

LAPORAN PROJECT
WORKSHOP EMBEDDED SYSTEM
SMARTLOCK BERBASIS FACE RECOGNITION DAN APLIKASI
ANDROID DENGAN USER AUTORIZATION



DOSEN PENGAJAR :

Moch. Zen Samsono Hadi, S.T., M.Sc., PhD.

OLEH :

Raffi Mahya Pratama	2423600002
Abyaz Prince Muhammad	2423600007
Muhammad Nizamudin M	2423600024
Wajihah Zahrah Ramadhani	2423600030

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA TAHUN 2024

1. LATAR BELAKANG

Di era digital, kebutuhan akan keamanan yang lebih pintar dan efisien semakin meningkat. Sistem keamanan tradisional, seperti kunci fisik atau PIN, seringkali memiliki kelemahan seperti risiko hilang atau mudah diretas. Untuk mengatasi ini, teknologi biometrik seperti pengenalan wajah mulai banyak digunakan karena lebih aman dan praktis. Dengan memanfaatkan pengenalan wajah, pengguna bisa mengakses sistem hanya dengan menghadapkan wajah ke perangkat, tanpa perlu mengingat kode atau membawa kunci fisik. Proyek ini berfokus pada pengembangan smartlock berbasis pengenalan wajah yang terhubung dengan aplikasi Android. Aplikasi tersebut mempermudah pengguna dalam mengatur akses, seperti menambah atau menghapus pengguna, dan memantau log akses secara real-time. Sistem ini dirancang untuk memberikan keamanan tinggi lewat otentikasi biometrik, sambil memastikan kemudahan penggunaan melalui integrasi dengan perangkat pintar berbasis IoT (Internet of Things).

2. DASAR TEORI

Keamanan digital merupakan aspek penting dalam melindungi akses terhadap sistem, data, atau perangkat elektronik dari ancaman pihak yang tidak berwenang. Salah satu inovasi dalam bidang keamanan adalah penguncian pintar (smartlock), yaitu perangkat pengunci elektronik yang dapat dioperasikan secara digital tanpa menggunakan kunci fisik. Smartlock memanfaatkan teknologi modern seperti Internet of Things (IoT) untuk memungkinkan kontrol jarak jauh dan integrasi dengan aplikasi berbasis Android, sehingga meningkatkan kemudahan dan keamanan pengguna.

Teknologi biometrik seperti pengenalan wajah (face recognition) menjadi salah satu fitur utama dalam sistem ini. Pengenalan wajah bekerja dengan mendeteksi area wajah, mengekstraksi fitur unik seperti jarak antar mata dan bentuk hidung, serta mencocokkannya dengan data yang tersimpan untuk verifikasi identitas. Dibandingkan metode autentikasi konvensional seperti penggunaan sandi, teknologi ini lebih sulit untuk dipalsukan, sehingga menawarkan tingkat keamanan yang lebih tinggi.

Aplikasi berbasis Android berfungsi sebagai antarmuka utama untuk mengontrol smartlock. Melalui aplikasi ini, pengguna dapat mendaftarkan data wajah, mengelola hak akses, dan memantau status kunci secara real-time. Proses otorisasi pengguna (user authorization) menjadi elemen penting dalam sistem ini, di mana autentikasi memastikan identitas pengguna sementara otorisasi menentukan hak aksesnya, seperti kemampuan untuk menambah atau menghapus pengguna lain.

Sistem ini juga memanfaatkan IoT untuk menghubungkan perangkat fisik ke internet, memungkinkan notifikasi real-time serta kendali jarak jauh. Meskipun demikian, tantangan seperti ketergantungan pada perangkat keras dan perangkat lunak, risiko serangan siber, serta kebutuhan koneksi internet yang stabil menjadi perhatian dalam pengembangannya. Secara keseluruhan, kombinasi teknologi face recognition, aplikasi Android, dan IoT dalam sistem smartlock ini memberikan solusi modern untuk meningkatkan keamanan dan kenyamanan pengelolaan akses.

3. BATASAN APLIKASI

Adapun Batasan aplikasi dengan tujuan agar perancangan dapat lebih focus dan menghindari meluasnya pembahasan

1. Perancangan aplikasi android menggunakan software android studio dan Bahasa pemrograman Kotlin
2. Aplikasi hanya bisa dijalankan pada android versi 10+
3. Penyimpanan data secara real-time menggunakan FireBase
4. Metode face recognition menggunakan siemens neural network (SNN)
5. Pada perancangan ini masih hanya melatih untuk mengenali satu wajah
6. Pengambilan gambar menggunakan esp32cam
7. Pemrosesan logic dari alat ini menggunakan esp8266
8. Proses pengenalan wajah menggunakan laptop

4. IMPLEMENTASI/SOLUSI

Sistem ini dirancang untuk menghadirkan solusi keamanan modern yang cocok untuk rumah, kantor, atau tempat lainnya. Beberapa tantangan utama yang diatasi adalah akurasi pengenalan wajah dalam kondisi pencahayaan beragam, kecepatan respons, dan perlindungan data pengguna. Dengan teknologi ini, diharapkan pengguna mendapatkan pengalaman keamanan yang lebih nyaman, canggih, dan mudah dikelola dibandingkan metode tradisional.

Alat ini menggunakan metode DeepLearning siemens neural network (SNN) yang terintegrasi dengan aplikasi mobile. Aplikasi mobile dapat mengontrol smartlock dengan fitur button, juga dapat melihat hasil gambar yang diambil oleh esp32cam dengan fitur penampilan gambar nya, dan aplikasi mobile ini juga memiliki user authentication dengan uid dari alat, username, dan password.

4.1. IMPLEMENTASI ALAT

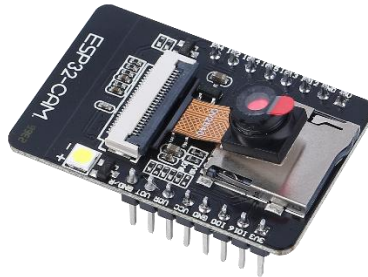
Alat ini menggunakan 2 microcontroller, esp32cam dan esp8266 sebagai pemrosesan logic. Esp32cam berfungsi sebagai penerima input dari sensor infrared, sensor infrared sendiri difungsikan sebagai trigger untuk memulai proses pengenalan wajah. Setelah esp32cam menerima trigger dari sensor infrared maka esp32cam akan memulai mengambil gambar. Setelah mengambil gambar maka esp32cam akan mengirimkan state "1" ke FireBase yang nantinya laptop yang berfungsi sebagai server untuk pengenalan wajah akan menerima state tersebut dan memulai proses pengenalan wajah. Setelah laptop selesai melakukan proses pengenalan wajah, maka laptop akan mengirimkan state ke FireBase dan state tersebut akan ditangkap oleh esp8266 untuk menjalankan output sesuai state yang diberikan oleh laptop.



4.1.1. KOMPONEN KOMPONEN YANG DIGUNAKAN

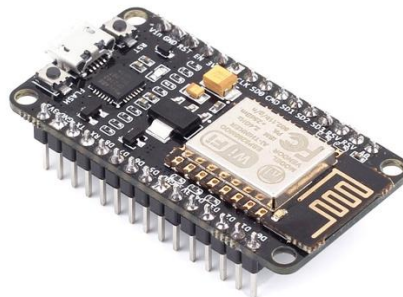
Adapun komponen komponen yang digunakan untuk merealisasikan proyek ini sebagai berikut:

4.1.1.1. Esp32cam



Mikrokontroler dengan fitur kamera yang mendukung pemrosesan gambar dan video. Memiliki konektivitas Wi-Fi dan Bluetooth, cocok untuk proyek IoT seperti pemantauan jarak jauh.

4.1.1.2. Esp8266



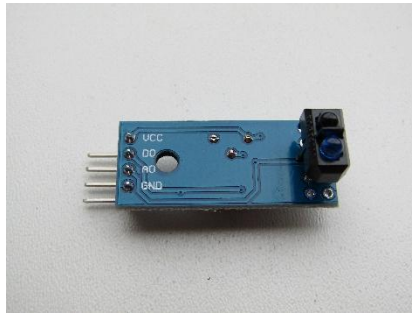
Mikrokontroler yang populer untuk aplikasi IoT karena memiliki modul Wi-Fi bawaan, memungkinkan konektivitas internet dengan harga terjangkau.

4.1.1.3. Relay DC low trigger



Komponen elektronik yang berfungsi sebagai saklar yang diaktifkan dengan sinyal tegangan rendah, biasanya digunakan untuk mengontrol perangkat listrik yang bekerja pada tegangan lebih tinggi.

4.1.1.4. Sensor infrared



Sensor yang mendeteksi keberadaan objek atau gerakan menggunakan sinyal inframerah, sering digunakan dalam aplikasi seperti deteksi jarak atau pengamanan.

4.1.1.5. OLED SSD 1306



Layar kecil berbasis OLED dengan driver SSD1306, sering digunakan untuk menampilkan informasi seperti teks atau grafik dalam berbagai aplikasi elektronik.

4.1.1.6. Solenoid lock 12v



Kunci elektronik yang bekerja dengan mekanisme solenoid. Aktif ketika diberi daya 12V, sering digunakan dalam sistem akses otomatis.

4.1.1.7. Adaptor 220v AC to 12v DC



Adaptor untuk mengubah arus listrik dari tegangan AC (220V) menjadi tegangan DC (12V), biasanya digunakan untuk menyalakan perangkat elektronik kecil.

4.1.1.8. Stepdown 12v DC to 5v DC



Modul yang menurunkan tegangan DC dari 12V menjadi 5V, digunakan untuk perangkat elektronik yang memerlukan tegangan lebih rendah seperti mikrokontroler.

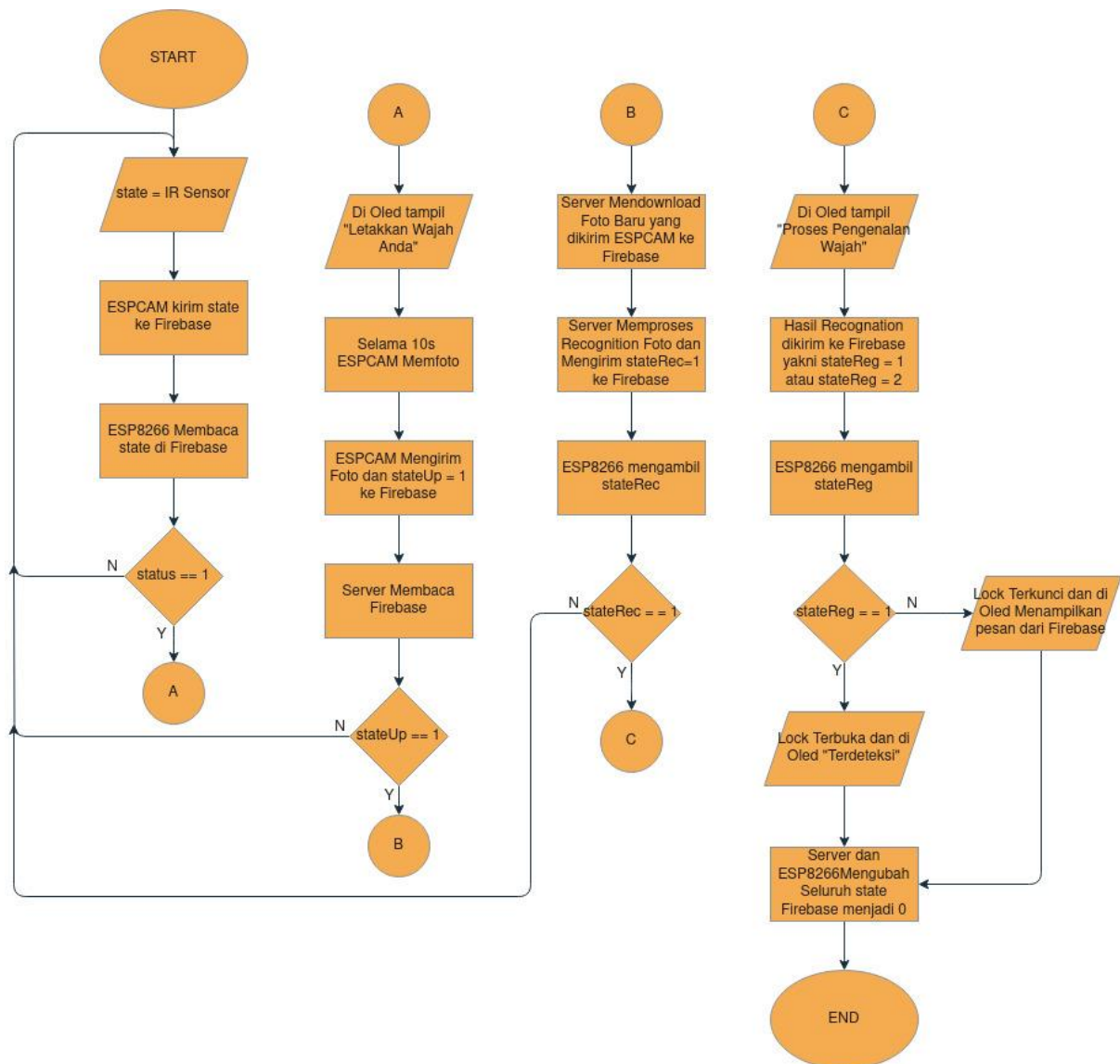
4.1.1.9. Dioda 1n4058



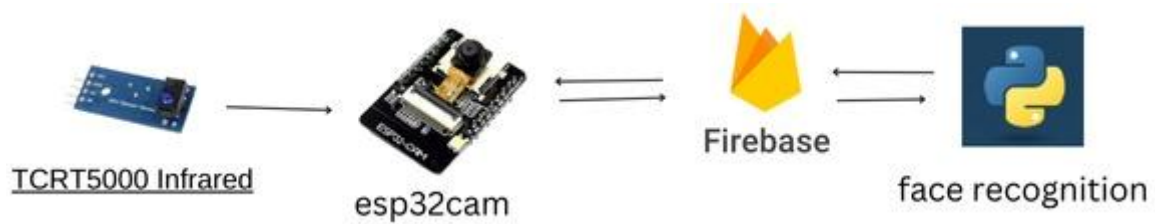
Dioda yang digunakan untuk mengarahkan arus listrik dalam satu arah dan mencegah arus balik, melindungi komponen dalam sirkuit elektronik.

4.1.2. ALUR KERJA ALAT

4.1.2.1 Flowchart

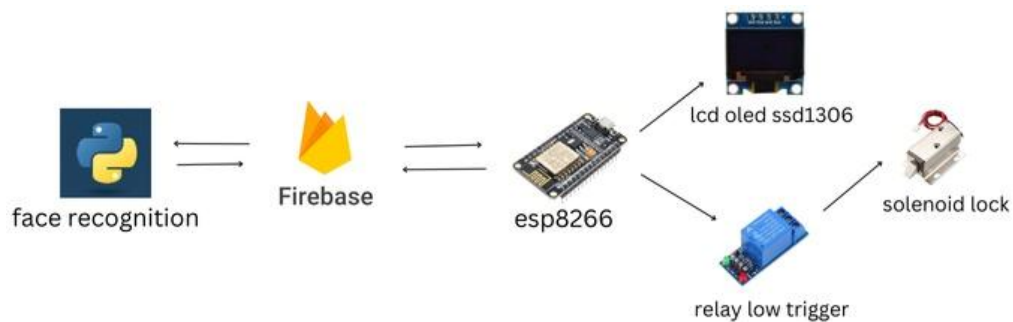


4.1.2.2 Cara kerja pengambilan gambar



1. Sensor inframerah mendeteksi keberadaan seseorang.
2. ESP32-CAM mengambil gambar wajah dan mengunggahnya ke Firebase.
3. Firebase menjadi penghubung antara perangkat keras dan algoritma pengenalan wajah.
4. Python memproses gambar untuk mengenali wajah, lalu mengirimkan hasilnya ke Firebase.
5. Firebase memberikan respons ke sistem, seperti membuka kunci atau menolak akses.

4.1.2.3 Cara kerja pemberian akses



1. Sistem pengenalan wajah (Python) memproses gambar dan memutuskan apakah wajah pengguna dikenali.
2. Hasil autentikasi dikirim ke Firebase.
3. Firebase meneruskan data ke ESP8266.
4. ESP8266 mengontrol perangkat keras berdasarkan hasil autentikasi:
 - Jika “akses diterima”:
 - Layar OLED menampilkan pesan "Selamat datang tuan".
 - Relay mengaktifkan solenoid lock untuk membuka pintu.
 - Jika “akses ditolak”:
 - Layar OLED menampilkan pesan "Anda siapa? ".
 - Relay tidak mengaktifkan solenoid lock.

4.1.3. PROGRAM ALAT

```
#include <eloquent_esp32cam.h>
using eloq::camera;

#include "esp_camera.h"
#include <WiFi.h>
#include <Firebase_ESP_Client.h>

#include "addons/TokenHelper.h"
#include "addons/RTDBHelper.h"

#define WIFI_SSID "eepis-mahya"
#define WIFI_KEY "1sampai9"
#define API_KEY "AIzaSyCF3UkvP6GN5nJswK2_IDt1NCaDxbamaXY"
#define DATABASE_URL "https://espcam-e3bb4-default-rtdb.asia-southeast1.firebaseio.com/"
#define STORAGE_BUCKET_ID "espcam-e3bb4.appspot.com"

#define FLASH_LED_PIN 4

FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;

bool signUpOk = false;
int sensor = 14;
int sensorValue;

void setup() {
    delay(3000);
    Serial.begin(115200);
    Serial.println("__GET YOUR FIRST PICTURE__");

    camera.pinout.aithinker();
    camera.brownout.disable();
    camera.resolution.face();
    camera.quality.high();

    // init camera
    while (!camera.begin().isOk())
        Serial.println(camera.exception.toString());

    Serial.println("Camera OK");
    Serial.println("Enter 'capture' (without quotes) to shot");

    pinMode(sensor, INPUT);

    WiFi.mode(WIFI_STA);
    WiFi.begin(WIFI_SSID, WIFI_KEY);
    while(WiFi.status() != WL_CONNECTED){
        Serial.print(",");
        delay(300);
    }
}
```

```

Serial.println();
Serial.print("connected with ip: ");
Serial.println(WiFi.localIP());

config.api_key = API_KEY;
config.database_url = DATABASE_URL;
if(Firebase.signUp(&config, &auth, "", "")){
    Serial.println("signed up");
    signUpOk = true;
}else{
    Serial.printf("%s\n", config.signer.signupError.message.c_str());
}
config.token_status_callback = tokenStatusCallback;
Firebase.begin(&config, &auth);
Firebase.reconnectWiFi(true);

pinMode(FLASH_LED_PIN, OUTPUT);

// Menyalakan flash LED
digitalWrite(FLASH_LED_PIN, HIGH); // FLASH ON
delay(1000); // LED tetap menyala selama 1 detik

// Mematikan flash LED
digitalWrite(FLASH_LED_PIN, LOW); // FLASH OFF
delay(1000); // Tunggu sebentar sebelum mengulangi
}

void loop() {
    sensorValue = digitalRead(sensor);
    if(sensorValue == 0){
        if (Firebase.RTDB.setString(&fbdo, "/qwerty123/status", "1")) {
            Serial.println("Pesan berhasil ditulis");
        } else {
            Serial.print("Error: ");
            Serial.println(fbdo.errorReason());
        }
        delay(7000);
        digitalWrite(FLASH_LED_PIN, HIGH);
        delay(3000);
        if (!camera.capture().isOk()) {
            Serial.println(camera.exception.toString());
            return;
        }

        String fileName = "/qwerty123/image.jpg";
        String fileNameSv = "/qwerty123/interface/image_" + (String)millis() + ".jpg";
        Serial.println("File name: " + fileName);

        uint8_t* buf = camera.frame->buf; // Data gambar (buffer)
        size_t len = camera.frame->len; // Ukuran gambar dalam byte
        if (!Firebase.Storage.upload(&fbdo, STORAGE_BUCKET_ID, buf, len, fileName, "image/jpeg") || !Firebase.Storage.upload(&fbdo, STORAGE_BUCKET_ID, buf, len, fileNameSv, "image/jpeg")) {
            Serial.println(fbdo.errorReason());
        }

        Serial.println("success upload to firebase!");
        digitalWrite(FLASH_LED_PIN, LOW);

        if (Firebase.RTDB.setString(&fbdo, "/qwerty123/stateUp", "1")) {
            Serial.println("Pesan berhasil ditulis");
        } else {
            Serial.print("Error: ");
            Serial.println(fbdo.errorReason());
        }
    }
}
}

```

```

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <ESP8266WiFi.h>
#include <Firebase_ESP_Client.h>
#include "addons/TokenHelper.h"
#include "addons/RTDBHelper.h"

#define WIFI_SSID "eepis-mahya"
#define WIFI_KEY "1sampai9"
#define API_KEY "AIzaSyCF3UkvP6GN5nJswK2_IDt1NCaDxbamaXY"
#define DATABASE_URL "https://espcam-e3bb4-default-rtdb.asia-southeast1.firebaseio.com/"
#define relay 13
// Definisikan lebar dan tinggi OLED
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
SoftwareSerial mySerial(3, 1); // RX=D5, TX=D6
// Inisialisasi OLED display dengan alamat I2C 0x3C
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

bool signUpOk = false;
String teks = "";
String pesan = "";
int ledState;
bool isDisplayed = false; // Flag untuk menandai apakah teks sudah ditampilkan

void setup() {
  Serial.begin(115200);

  pinMode(relay, OUTPUT);

  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Pastikan alamat 0x3C benar
    Serial.println(F("SSD1306 allocation failed"));
    for (;;); // Berhenti jika OLED tidak terdeteksi
  }
  display.clearDisplay();
  display.display(); // Update untuk menghapus layar

```

```

WiFi.mode(WIFI_STA);
WiFi.begin(WIFI_SSID, WIFI_KEY);
while(WiFi.status() != WL_CONNECTED){
    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(SSD1306_BLACK, SSD1306_WHITE);
    display.setCursor(0, 10);
    display.println(".");
    display.display();
    delay(300);
}
Serial.println();
display.clearDisplay();
display.setTextSize(2);
display.setTextColor(SSD1306_BLACK, SSD1306_WHITE);
display.setCursor(0, 10);
display.println(WiFi.localIP());
display.display();

config.api_key = API_KEY;
config.database_url = DATABASE_URL;
if(Firebase.signUp(&config, &auth, "", "")){
    display.clearDisplay();
    display.setTextSize(2);
    display.setTextColor(SSD1306_BLACK, SSD1306_WHITE);
    display.setCursor(0, 10);
    display.println("sign up");
    display.display();
    signUpOk = true;
}else{
    display.clearDisplay();
    display.setTextSize(2);
    display.setTextColor(SSD1306_BLACK, SSD1306_WHITE);
    display.setCursor(0, 10);
    display.println(config.signer.signupError.message.c_str());
    display.display();
}
config.token_status_callback = tokenStatusCallback;
Firebase.begin(&config, &auth);
Firebase.reconnectWiFi(true);
}

```



```

void loop() {
  if(Firebase.RTDB.getInt(&fbdo, "/qwerty123/pesan")){
    if(fbdo.dataType() == "string"){
      pesan = fbdo.stringData();
      if (pesan != teks) {
        teks = pesan;
        display.clearDisplay();
        display.setTextSize(2);
        display.setTextColor(SSD1306_WHITE);
        display.setCursor(0, 10);
        display.println(pesan);
        display.display();
        Serial.println("Pesan baru: " + pesan); // Tampilkan di Serial Monitor juga
      }
    }
  }
  else{
    Serial.println("FAILED: " + fbdo.errorReason());
  }
}

```

```

if(Firebase.RTDB.getInt(&fbdo, "/qwerty123/buttonState")){
  if(fbdo.dataType() == "string"){
    String pesan = fbdo.stringData();
    Serial.println(pesan);
    if(asu == "1"){
      Serial.println("nyala");
      digitalWrite(relay, LOW);
    }else if(asu == "0"){
      Serial.println("mati");
      digitalWrite(relay, HIGH);
    }
  }
}
else{
  Serial.println("FAILED: " + fbdo.errorReason());
}

```

```

if(Firebase.RTDB.getInt(&fbdo, "/qwerty123/stateReg")){
  if(fbdo.dataType() == "string"){
    String stateRec = fbdo.stringData();
    if(stateRec == "1"){
      display.clearDisplay();
      display.setTextSize(2);
      display.setTextColor(SSD1306_WHITE);
      display.setCursor(0, 10);
      display.println("selamat datang tuan");
      display.display();
      digitalWrite(relay, HIGH);
      delay(5000);
      digitalWrite(relay, LOW);
      if (Firebase.RTDB.setString(&fbdo, "/qwerty123/stateReg", "0")) {
        Serial.println("ok");
      } else {
        Serial.print("Error: ");
        Serial.println(fbdo.errorReason());
      }
    }
  }
}

```

```

display.clearDisplay();
display.setTextSize(2);
display.setTextColor(SSD1306_WHITE);
display.setCursor(0, 10);
display.println(pesan);
display.display();
Serial.println("Pesan baru: " + pesan); // Tampilkan di Serial Monitor juga
} else if(stateRec == "2"){
  display.clearDisplay();
  display.setTextSize(2);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 10);
  display.println("anda siapa?");
  display.display();
  delay(5000);
  if (Firebase.RTDB.setString(&fbdo, "/qwerty123/stateReg", "0")) {
    Serial.println("ok");
  } else {
    Serial.print("Error: ");
    Serial.println(fbdo.errorReason());
  }
}

```

```

        display.clearDisplay();
        display.setTextSize(2);
        display.setTextColor(SSD1306_WHITE);
        display.setCursor(0, 10);
        display.println(pesan);
        display.display();
        Serial.println("Pesan baru: " + pesan); // Tampilkan di Serial Monitor juga
    }
}
}
else{
    Serial.println("FAILED: " + fbdo.errorReason());
}

if(Firebase.RTDB.getInt(&fbdo, "/qwerty123/status")){
    if(fbdo.dataType() == "string"){
        String stateRec = fbdo.stringData();
        if(stateRec == "1"){
            display.clearDisplay();
            display.setTextSize(1);
            display.setTextColor(SSD1306_WHITE);
            display.setCursor(0, 10);
            display.println("letakkan wajah +-35cm di depan kamera");
            display.display();
            delay(10000);
        }else if(stateRec == "2"){
            display.clearDisplay();
            display.setTextSize(1);
            display.setTextColor(SSD1306_WHITE);
            display.setCursor(0, 10);
            display.println("proses pengenalan wajah...");
            display.display();
        }
    }
}
else{
    Serial.println("FAILED: " + fbdo.errorReason());
}
}

```

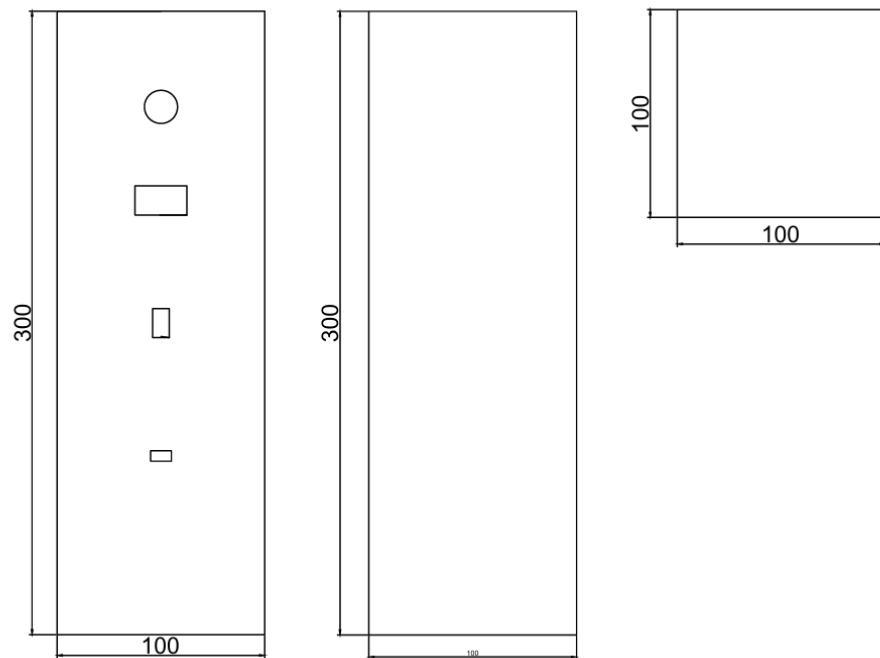
```

if(Firebase.RTDB.getInt(&fbdo, "/qwerty123/errCam")){
    if(fbdo.dataType() == "string"){
        String stateRec = fbdo.stringData();
        if(stateRec == "1"){
            display.clearDisplay();
            display.setTextSize(1);
            display.setTextColor(SSD1306_WHITE);
            display.setCursor(0, 10);
            display.println("gagal menangkap gambar");
            display.display();
            delay(10000);
            display.clearDisplay();
        }
    }
}
else{
    Serial.println("FAILED: " + fbdo.errorReason());
}
}

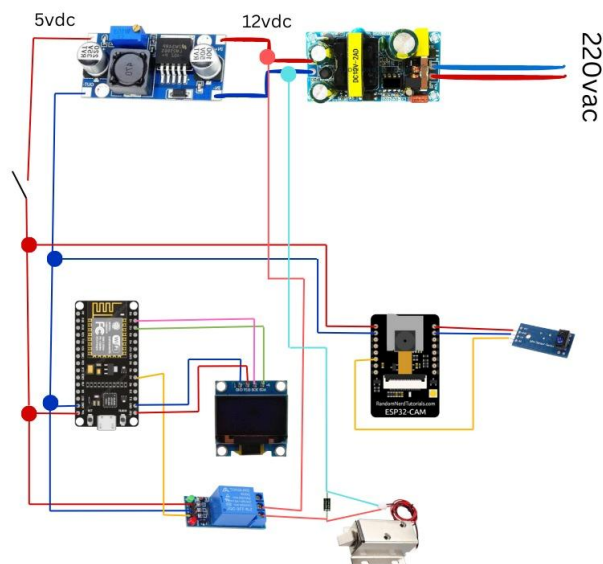
```


4.1.4. DESAIN ALAT

4.1.4.1 Desain case



4.1.4.2 Desain Perangkat



4.2 . FACE RECOGNITION

Pada proyek ini, face recognition atau pengenalan wajah menggunakan model DeepLearning siamese neural network (SNN) dan oneshot learning. Dalam metode ini, oneshot learning dapat mengenali wajah hanya dengan satu contoh atau sample data wajah saja.

4.2.1. Siamese neural network (SNN)

SNN merupakan model dalam DeepLearning yang digunakan untuk membandingkan antara dua gambar. Terdapat tiga parameter dalam SNN. Yaitu, anchor, positive, dan negative. Anchor merupakan data yang akan diuji (sample A), positive merupakan data yang sama dengan data yang diuji tetapi memiliki karakteristik yang berbeda(data A), dan negative merupakan data yang sangat berbeda (data B). SNN akan membandingkan data anchor dengan data positive untuk model mempelajari kemiripan antara data yang sama dan membandingkan data anchor dengan data negative untuk model mempelajari perbedaan antara data yang berbeda.



4.2.2. OneShot Learning

OneShot learning merupakan pendekatan dari machine learning untuk membedakan gambar hanya dengan satu contoh saja. OneShot learning memanfaatkan model SNN yang telah dilatih untuk pengenalan. Mesin akan diberikan satu gambar, kemudian oneshot learning akan mendeteksi gambar tersebut apakah dikenali atau tidak berdasarkan model SNN yang telah dilatih.



4.3 APLIKASI ANDROID

4.3.1 Fitur

1. Registrasi

- Pengguna mendaftar dengan memasukkan UID, username, password, dan konfirmasi password.
- Sistem memeriksa:
 - Jika UID valid dan password cocok dengan konfirmasi, data disimpan di database.
 - Jika tidak, pengguna diminta memperbaiki input.

2. Login

- Pengguna login dengan UID dan password.
- Sistem mengecek apakah data ada di database:
 - Jika valid, pengguna diarahkan ke halaman beranda.
 - Jika tidak valid, pengguna diminta login ulang.

3. Halaman Beranda

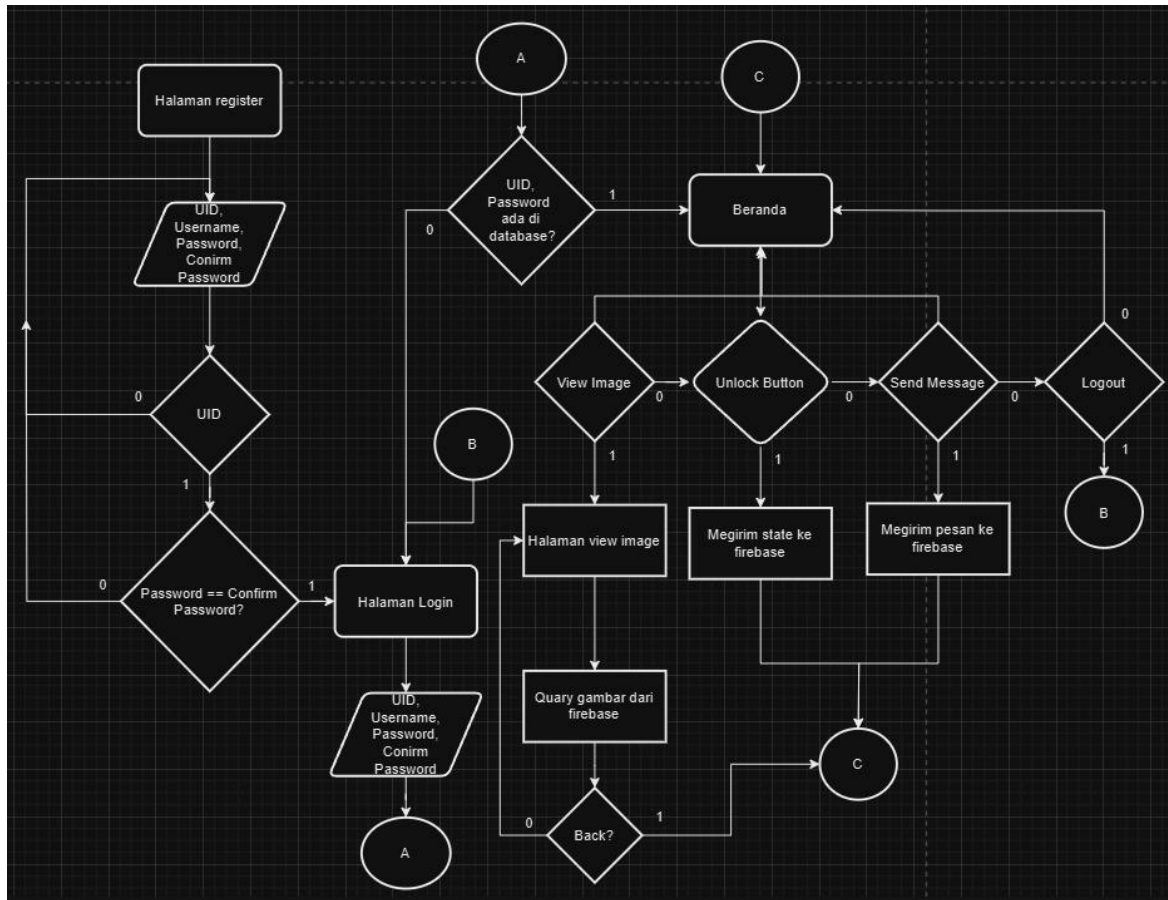
Pada beranda, pengguna dapat memilih fitur berikut:

1. View Image:
 - Menampilkan gambar yang diambil dari Firebase.
 - Ada opsi untuk kembali ke beranda.
2. Unlock Button:
 - Membuka kunci dengan mengirim status ke Firebase.
3. Send Message:
 - Mengirim pesan ke Firebase.
4. Logout:
 - Keluar dari aplikasi dan kembali ke halaman login.

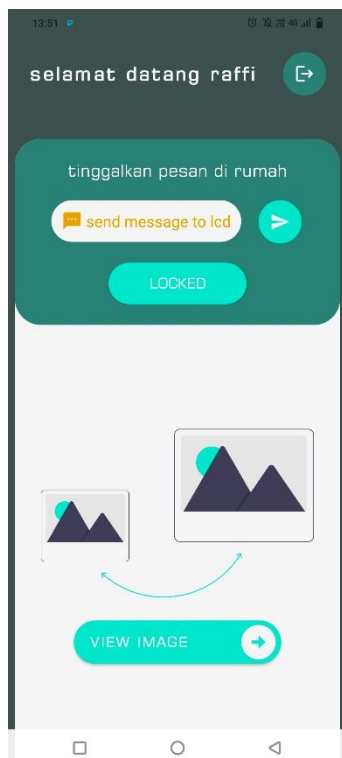
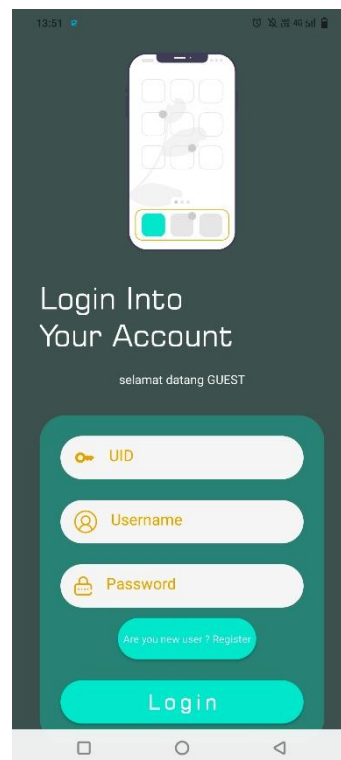
4. Integrasi Firebase

- Firebase digunakan untuk:
 - Menyimpan data pengguna.
 - Mengelola gambar.
 - Mengirim status tombol unlock.
 - Menyimpan pesan yang dikirimkan.

4.3.2 Flowchat cara kerja aplikasi



4.3.3 Tampilan Aplikasi



4.3.3 Program

- Logic welcome screen

```
package com.example.smartlock

import android.os.Bundle
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import android.content.Intent
import android.os.Handler
import android.os.Looper

class splashAct : AppCompatActivity() {
    private fun goToMainActivity() {
        Intent(packageContext: this, login::class.java).also { it: Intent
            startActivity(it)
            finish()
        }
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.splashkit)

        Handler(Looper.getMainLooper()).postDelayed({
            goToMainActivity()
        }, delayMillis: 2000L)
    }
}
```

- Register

```
package com.example.smartlock

import android.content.Intent
import android.os.Bundle
import android.util.Log
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import com.google.firebase.Firebase
import com.google.firebase.database.DataSnapshot
import com.google.firebase.database.DatabaseError
import com.google.firebase.database.ValueEventListener
import com.google.firebase.database.database
```

```

class registerPage : AppCompatActivity() {
    private lateinit var uid:EditText
    private lateinit var username:EditText
    private lateinit var password:EditText
    private lateinit var confPass:EditText
    private lateinit var registerSubmit : Button
    private lateinit var peringatan:TextView

    private fun initComponents(){
        uid = findViewById(R.id.editTextTextSignUpUID)
        username = findViewById(R.id.editTextTextSignUpUsername)
        password = findViewById(R.id.editTextTextSignUpPassword)
        confPass = findViewById(R.id.editTextTextSignUpConfirmPassword)
        registerSubmit = findViewById(R.id.btnSignUp)
        peringatan = findViewById(R.id.alertReg)
    }

    private fun pindahHalaman(){
        Intent( packageContext: this, login::class.java).also{ it: Intent
            startActivity(it)
            Toast.makeText( context: this, resources.getString(R.string.akunberhasil), Toast.LENGTH_SHORT).show()
        }
    }
}

```

```

private fun register(uid:String, username:String, password:String, confPass:String){
    val database = Firebase.database
    val myRef = database.getReference(uid)

    if(uid == "" || username == "" || password == "" || confPass == ""){
        peringatan.text = resources.getString(R.string.uidusernameSalah)
    }else{
        myRef.addListenerForSingleValueEvent(object : ValueEventListener {
            override fun onDataChange(snapshot: DataSnapshot) {
                if(snapshot.exists()) {
                    Log.i( tag: "cekreg", uid)
                    if (password == confPass) {
                        myRef.child(username).setValue(password)
                        pindahHalaman()
                    } else {
                        peringatan.text = resources.getString(R.string.passTidakCocokReg)
                    }
                }else{
                    peringatan.text = resources.getString(R.string.uidusernameSalah)
                }
            }
        })
    }
}

```

```

        override fun onCancelled(error: DatabaseError) {
            Log.w( tag: "registerGagal", msg: "onCancelled: ", error.toException())
        }
    })
}

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    enableEdgeToEdge()
    setContentView(R.layout.registerpage)

    initComponents()

    registerSubmit.setOnClickListener { it: View!
        register(uid.text.toString(), username.text.toString(), password.text.toString(), confPass.text.to:
    }
}
}

```



```
package com.example.smartlock

import android.content.Context
import android.content.Intent
import android.os.Bundle
import android.util.Log
import android.widget.Button
import android.widget.EditText
import android.widget.ImageButton
import android.widget.TextView
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import com.google.firebase.Firebase
import com.google.firebase.database.DataSnapshot
import com.google.firebase.database.DatabaseError
import com.google.firebase.database.ValueEventListener
import com.google.firebase.database.database
import com.google.firebase.database.getValue

class MainActivity2 : AppCompatActivity() {
    private lateinit var welcome:TextView
    private lateinit var viewImageButt:Button
    private lateinit var logoutButton: ImageButton
    private lateinit var lockButton: Button
    private lateinit var isiPesan: EditText
    private lateinit var sendBut:ImageButton

    private fun initComponents(){
        welcome = findViewById(R.id.textView4)
        viewImageButt = findViewById(R.id.button4)
        logoutButton = findViewById(R.id.imageButton3)
        lockButton = findViewById(R.id.button2)
        isiPesan = findViewById(R.id.editTextText)
        sendBut = findViewById(R.id.imageButton)
    }

    private fun pindahPageLogin(){
        Intent( packageContext: this, login::class.java).also { it: Intent
            startActivity(it)
        }
    }
}
```



```

private fun logOut(){
    val sharedPref = getSharedPreferences(resources.getString(R.string.keyNama), Context.MODE_PRIVATE)
    sharedPref.edit().clear().apply()
    pindahPageLogin()
}

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    enableEdgeToEdge()
    setContentView(R.layout.mainpage)

    initComponents()

    val sharedPref = getSharedPreferences(resources.getString(R.string.keyNama), Context.MODE_PRIVATE)
    val nama = sharedPref.getString("UID", null)
    val nrp = sharedPref.getString("username", null)
    val database = Firebase.database
    val myRef = database.getReference(path: "${nama}/buttonState")
    val kirimPesanan = database.getReference(path: "${nama}/pesan")

```

```

myRef.addValueEventListener(object : ValueEventListener {
    override fun onDataChange(snapshot: DataSnapshot) {
        val state = snapshot.getValue<String>().toString()
        if(state == "0"){
            lockButton.text = "unlocked"
        }else{
            lockButton.text = "locked"
        }
    }

    override fun onCancelled(error: DatabaseError) {
        TODO(reason: "Not yet implemented")
    }
})

welcome.text = resources.getString(R.string.welcomeLogged, nrp)

logoutButton.setOnClickListener { it: View!
    logOut()
}

```

```

lockButton.setOnClickListener { it: View!
    myRef.addListenerForSingleValueEvent(object : ValueEventListener {
        override fun onDataChange(snapshot: DataSnapshot) {
            val value = snapshot.getValue<String>()
            if(value != "1"){
                myRef.setValue("1")
                lockButton.text = "locked"
            }else if(value == "1"){
                myRef.setValue("0")
                lockButton.text = "unlocked"
            }
        }
    })
}

override fun onCancelled(error: DatabaseError) {
    Log.w( tag: "errorQuery", msg: "onCancelled: ", error.toException())
}

sendBut.setOnClickListener { it: View!
    kirimPesan.setValue(isiPesan.text.toString())
    Toast.makeText( context: this, resources.getString(R.string.pesanterkirim), Toast.LENGTH_SHORT).
}

```

```

viewImageBut.setOnClickListener { it: View!
    val intent = Intent( packageContext: this, ImagesActivity::class.java)
    startActivity(intent)
}
}
}

```

- Login

```

package com.example.smartlock

import android.content.Context
import android.content.Intent
import android.os.Bundle
import android.util.Log
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import com.google.firebase.Firebase
import com.google.firebase.database.DataSnapshot
import com.google.firebase.database.DatabaseError
import com.google.firebase.database.ValueEventListener
import com.google.firebase.database.database
import com.google.firebase.database.getValue

```

```

class login : AppCompatActivity() {
    private lateinit var uid:EditText
    private lateinit var username:EditText
    private lateinit var password:EditText
    private lateinit var tombol:Button
    private lateinit var sayHello:TextView
    private lateinit var registerButt:Button

    private fun initComponents(){
        uid = findViewById(R.id.editTextTextUID)
        username = findViewById(R.id.editTextTextUserName)
        password = findViewById(R.id.editTextTextPassword)
        tombol = findViewById(R.id.btnLogin)
        sayHello = findViewById(R.id.alert)
        registerButt = findViewById(R.id.butReg)
    }

    private fun pindahPage(){
        Intent( packageContext: this, MainActivity2::class.java).also { it: Intent
        }
        startActivity(it)
    }

    private fun pindahReg(){
        Intent( packageContext: this, registerPage::class.java).also{ it: Intent
        }
        startActivity(it)
    }
}

```

```

private fun loginFun(UID:String, username:String, passwordIn:String) {
    val database = Firebase.database
    val myRef = database.getReference( path: "${UID}/${username}")
    val sharedPref = getSharedPreferences(resources.getString(R.string.keyNama), Context.MODE_PRIVATE)

    initComponents()

    if(UID == "" || username == "" || passwordIn == ""){
        sayHello.text = resources.getString(R.string.uidusernameSalah)
    }else{
        myRef.addListenerForSingleValueEvent(object : ValueEventListener {
            override fun onDataChange(snapshot: DataSnapshot) {
                if(snapshot.exists()){
                    val password = snapshot.getValue<String>()
                    if(passwordIn == password){
                        with(sharedPref.edit()){ this: SharedPreferences.Editor!
                            putString("UID", UID)
                            putString("username", username)
                            apply()
                        }
                        pindahPage()
                    }else{
                        sayHello.text = resources.getString(R.string.passwordSalah)
                    }
                }else{
                    sayHello.text = resources.getString(R.string.uidusernameSalah)
                }
            }
        })
    }
}

```

```

        override fun onCancelled(error: DatabaseError) {
            Log.w( tag: "errorLogin", msg: "onCancelled: ", error.toException())
        }
    })
}
}

```

```

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    enableEdgeToEdge()
    setContentView(R.layout.Login)

    initComponents()

    val sharedPref = getSharedPreferences(resources.getString(R.string.keyNama), Context.MODE_PRIVATE)
    val namaaa = sharedPref.getString("username", "GUEST")
    sayHello.text = resources.getString(R.string.welcomeLogged, namaaa)

    if(namaaa != "GUEST"){
        pindahPage()
    }

    tombol.setOnClickListener { it: View!
        val UID = uid.text.toString()
        val username = username.text.toString()
        val password = password.text.toString()

        loginFun(UID, username, password)
    }
}

```

```

        registerButt.setOnClickListener { it: View!
            pindahReg()
        }
    }
}
}

```


- facerecog.facerecog1

```
from downloadFunc import download_image, delete_image
from cekState import read_data, write_data
import time

import tensorflow as tf
from layers import L1Dist
import os
import numpy as np

#build app and layout
|
def preprocess(file_path):
    #read in image from fie path
    byte_img = tf.io.read_file(file_path)
    #load in the image
    img = tf.io.decode_jpeg(byte_img)
    #preprocessing - resizing to 100x100
    img = tf.image.resize(img, (100,100))
    img = img / 255.0
    return img

#verification func
def verify(*args):
    detection_threshold = 0.95
    verification_threshold = 0.95

    file_path = "qwerty123/image.jpg" # Ganti dengan path file di Storage Anda
    local_path = "application_data/input_img/downloadednew_image.jpg"

    download_image(file_path, local_path)

    time.sleep(5)
```

```
results = []
for image in os.listdir(os.path.join('application_data', 'verification_img')):
    input_img = preprocess(os.path.join('application_data', 'input_img', 'downloadednew_image.jpg'))
    validation_img = preprocess(os.path.join('application_data', 'verification_img', image))

    #make prediction
    result = model.predict(list(np.expand_dims([input_img, validation_img], axis=1)))
    results.append(result)

#detection threshold: metrc above which a prediction is considered positive
detection = np.sum(np.array(results) > detection_threshold)

#verification threshold
verification = detection / len(os.listdir(os.path.join('application_data', 'verification_img')))
verified = verification / verification_threshold

#set verification text
# verification_Label.text = 'Verified' if verification > verification_threshold else 'Unverified'
if verification > verification_threshold:
    print("abyaz")
    write_data("qwerty123/status", "0")
    write_data("qwerty123/stateReg", "1")
    file_to_delete = "qwerty123/image.jpg" # Path file di Firebase Storage
    write_data("qwerty123/stateUp", "0")
    delete_image(file_to_delete)
```

```

        else:
            print("unknown")
            write_data("qwerty123/status", "0")
            write_data("qwerty123/stateReg", "2")
            file_to_delete = "qwerty123/image.jpg" # Path file di Firebase Storage
            write_data("qwerty123/stateUp", "0")
            delete_image(file_to_delete)

    print(verification)

    return results, verified

while True:
    state = read_data("qwerty123/stateUp")
    if state == "1":
        #preprocess Load image and convert to 100x100
        write_data("qwerty123/status", "2")
        model = tf.keras.models.load_model('siamesemodelRev2.h5', custom_objects={'L1Dist':L1Dist})
        verify()
    time.sleep(5)

```

- facerecog.Layers

```

#custom L1 distance Layer module
import tensorflow as tf
from tensorflow.keras.layers import Layer

#custom L1 Distance Layer from colab
#siamese L1 Distance class
class L1Dist(Layer):
    #Init method - inheritance
    def __init__(self, **kwargs):
        super().__init__()
    #magic happens here - similarity calculation
    def call(self, input_embedding, validation_embedding):
        return tf.math.abs(tf.convert_to_tensor(input_embedding) - tf.convert_to_tensor(validation_embedding))

```

- facerecog.query gambar firebase1

```
import firebase_admin
from firebase_admin import credentials, storage

# Inisialisasi aplikasi Firebase
cred = credentials.Certificate("firebasekey.json") # Ganti dengan path file kunci Anda
firebase_admin.initialize_app(cred, {
    'storageBucket': 'espcam-e3bb4.appspot.com'
}, name="downloadImg")

# Fungsi untuk mengunduh file dari Firebase Storage
def download_image(file_path, local_path):
    try:
        app = firebase_admin.get_app("downloadImg")
        bucket = storage.bucket(app=app)
        blob = bucket.blob(file_path)

        # Unduh file ke path lokal
        blob.download_to_filename(local_path)
        print(f"Gambar berhasil diunduh ke: {local_path}")
        return True # Return True jika berhasil
    except Exception as e:
        print(f"Error saat mengunduh gambar: {e}")
        return False # Return False jika gagal
```

```
def delete_image(file_path):
    try:
        app = firebase_admin.get_app("downloadImg")
        bucket = storage.bucket(app=app)
        blob = bucket.blob(file_path)
        blob.delete()
        print(f"File '{file_path}' berhasil dihapus dari Firebase Storage.")
    except Exception as e:
        print(f"Error saat menghapus file '{file_path}': {e}")
```

- facerecog.query state

```
import firebase_admin
from firebase_admin import credentials, db

# Inisialisasi aplikasi Firebase
cred2 = credentials.Certificate("firebasekey.json") # Ganti dengan path file kunci JSON Anda
firebase_admin.initialize_app(cred2, {
    'databaseURL': 'https://espcam-e3bb4-default-rtdb.asia-southeast1.firebaseio.com/'
}, name= "cekState")

def read_data(reference):
    try:
        app = firebase_admin.get_app("cekState")
        ref = db.reference(reference, app=app)
        data = ref.get()
        print(f"Data berhasil dibaca dari: {reference}")
        return data
    except Exception as e:
        print(f"Error saat membaca data: {e}")
        return None

def write_data(reference, data):
    try:
        app = firebase_admin.get_app("cekState")
        ref = db.reference(reference, app=app)
        ref.set(data)
        print(f"Data berhasil ditulis ke: {reference}")
        return True
    except Exception as e:
        print(f"Error saat menulis data: {e}")
        return False
```


5. DOKUMENTASI



6. ANALISA DAN KESIMPULAN

ANALISA

Proyek ini bertujuan untuk menjawab tantangan keamanan modern dengan memanfaatkan teknologi biometrik, khususnya pengenalan wajah, sebagai solusi inovatif. Sistem keamanan tradisional, seperti kunci fisik atau PIN, memiliki kelemahan seperti risiko kehilangan atau peretasan. Dengan menggunakan teknologi IoT yang terhubung melalui aplikasi Android, sistem ini memungkinkan pengelolaan akses yang lebih praktis dan aman. Proyek ini dirancang untuk memenuhi kebutuhan keamanan di berbagai lingkungan, seperti rumah atau kantor, dengan mengutamakan kenyamanan dan efisiensi.

Teknologi yang digunakan dalam proyek ini meliputi metode deep learning berbasis **Siamese Neural Network (SNN)** untuk pengenalan wajah, serta integrasi perangkat keras ESP32-CAM dan ESP8266 untuk akuisisi data dan logika kontrol. Firebase dipilih sebagai media penyimpanan real-time yang mendukung komunikasi antara perangkat keras dan aplikasi Android. Namun, sistem ini masih memiliki keterbatasan, seperti hanya dapat mengenali satu wajah dan ketergantungan pada perangkat keras tertentu, sehingga pengembangan lebih lanjut diperlukan untuk meningkatkan fleksibilitas dan skalabilitas.

Implementasi sistem ini melibatkan alur kerja yang terintegrasi, mulai dari pendeteksian wajah dengan sensor infrared hingga kontrol perangkat keras melalui ESP8266 berdasarkan hasil autentikasi wajah. Aplikasi Android dirancang sebagai antarmuka utama yang memungkinkan pengguna untuk mengelola akses, melihat log, dan mengontrol kunci secara real-time. Meskipun desainnya fungsional, pengembangan lebih lanjut diperlukan untuk meningkatkan kompatibilitas dan pengalaman pengguna, terutama untuk mendukung berbagai versi Android.

KESIMPULAN

Kesimpulannya, proyek **Smartlock berbasis Face Recognition dan aplikasi Android dengan User Authorization** ini berhasil menghadirkan solusi keamanan yang modern dan canggih. Kombinasi teknologi biometrik, IoT, dan aplikasi mobile memberikan pengalaman yang lebih aman dan efisien dibandingkan metode tradisional. Untuk ke depan, pengembangan sistem dapat difokuskan pada peningkatan akurasi pengenalan wajah dalam berbagai kondisi, serta penambahan fitur agar lebih adaptif terhadap kebutuhan pengguna yang lebih luas.