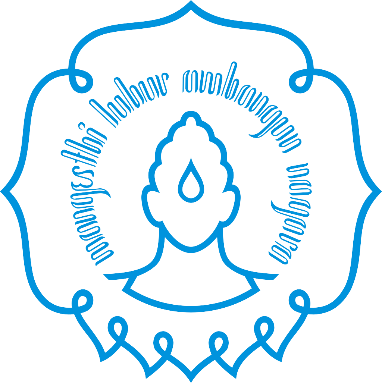
**LAPORAN UJIAN AKHIR SEMESTER**

**SISTEM KEAMANAN DATA, IoT, dan MOBILE**



**Disusun oleh :**

Puput Surya Ningtyas (V3923015)

**Dosen**

Yusuf Fadila Rachman. S.Kom., M.Kom

**PS D-III TEKNIK INFORMATIKA**

**SEKOLAH VOKASI**

**UNIVERSITAS SEBELAS MARET**

**2024**

[1 IoT 3](#_Toc186150418)

[1.1 Gambar dan penjelasan skema IoT 3](#_Toc186150419)

[1.2 Penjelasan Flowchart Iot 4](#_Toc186150420)

[1.3 UI/UX Aplikasi (Blynk) 6](#_Toc186150421)

[1.4 Penjelasan kode Implementasi 6](#_Toc186150422)

[1.4.1 Kode 6](#_Toc186150423)

[1.4.2 Penjelasan 8](#_Toc186150424)

[1.5 Hasil pengujian 9](#_Toc186150425)

[1.6 Link video 10](#_Toc186150426)

[2 SISTEM KEAMANAN DATA 10](#_Toc186150427)

[2.1 Penjelasan algoritma Bcyript 10](#_Toc186150428)

[2.2 Penjelasan kode 11](#_Toc186150429)

[2.3 Flowchart 12](#_Toc186150430)

[2.4 Screenshot hasil 13](#_Toc186150431)

[3 MOBILE 13](#_Toc186150432)

[3.1 UI/UX Mobile 13](#_Toc186150433)

[3.1.1 Register 13](#_Toc186150434)

[3.1.2 Login 14](#_Toc186150435)

[3.1.3 Dasboard admin 15](#_Toc186150436)

[3.1.4 Data pengguna 16](#_Toc186150437)

[3.1.5 Peta 16](#_Toc186150438)

[3.1.6 Daftar chat 17](#_Toc186150439)

[3.1.7 Pengaduan dan laporan 18](#_Toc186150440)

[3.1.8 History 19](#_Toc186150441)

[3.1.9 Setting profil 20](#_Toc186150442)

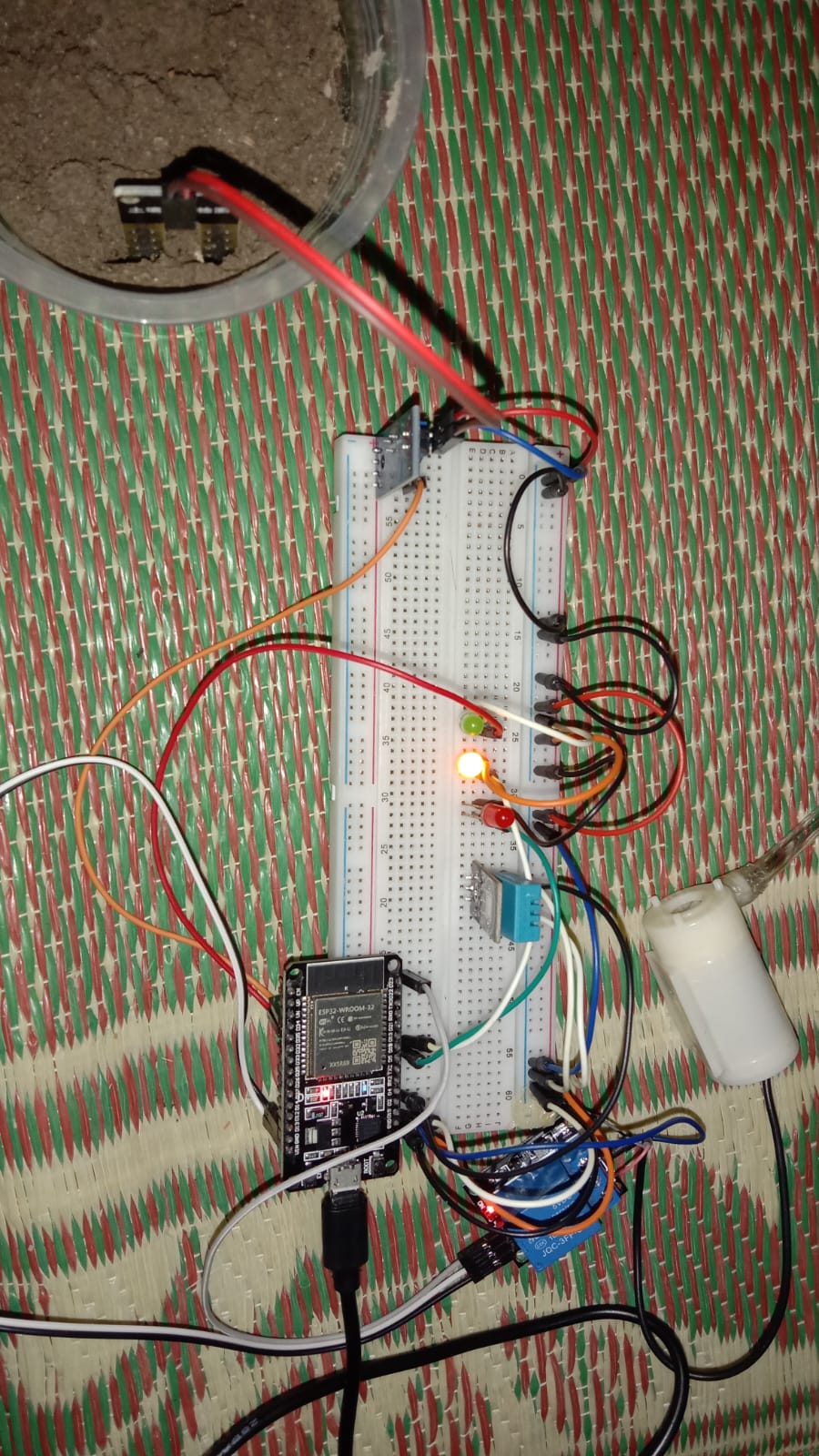
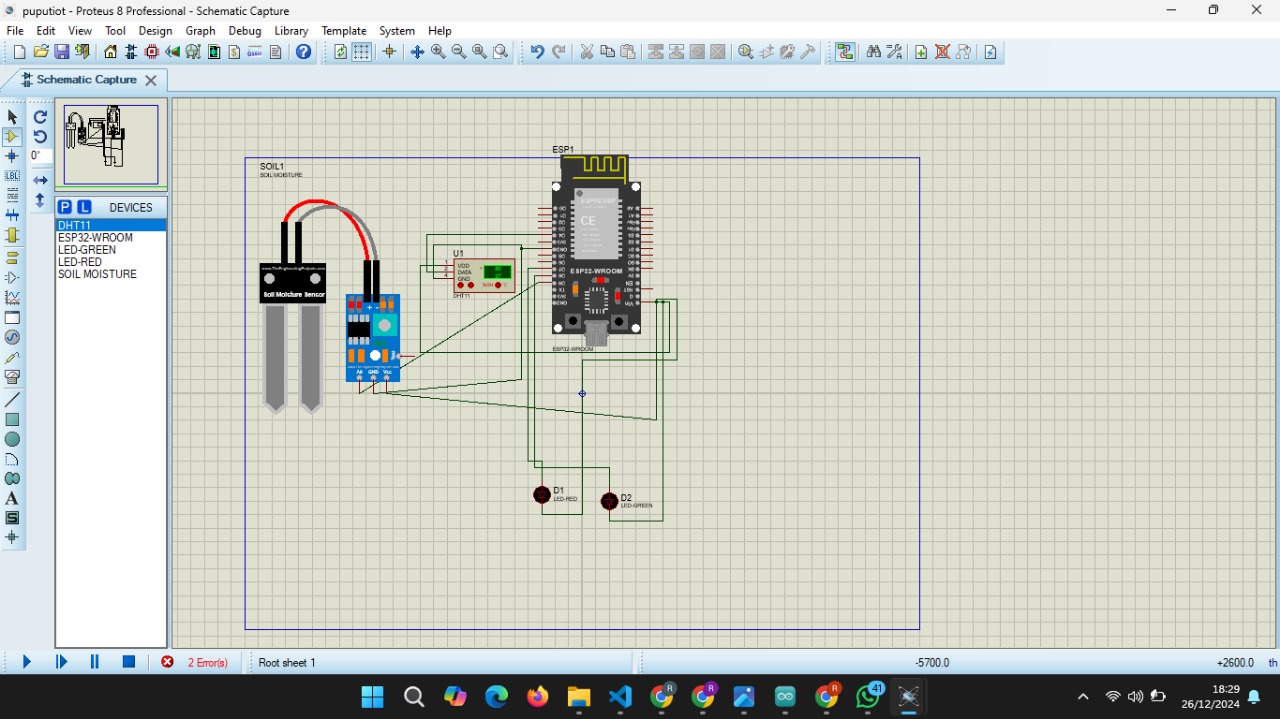
[3.2 Penjelasan kode program 20](#_Toc186150443)

[3.3 Link apk 20](#_Toc186150444)

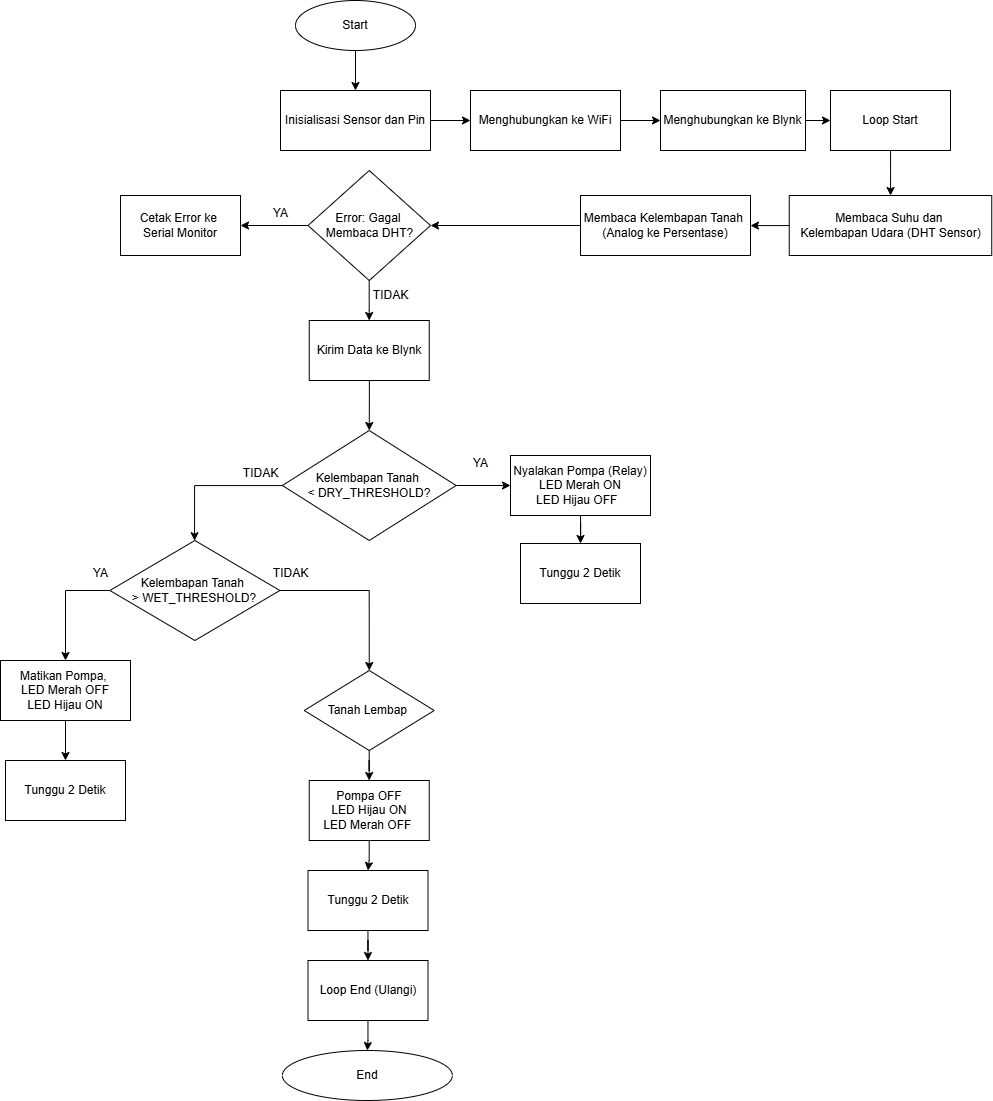
[4 Referensi 21](#_Toc186150445)

# IoT

## Gambar dan penjelasan skema IoT



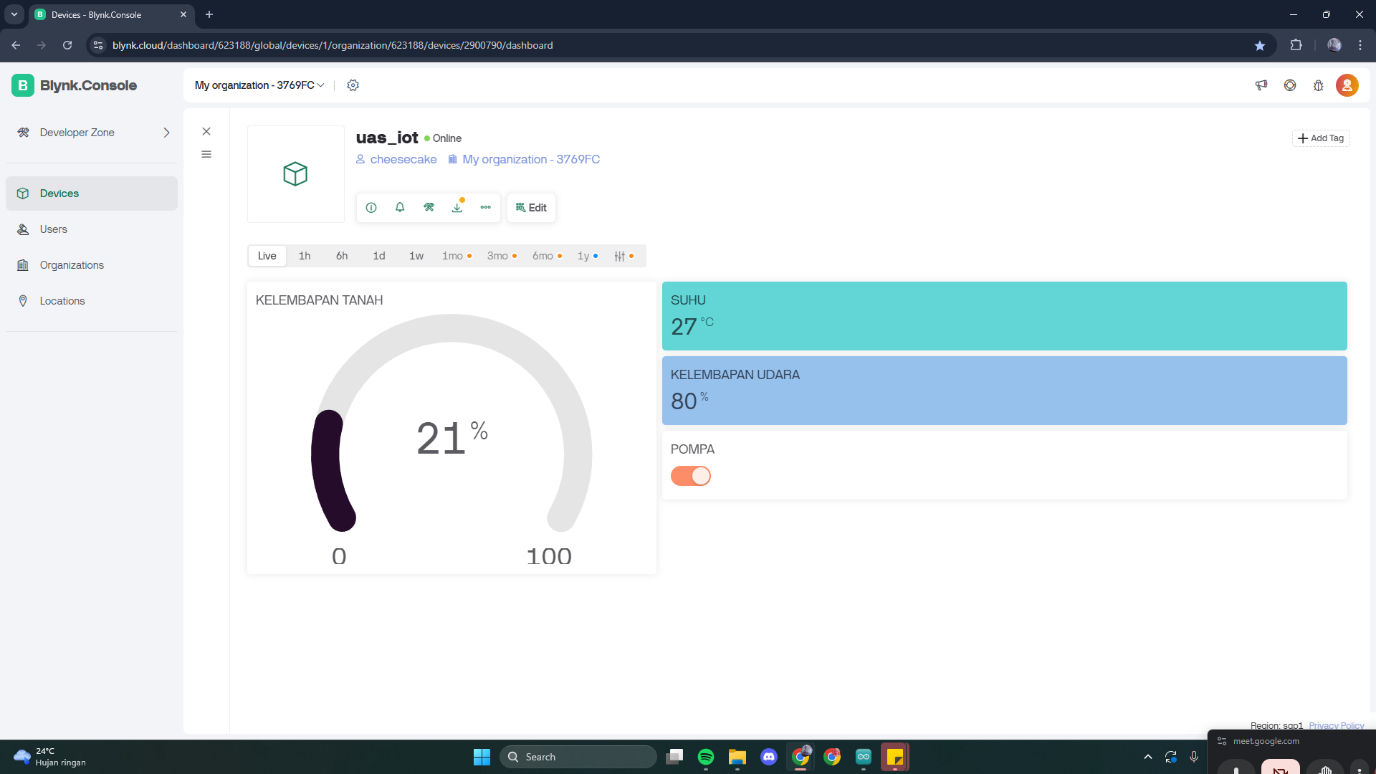
## Penjelasan Flowchart Iot



Berikut penjelasan setiap bagian:

1. **Start** Proses dimulai dengan inisialisasi perangkat dan pengaturan awal.
2. **Inisialisasi Sensor dan Pin**
   * Sistem menginisialisasi sensor DHT, pin digital, dan koneksi serial.
   * Ini memastikan perangkat keras siap untuk digunakan.
3. **Menghubungkan ke WiFi**
   * Sistem mencoba menghubungkan perangkat ke jaringan WiFi menggunakan kredensial yang telah disediakan.
   * Setelah terhubung, sistem melanjutkan ke langkah berikutnya.
4. **Menghubungkan ke Blynk**
   * Setelah WiFi tersambung, perangkat akan menginisialisasi koneksi ke platform Blynk untuk pengiriman dan penerimaan data.
5. **Loop Start**
   * Sistem memasuki loop utama untuk membaca data dan melakukan kontrol berdasarkan kondisi yang terdeteksi.
6. **Membaca Suhu dan Kelembapan Udara (DHT Sensor)**
   * Sensor DHT membaca data suhu dan kelembapan udara.
   * Jika terjadi kesalahan, sistem mencetak pesan error ke Serial Monitor.
7. **Membaca Kelembapan Tanah (Analog ke Persentase)**
   * Sensor kelembapan tanah membaca nilai analog, yang kemudian dikonversi menjadi persentase untuk evaluasi.
8. **Kirim Data ke Blynk**
   * Data suhu, kelembapan udara, dan kelembapan tanah dikirim ke server Blynk untuk ditampilkan di aplikasi pengguna.
9. **Evaluasi Kelembapan Tanah**
   * **Jika kelembapan tanah < DRY\_THRESHOLD:**
     + Sistem menganggap tanah kering.
     + Pompa dinyalakan, LED merah menyala, dan LED hijau mati.
     + Setelah 2 detik, pompa dimatikan.
   * **Jika kelembapan tanah > WET\_THRESHOLD:**
     + Sistem menganggap tanah terlalu basah.
     + Pompa dimatikan, LED hijau menyala, dan LED merah mati.
   * **Jika kelembapan tanah di antara ambang batas:**
     + Sistem menganggap tanah cukup lembap.
     + Pompa tetap mati, LED hijau menyala, dan LED merah mati.
10. **Tunggu 2 Detik**
    * Delay 2 detik diberikan sebelum kembali mengevaluasi kondisi.
11. **Loop End (Ulangi)**
    * Sistem kembali ke awal loop untuk terus membaca data sensor dan melakukan kontrol.
12. **End**
    * Flowchart ini diakhiri di tahap loop (sistem terus berjalan selama perangkat aktif).

## UI/UX Aplikasi (Blynk)



## Penjelasan kode Implementasi

### Kode

#define BLYNK\_TEMPLATE\_ID "TMPL6OIR0Hr4r"

#define BLYNK\_TEMPLATE\_NAME "uasiot"

#define BLYNK\_AUTH\_TOKEN "r7W9-mskFiNHggQN3KXTTJtrH3rgH3qH"

#include <Wire.h>

#include <WiFi.h>

#include <DHT.h>

#include <BlynkSimpleEsp32.h>

// Pin Definitions

#define DHTPIN 2

#define DHTTYPE DHT11

#define SOILPIN 34

#define RELAYPIN 23

#define GREEN\_LED 35

#define RED\_LED 5

// Virtual Pin untuk Blynk

#define VIRTUAL\_TEMP V2

#define VIRTUAL\_HUMIDITY V1

#define VIRTUAL\_SOIL V4

#define VIRTUAL\_PUMP V3

// WiFi credentials

const char\* ssid = "CheeseCake";

const char\* password = "godot4444";

// Thresholds

const int DRY\_THRESHOLD = 40;

const int WET\_THRESHOLD = 80;

// DHT Sensor

DHT dht(DHTPIN, DHTTYPE);

// Waktu terakhir pompa menyala

unsigned long lastPumpTime = 0;

const unsigned long pumpInterval = 86400000; // 24 jam dalam milidetik

void setup() {

  Serial.begin(115200);

  // Initialize DHT Sensor

  dht.begin();

  // Initialize Pins

  pinMode(SOILPIN, INPUT);

  pinMode(RELAYPIN, OUTPUT);

  pinMode(GREEN\_LED, OUTPUT);

  pinMode(RED\_LED, OUTPUT);

  // Default States

  digitalWrite(RELAYPIN, LOW);

  digitalWrite(GREEN\_LED, LOW);

  digitalWrite(RED\_LED, LOW);

  // Connect to WiFi

  Serial.println("Menghubungkan ke WiFi...");

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL\_CONNECTED) {

      delay(1000);

      Serial.println("Menghubungkan...");

  }

  Serial.println("WiFi terhubung.");

  // Initialize Blynk

  Blynk.begin(BLYNK\_AUTH\_TOKEN, ssid, password);

  Serial.println("Terhubung ke Blynk.");

}

void loop() {

  // Membaca suhu dan kelembapan udara

  float suhu = dht.readTemperature();

  float kelembapan = dht.readHumidity();

  // Membaca nilai kelembapan tanah

  int soilMoistureValue = analogRead(SOILPIN);

  float soilMoisture = map(soilMoistureValue, 0, 4095, 0, 100);

  // Kirim data ke Blynk

  Blynk.virtualWrite(VIRTUAL\_TEMP, suhu);

  Blynk.virtualWrite(VIRTUAL\_SOIL, soilMoisture);

  Blynk.virtualWrite(VIRTUAL\_HUMIDITY, kelembapan);

  // Logika kontrol untuk delay 24 jam

  unsigned long currentTime = millis();

  if (currentTime - lastPumpTime >= pumpInterval && soilMoisture < DRY\_THRESHOLD) {

    // Tanah kering, pompa ON

    digitalWrite(RELAYPIN, HIGH);

    digitalWrite(RED\_LED, HIGH);

    digitalWrite(GREEN\_LED, LOW);

    Blynk.virtualWrite(VIRTUAL\_PUMP, 1);

    Serial.println("Kondisi: Tanah kering, pompa ON.");

    delay(2000); // Pompa menyala selama 2 detik

    digitalWrite(RELAYPIN, LOW);

    Serial.println("Pompa OFF setelah 2 detik.");

    lastPumpTime = currentTime; // Perbarui waktu terakhir pompa menyala

  } else if (soilMoisture > WET\_THRESHOLD) {

    digitalWrite(RELAYPIN, LOW);

    digitalWrite(GREEN\_LED, HIGH);

    digitalWrite(RED\_LED, LOW);

    Blynk.virtualWrite(VIRTUAL\_PUMP, 0);

    Serial.println("Kondisi: Tanah basah, pompa OFF.");

  } else {

    digitalWrite(RELAYPIN, LOW);

    digitalWrite(RED\_LED, LOW);

    digitalWrite(GREEN\_LED, HIGH);

    Blynk.virtualWrite(VIRTUAL\_PUMP, 0);

    Serial.println("Kondisi: Tanah cukup lembap.");

  }

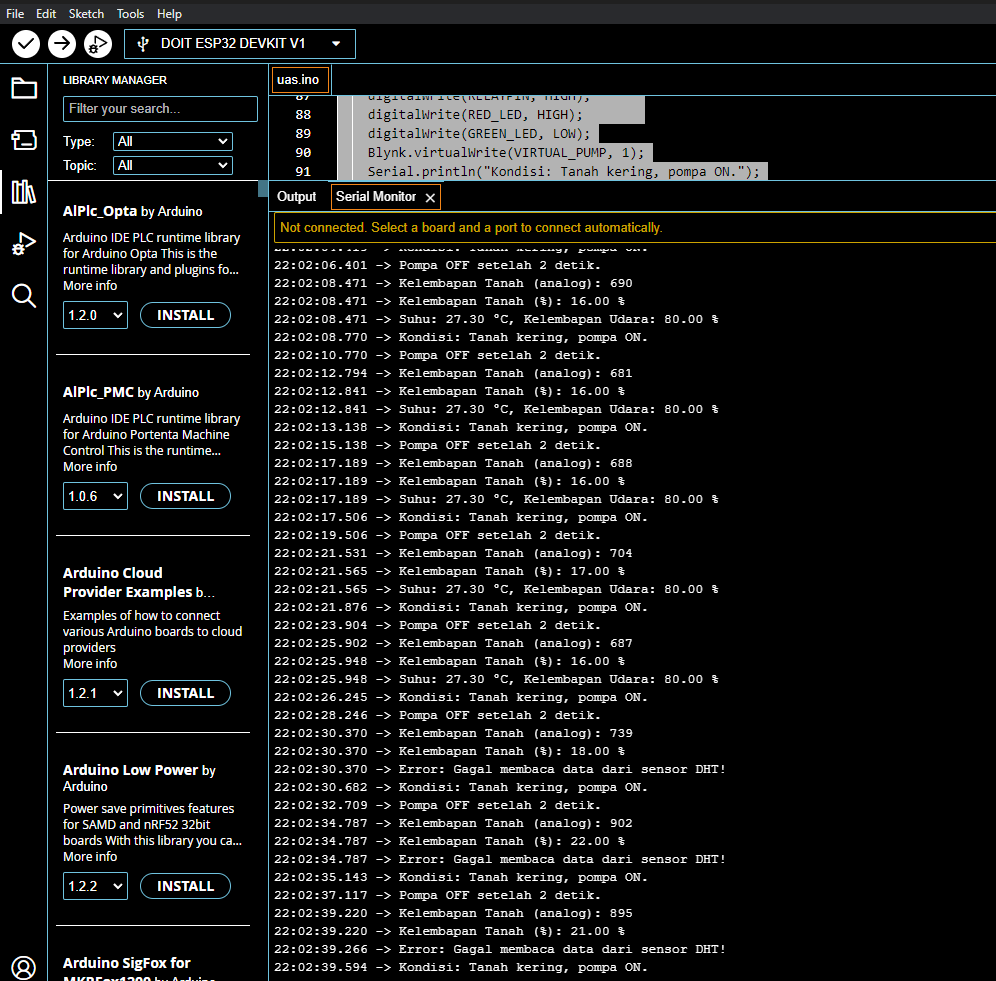
  delay(1000); // Tunggu 1 detik sebelum loop berikutnya

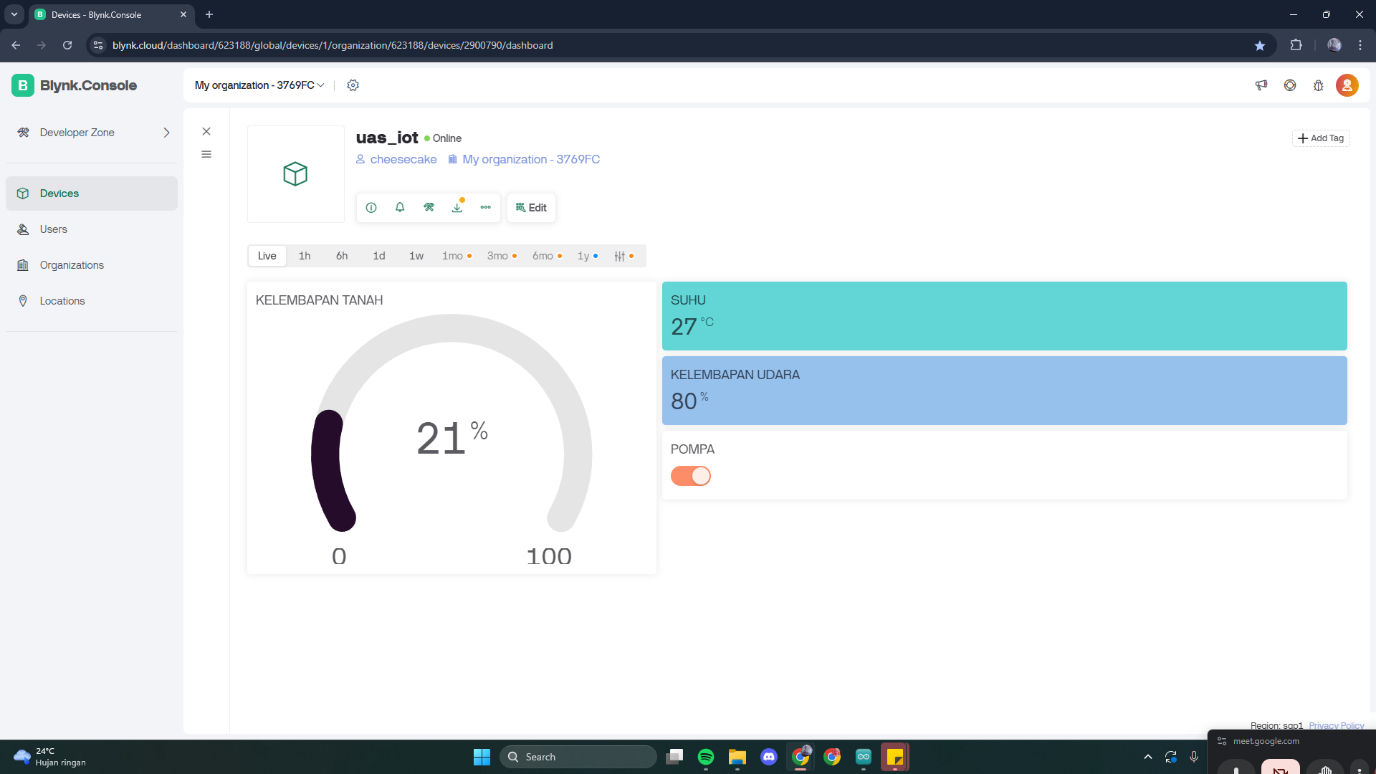
}

### Penjelasan

Program dimulai dengan mendeklarasikan konfigurasi Blynk, WiFi, pin sensor (DHT11 untuk suhu dan kelembapan udara, serta sensor kelembapan tanah), relay untuk pompa, dan LED indikator. Pada fungsi setup, sensor DHT diinisialisasi, pin-pin disiapkan, dan perangkat dihubungkan ke WiFi serta Blynk. Dalam loop utama, suhu, kelembapan udara, dan kelembapan tanah dibaca, lalu data dikirim ke Blynk melalui pin virtual. Logika kontrol menentukan tindakan berdasarkan nilai kelembapan tanah: jika tanah kering (di bawah DRY\_THRESHOLD), pompa diaktifkan selama 2 detik dan diatur tidak menyala lebih dari sekali dalam 24 jam. Jika tanah terlalu basah (di atas WET\_THRESHOLD), pompa dimatikan, dan LED hijau menyala. Untuk kondisi tanah cukup lembap, pompa tetap mati dan LED hijau menyala. Program ini menggunakan jeda 1 detik untuk iterasi berikutnya, memastikan pengoperasian yang efisien dan stabil.

## Hasil pengujian





## Link video

<https://drive.google.com/file/d/1DhGnaujEE89QaXERUr202n0Ih4aQuDRM/view?usp=drive_link>

# SISTEM KEAMANAN DATA

## Penjelasan algoritma Bcyript

Bcrypt adalah algoritma hashing yang dirancang untuk mengamankan kata sandi dengan cara yang adaptif dan aman. Algoritma ini dikembangkan oleh Niels Provos dan David Mazières, dan diperkenalkan pada tahun 1999. Bcrypt dibangun di atas cipher Blowfish dan menggunakan teknik yang disebut "salting" untuk melindungi data dari serangan seperti rainbow table[1][7].

**Karakteristik Utama Bcrypt**

1. Bcrypt menghasilkan nilai salt acak yang ditambahkan ke password sebelum proses hashing. Ini memastikan bahwa dua password identik tidak akan menghasilkan hash yang sama, sehingga meningkatkan keamanan[1][2].
2. Salah satu fitur utama dari bcrypt adalah kemampuannya untuk menyesuaikan jumlah iterasi (dikenal sebagai cost factor). Seiring dengan meningkatnya kekuatan komputasi, nilai cost dapat ditingkatkan untuk menjaga keamanan algoritma terhadap serangan brute-force[1][7].
3. Hasil dari bcrypt biasanya dimulai dengan prefix seperti "$2a$" atau "$2b$", diikuti oleh parameter cost, salt, dan hash password yang dihasilkan. Format ini memudahkan sistem untuk mengenali bahwa hash tersebut dihasilkan menggunakan algoritma bcrypt[4][5].

**Proses Algoritma Bcrypt**

Algoritma bcrypt terdiri dari beberapa langkah penting:

1. Inisialisasi: Membentuk array P sebanyak 18 elemen, masing-masing bernilai 32-bit.
2. Pengaturan Kunci: Menggunakan fungsi EksBlowfishSetup untuk menghasilkan kunci dari password dan salt.
3. Enkripsi: Melakukan enkripsi pada string "OrpheanBeholderScryDoubt" sebanyak 64 kali menggunakan mode ECB (Electronic Code Book) untuk menghasilkan hash akhir[1][3].
4. Penggabungan Hasil: Menggabungkan nilai cost, salt, dan ciphertext untuk membentuk hasil akhir dari hashing[2][6].

## Penjelasan kode

const bcrypt = require("bcryptjs"); // SKD Untuk hashing password

// Register Route

app.post(

  "/register",

  multer({ storage: diskStorage }).single("img"),

  (req, res) => {

    const { name, password, role, no\_whatsapp, foto\_profile } = req.body;

    const fileName = path.basename(req.file.path);

    // Hash password menggunakan bcrypt

    bcrypt.hash(password, 10, (err, hashedPassword) => {

      if (err) {

        console.log(err);

        return res.status(500).send("Error hashing password");

      }

      const query =

        "INSERT INTO user (name, password, role, no\_whatsapp, foto\_profile) VALUES (?, ?, ?, ?, ?)";

      db.query(

        query,

        [name, hashedPassword, role, no\_whatsapp, fileName],

        (err, result) => {

          if (err) {

            console.log(err);

            return res.status(500).send("Error creating user");

          }

          res.status(201).send("User registered successfully");

        }

      );

    });

  }

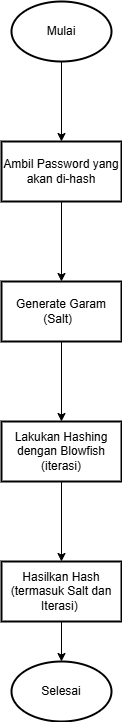
);

**bcrypt.hash(password, 10, callback)**:

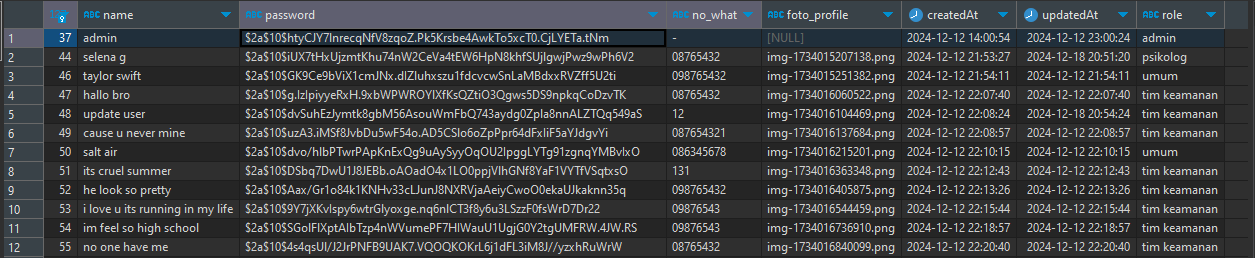
* **password**: Kata sandi asli yang akan di-hash.
* **10**: Nilai *salt rounds* (tingkat kompleksitas hashing). Semakin besar angkanya, semakin aman, tetapi juga memperlambat proses hashing.
* **callback**: Fungsi yang akan dipanggil setelah hashing selesai. Jika hashing berhasil, fungsi akan mengembalikan **hashedPassword**. Jika terjadi kesalahan, fungsi mengembalikan pesan error.

**Menggunakan hashed password** kata sandi yang telah di-hash disimpan ke database melalui query SQL, menggantikan kata sandi asli, sehingga lebih aman jika database diretas.

## Flowchart



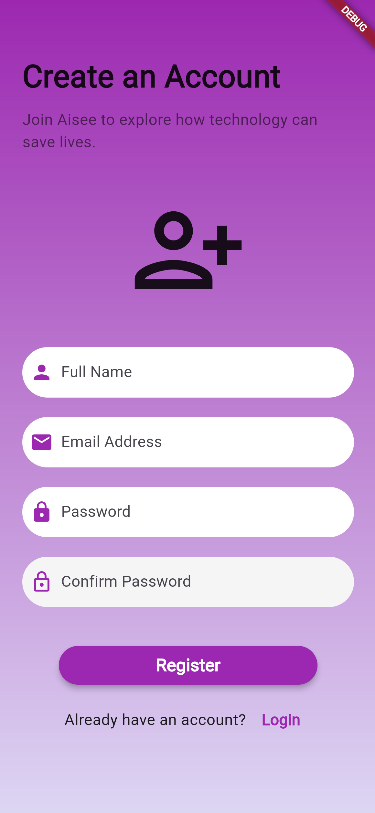
## Screenshot hasil



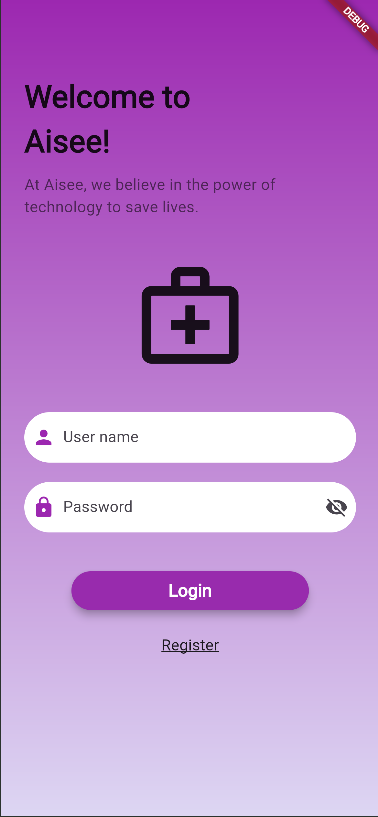
# MOBILE

## UI/UX Mobile

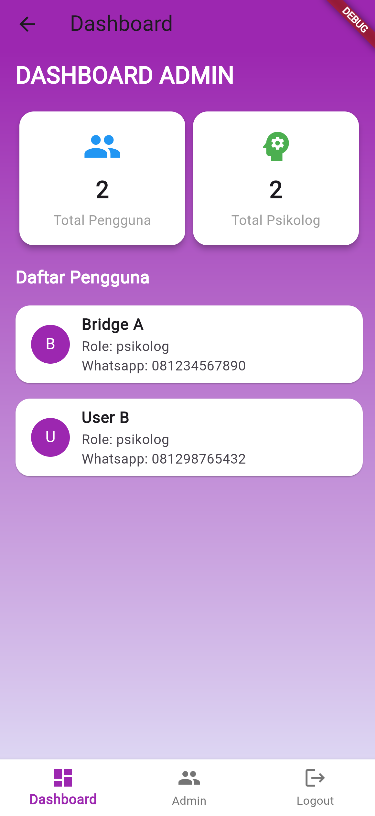
### Register



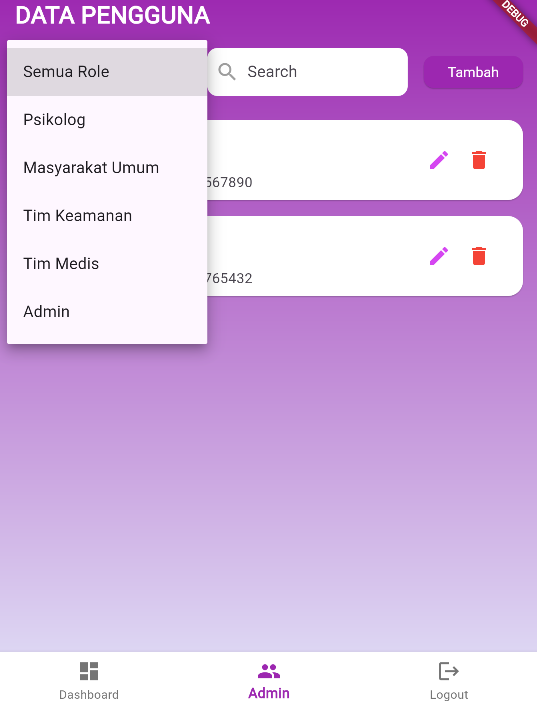
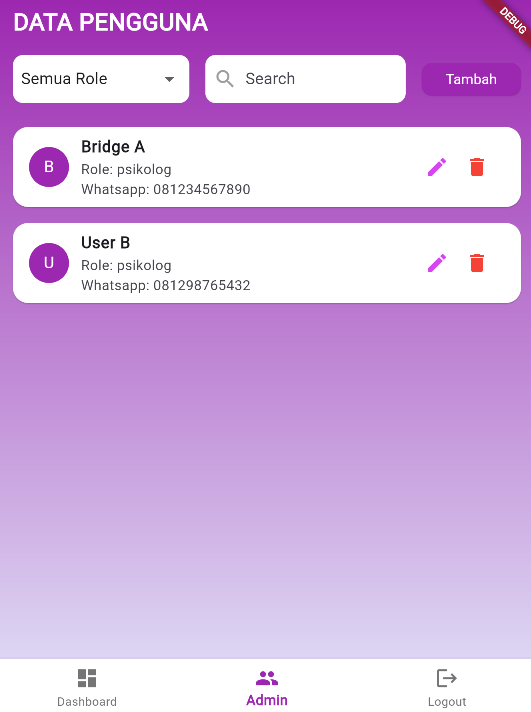
### Login

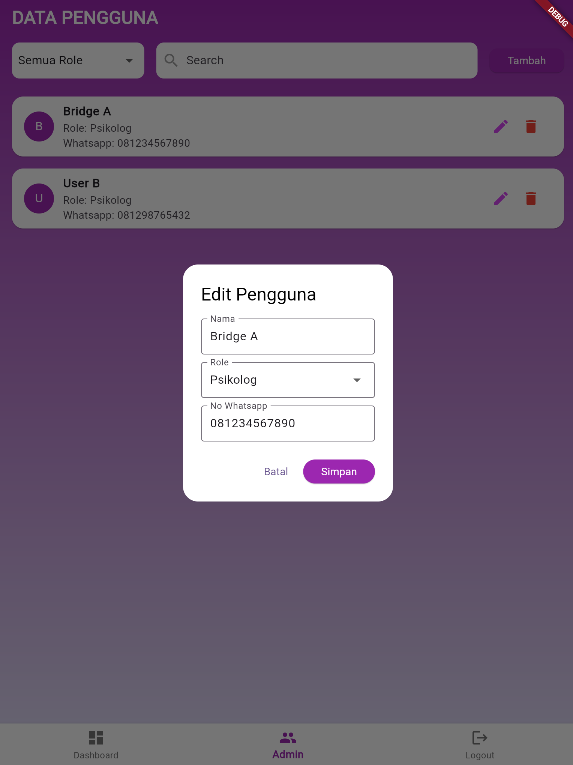
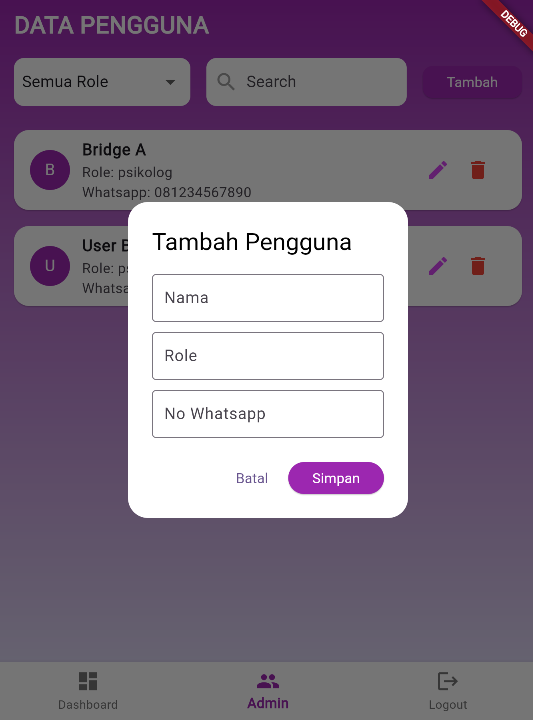


### Dasboard admin

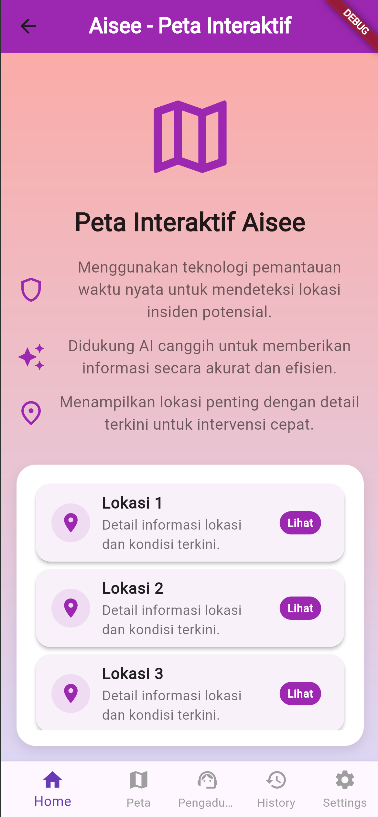


### Data pengguna

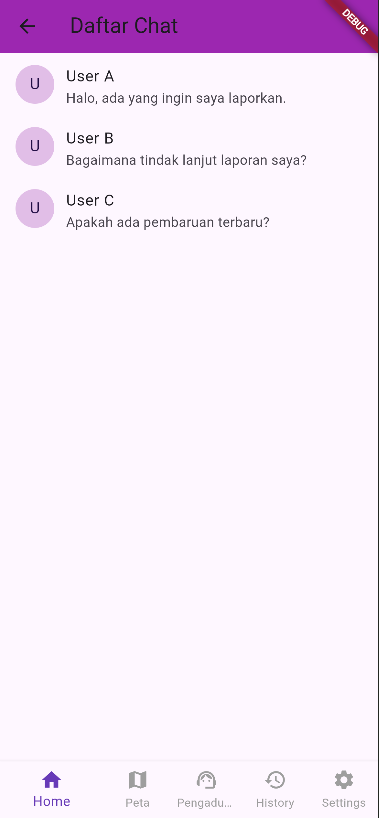




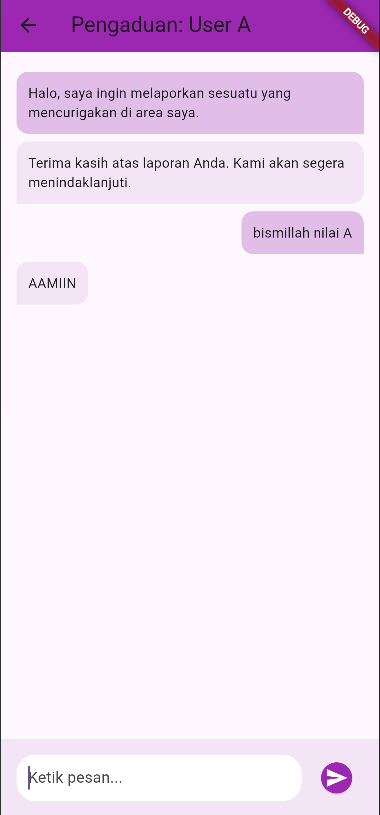
### Peta



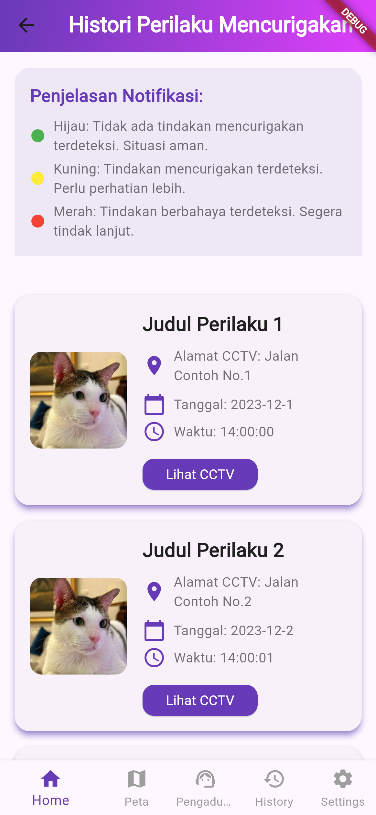
### Daftar chat



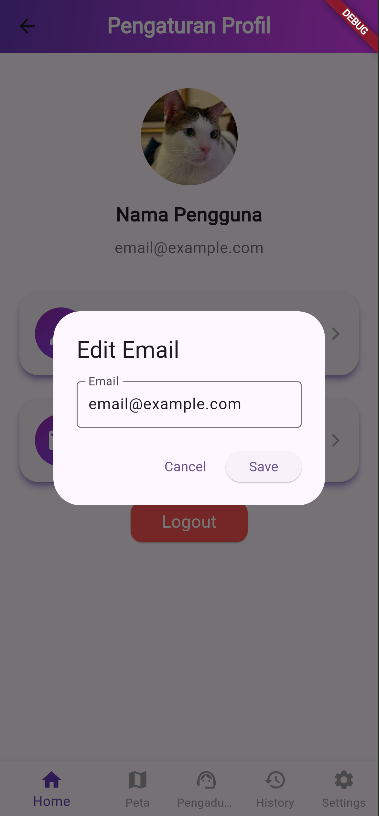
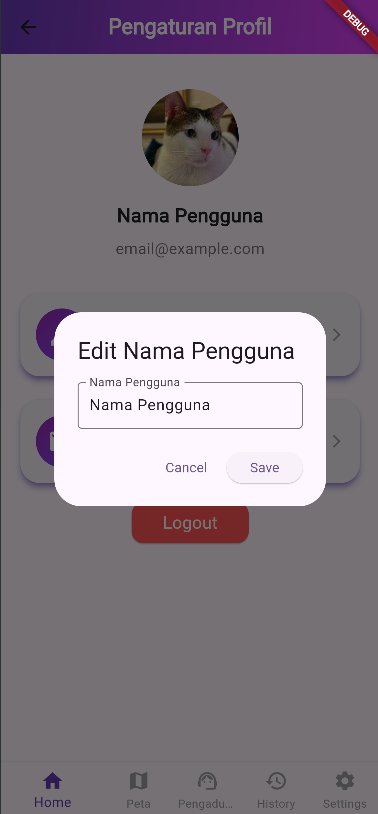
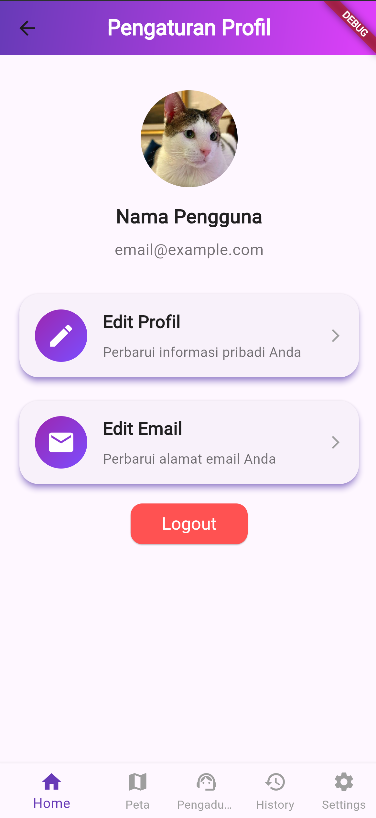
### Pengaduan dan laporan



### History



### Setting profil



## Penjelasan kode program

## Link apk

# Referensi

SKD:

[1] <https://id.wikipedia.org/wiki/Bcrypt>

[2] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Makalah-UAS/Makalah-UAS-Kripto-2020%20(16).pdf>

[3] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/Makalah2021/Makalah-Matdis-2021%20(48).pdf>

[4] <https://repository.nusamandiri.ac.id/repo/files/247952/download/JURNAL-06-DES_23.pdf>

[5] <http://lib.unnes.ac.id/37008/1/5302415013_Optimized.pdf>

[6] <https://www.researchgate.net/publication/379928628_PENGAMANAN_DATA_PASSWORD_PENGGUNA_MENGGUNAKAN_ALGORITMA_BCYRPT_PADA_SISTEM_PELAYANAN_KEUANGAN_DI_SUB_BAGIAN_KEUANGAN_DSDABM_KOTA_BANDUNG>

[7] <https://indodax.com/academy/bcrypt-cara-kerja-dan-keunggulan/>

[8] <https://ejurnal.stmik-budidarma.ac.id/index.php/JSON/article/download/4285/2796>