram tersebut dapat dilihat pada Gambar 3.10.



Gambar 3. 10 Wiring Diagram Pengembangan Sistem

3.1.2.6 Diagram blok

Dalam diagram blok pada pengembangan sistem perangkat keras ini, terdapat beberapa komponen penting seperti sebagai input, mikrokontroler sebagai proses, dan hingga modul aktuator sebagai output. Adapun detail dari komponen tersebut dapat dilihat pada Gambar 3.11.



Gambar 3. 11 Diagram blok

Diagram blok tersebut merupakan instrumen alat visual yang penting untuk penulis sebagai pengembang dalam memberikan representasi visual berupa segi

empat dan bagaimana mereka berinteraksi satu sama lain. Diagram blok menggambarkan sistem yang terdiri dari beberapa komponen elektronik yang terhubung dan berinteraksi untuk melakukan tugas tertentu. Sistem ini menggunakan komputer atau laptop sebagai unit pemrosesan utama, yang terhubung dengan webcam HD 1080p merek NYK Nemesis A70. Webcam ini berfungsi sebagai input perangkat keras yang mengirimkan data visual ke mikrokontroler ESP32 WROOM Devkit V1 DOIT. Seluruh sistem ditenagai melalui adaptor DC 12V yang terhubung dengan DC jack female power adapter. Diagram ini juga menandai arus listrik langsung (DC) yang mengalir dalam sistem, menunjukkan bahwa power supply DC 12V digunakan untuk mengaktifkan output hardware seperti solenoid pintu.

3.1.3 Pengembangan Sistem

Tahap Pengembangan Sistem adalah tahap di mana implementasi dari perancangan yang telah dibuat sebelumnya direalisasikan. Pada tahap ini, setiap komponen sistem yang telah dirancang mulai dari arsitektur hingga diagram diagram pendukung seperti use case, aktivitas, wiring, dan diagram blok diwujudkan dalam bentuk kode program dan pengaturan perangkat keras. Pengembangan antarmuka pengguna grafis (GUI) menggunakan Python juga dilakukan pada tahap ini, dengan mengikuti desain aplikasi yang telah dirancang untuk memastikan pengalaman pengguna yang optimal. Selain itu, setiap modul dan fungsi yang diidentifikasi dalam perancangan diimplementasikan secara detail untuk memastikan bahwa mereka beroperasi sesuai dengan spesifikasi. Selama pengembangan, pengujian awal dilakukan untuk mengidentifikasi dan memperbaiki masalah yang muncul, sehingga sistem yang dikembangkan sesuai dengan harapan dan siap untuk diuji secara lebih menyeluruh pada tahap berikutnya.

3.1.4 Pengujian Sistem

Rencana pengujian sistem akan mencakup berbagai aspek untuk memastikan keandalan dan kinerja dari sistem yang dikembangkan. Pengujian perangkat keras akan difokuskan pada verifikasi kinerja komponen fisik untuk memastikan bahwa semua perangkat berfungsi sesuai dengan spesifikasi. Pengujian perangkat lunak akan menguji integrasi dan stabilitas aplikasi, serta memastikan

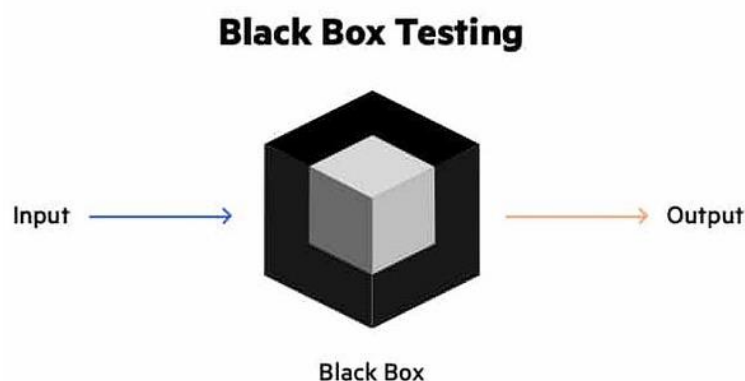
kompatibilitas dengan lingkungan yang berbeda. Pengujian model *Local binary pattern* akan mengevaluasi akurasi model deteksi wajah berdasarkan dataset yang digunakan. Terakhir, pengujian sistem *black box* akan dilakukan untuk memeriksa fungsionalitas keseluruhan dari sistem tanpa melihat ke dalam kode sumber.

3.1.4.1 Pengujian Perangkat Lunak

Pengujian perangkat lunak, rencana pengujian akan mencakup beberapa aspek kritis untuk memastikan kualitas dan kinerja sistem. Pertama, hasil dari pengujian GUI aplikasi akan diperiksa untuk memastikan bahwa antarmuka pengguna berfungsi dengan baik dan responsif. Pengujian *black box* akan dilakukan untuk mengevaluasi fungsi perangkat lunak tanpa memeriksa kode internalnya, memastikan semua fitur bekerja sesuai harapan. Pengumpulan dataset juga akan menjadi bagian penting, dimana dataset yang digunakan harus memadai dan relevan untuk tujuan pengujian. Selain itu, hasil proses algoritma *Local binary pattern* akan dianalisis untuk mengevaluasi akurasi dan efisiensi dalam mengenali pola wajah. Selanjutnya, pengujian pengenalan wajah menggunakan *Local binary pattern* dan pengujian encode menggunakan Base64 akan dilakukan untuk mengukur kinerja dan keandalan sistem dalam berbagai skenario penggunaan.

3.1.4.2 Pengujian Sistem *Black box*

Pengujian *black box* dirancang untuk mengidentifikasi berbagai tipe kesalahan, termasuk fungsi yang salah atau tidak berfungsi, kesalahan antarmuka, masalah dalam struktur data, kesalahan perilaku atau kinerja aplikasi, serta kesalahan pada proses inisialisasi dan terminasi aplikasi (Sulyz Andrey, 2023). Untuk melihat lebih detail mengenai bagaimana metode pengujian *black box* diimplementasikan, dapat dilihat pada Gambar 3.12.



Gambar 3. 12 metode pengujian *black box*

Pengujian ini dilakukan menggunakan perangkat Persocal Computer Intel I3-12100F, Ram 16GB. Dalam skripsi ini, penulis menggunakan metode pengujian *Black box* untuk mengevaluasi aplikasi Kunci pintu yang dikembangkan. Pengujian *Black box*, atau pengujian fungsional, adalah pendekatan di mana penulis menguji aplikasi berdasarkan spesifikasi fungsionalnya tanpa mempertimbangkan struktur internal, detail implementasi, atau kode sumber. Tujuan dari pengujian ini adalah untuk memverifikasi bahwa semua fungsi aplikasi beroperasi sesuai dengan persyaratan yang ditentukan dan menangani input dari pengguna dengan cara yang diharapkan.

3.1.5 Evaluasi Sistem

Setelah menyelesaikan fase pengembangan, langkah berikutnya yang diambil oleh penulis adalah melakukan pengujian dan evaluasi terhadap aplikasi yang telah dibangun. Tahap ini krusial untuk memverifikasi bahwa aplikasi memenuhi spesifikasi dan kebutuhan yang telah ditetapkan sebelumnya. Selama proses ini, penulis menggunakan metode pengujian *black box*, atau yang juga dikenal sebagai behavioral testing, yang berfokus pada persyaratan fungsional dari fitur perangkat lunak. Metode pengujian *black box* ini dipilih karena memungkinkan pengujian sistem berdasarkan input dan output tanpa perlu mengetahui detail implementasi internal atau kode program. Penulis menguji setiap fitur dengan memberikan berbagai jenis input untuk melihat bagaimana sistem merespons dan memastikan bahwa hasil yang diperoleh sesuai dengan yang diharapkan.

3.2 Spesifikasi Pengembangan Sistem

Perangkat penunjang penelitian terdiri dari dua bagian yaitu, perangkat keras dan perangkat lunak. Adapun kebutuhan teknologi yang sesuai dengan fungsionalitasnya dalam menunjang pengembangan sistem berdasarkan penelitian sebelumnya yang memaparkan fungsionalitas disetiap alat penunjang (Mohanty dkk., 2021). Teknologi yang tersajikan pada Tabel 3.1 dan Tabel 3.2.

Tabel 3. 1
Perangkat Lunak Yang Digunakan

Software	Versi	Fungsi
Visual Studio Code	1.91.1	Sebagai editor kode dalam lingkup pengembangan aplikasi. Digunakan untuk mendebug kode program
PlatformIO	6.1.8	Berfungsi sebagai platform pengembangan terintegrasi (IDE) untuk mikrokontroler.
Windows 11	23H2 x64-based Systems	Sebagai sistem operasi perangkat lunak pengembangan.

Tabel 3. 2
Perangkat Keras Yang Digunakan

Hardware	Spesifikasi	Fungsi
Personal Computer	Prosesor Intel I3-12100F, memori 16 GB RAM DDR4	Komputer digunakan sebagai pusat pengolahan data dan pengembangan perangkat lunak.
Web Cam	Web Cam HD Resolusi 1080P Cam	Digunakan sebagai perangkat input utama untuk mendeteksi dan menangkap gambar wajah.
Mikrokontroler	ESP32-WROOM-32	Sebagai otak pada sistem tertanam perangkat keras dalam menangani proses output.

Hardware	Spesifikasi	Fungsi
<i>Doorlock</i>	<i>Solenoid 12V</i>	Sebagai aktuator komponen kunci pintu yang mengubah energi listrik menjadi gerakan mekanis.
<i>Relay</i>	<i>Relay Module 5V</i>	Sebagai sakelar dalam membuka atau menutup kunci pintu
<i>Adapter</i>	<i>Adapter 12V</i>	Sebagai sumber daya dalam mengaktifkan komponen perangkat keras.
<i>Jack</i>	<i>Jack DC Female</i>	Sebagai penghubung antar komponen perangkat keras dengan daya listrik.
Kabel	Kabel Jumper	Sebagai penghubung antar komponen perangkat keras sementara.

Adapun spesifikasi diatas yang hampir menyerupai spesifikasi pengujian dengan pc mini lenovo thinkcentre m710Q Tiny i5 yang dapat diimplementasikan seperti pengujian. (Lenovo, t.t.)



Gambar 3. 13 PC Mini Lenovo Thinkcentre M710Q Tiny

(Lenovo, t.t.)