LAMPIRAN

Lampiran 1. Program Arduino IDE

```
#include <WiFi.h>  // Library wifi esp32
#include <Wire.h>
#include <DFRobot_SHT20.h> // Library sensor SHT20
#include <ESP32-HUB75-MatrixPanel-I2S-DMA.h> // Library
PAnel P5
#include "time.h"
#include <Firebase_ESP_Client.h>
#include "addons/RTDBHelper.h"
#include "addons/TokenHelper.h"


// --- WiFi ---
const char* ssid = "xxxx";
const char* password = "xxxx";

#define API_KEY
"AIzaSyAXQ5KprTtoQrYof_7CL8A5sY6_lgB78zQ"
#define DATABASE_URL "fir-iot-12265-default-
rtdb.firebaseio.com"

FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;

// --- NTP Server dan Timezone ---
const char* ntpServer = "pool.ntp.org";
const long gmtOffset_sec = 25200; // GMT+7 untuk
Jakarta, Indonesia (7 * 3600 detik)
const int daylightOffset_sec = 0; // Tidak ada daylight
saving

// --- Sensor Suhu & Kelembaban ---
DFRobot_SHT20 sht20(&Wire, SHT20_I2C_ADDR);

// --- MQ7 ---
#define MQ7_SENSOR_PIN 33    // Konfigurai pin analog
MQ-7
#define  MQ7_PWM_PIN 32      // pin PWM untuk
memanaskan MQ-7
const int MQ7_PWM_CHANNEL = 0;    // Channel PWM yang
digunakan untuk MQ-7
```

```
const int MQ7_PWM_FREQ = 5000;     // Frequensi PWM
const int MQ7_PWM_RES = 8;         // Resolusi PWM (8
bit = 0-255)
const int MQ7_HIGH_DUTY = 255;     // Nilai PWM untuk
high phase
const int MQ7_LOW_DUTY = 72;       // Nilai PWM untuk
low phase
const float MQ7_RL = 10.0;         // Nilai res. beban
(khm)
const float MQ7_VCC = 5.0;         // Teg. referensi MQ-
7 (karena sensor bekerja di 5V)
const float MQ7_Ro = 22.61;        // Nilai RO hasil
kalibrasi
const float MQ7_a = -0.45;       // Kemiringan kurva
logaritmik pada kadar CO
const float MQ7_b = 0.9;        // Intersep kurva
logarirmik pada kadar CO
const unsigned long MQ7_PREHEAT_TIME = 120000;  //
Durasi pemanasan awal 2 menit
const unsigned long MQ7_HIGH_TIME = 60000;      //
Durasi siklus highphase 1 mneit
const unsigned long MQ7_LOW_TIME = 60000;       //
Durasi siklus lowphase 1 menit
const int MQ7_SAMPLES = 10;              // rata"
sampling
unsigned long mq7_previousMillis = 0;    // Variabel
untuk menyimpan waktu sebelumnya
bool mq7_isHighPhase = true;             // Siklus
pemanasan tinggi
bool mq7_isPreheating = true;            // Siklus
pemanasan awal
float mq7_last_ppm = 0;                  // membaca data
hasil terakhir

// --- MQ135 ---
#define MQ135_PIN 34               // Konfigurasi pin
analog MQ-135
#define MQ135_RL 10000.0           // Nilai res. beban
#define VREF 3.3                   // Teg. referensi ADC
esp32
#define ADC_RES 4095.0             // Resolusi ADC esp32
12-bit (0-4095)
float MQ135_Ro = 157.77;           // Nilai RO pada
kalibrasi
const float MQ135_a = 2010.2743;  // Nilai regresi dari
kurva datasheet pada kadar CO2
const float MQ135_b = -1.139187;  // Nilai regresi dari
```

```
kurva datasheet pada kadar CO2
float mq135_last_ppm = 0;          // Variabel untuk
hasil ppm dari MQ-135
#define MQ135_PREHEAT_TIME 60000  // Durasi pemanasan
awal 1 menit
bool mq135_ready = false;          //
unsigned long mq135_start_time = 0; //

// --- MQ2 ---
#define MQ2_PIN 35                              //
Konfigurasi pin analog MQ-2
#define MQ2_RL 10000.0                          // Res.
beban pada MQ-2
const float MQ2_R0_CALIBRATED = 124475.73;  // Nilai RO
dari kalibrasi
const float MQ2_lpgSlope = -0.50;           // Nilai
default slope dari kalibrasi
const float MQ2_lpgIntercept = 1.45;        // Nilai
default intercept dari kalibrasi
float mq2_last_ppm = 0;                     // Variabel
untuk hasil ppm dari MQ-2
#define MQ2_PREHEAT_TIME 60000              // Durasi
pemanasan awal 1 menit
bool mq2_preheating = true;
unsigned long mq2_start_time = 0;

//tombol ganti wifi/tanpa wifi
#define tombol  25                                      //
Pin input saklar (pakai 3.3V jangan 5V)
// --- Relay ---
#define RELAY_PIN 23      // Konfigurasi pin relay

// --- LED Matrix Panel ---
#define R1_PIN 19     // Merah baris atas
#define G1_PIN 13     // Hijau baris atas
#define B1_PIN 18     // Biru baris atas
#define R2_PIN 5      // Merah baris bawah
#define G2_PIN 12     // Hijau baris bawah
#define B2_PIN 17     // Biru baris bawah
#define A_PIN 16      // Alamat bit baris 0
#define B_PIN 14      // Alamat bit baris 1
#define C_PIN 4       // Alamat bit baris 2
#define D_PIN 27      // Alamat bit baris 3
#define E_PIN -1      // Alamat bit baris 4, untuk
panel tertentu, jika tidak dipakai diisi -1
#define LAT_PIN 26    // Latch
#define OE_PIN 15     // Output Enable
```

```
#define CLK_PIN 2     // Clock
#define PANEL_RES_X 64  // Resolusi panel X
#define PANEL_RES_Y 32  // Resolusi panel Y
#define PANEL_CHAIN 2   // Jumlah panel yang dirangkai

unsigned long lastGoogleSend = 0;                //
const unsigned long googleSendInterval = 60000;

MatrixPanel_I2S_DMA *dma_display = nullptr;     //
Pointer objek panel LED
unsigned long lastSwitch = 0;                    //
Waktu terakhir ganti tampilan
bool ganti = true;                               //
Status tampilan berganti
const int switchInterval = 10000;                //
interval pergantian tampilan 10 detik
int xrelay;
int relay;

int displayMode = 0; // 0: data sensor, 1: tulisan, 2:
jam

// --- Fungsi untuk tampilan di panel LED ---
void tulisan() {
  dma_display->setTextColor(dma_display->color565(255,
255, 0));  // Set Warna kuning
  dma_display->setCursor(10, 1);                 // set
koordinat
  dma_display->print("TEKNIK ELEKTRONIKA");     //
Menampilkan Teks
  dma_display->setCursor(2, 8);
  dma_display->print("--------------------");  //
Menampilkan garis pemisah
  dma_display->setCursor(34, 14);
  dma_display->print("UNIVERSITAS");
  dma_display->setCursor(28, 23);
  dma_display->print("HARKAT NEGERI");
}

// --- Fungsi untuk menampilkan jam di panel LED ---
void displayClock() {
  struct tm timeinfo;
  if (!getLocalTime(&timeinfo)) {
    Serial.println("Gagal mendapatkan waktu");
    return;
  }
```

```
  // Format waktu menjadi string "HH:MM:SS"
  char timeStr[9];
  strftime(timeStr, sizeof(timeStr), "%H:%M:%S",
&timeinfo);

  dma_display->fillScreen(0);
  dma_display->setTextSize(1);
  dma_display->setTextColor(dma_display->color565(255,
255, 0)); // Warna kuning
  dma_display->setCursor(35, 1); // Atur posisi
tampilan tengah
  dma_display->print("PEMANTAUAN"); // Tampilkan NAMA
ALAT
  dma_display->setCursor(23, 10);
  dma_display->print("KUALITAS UDARA"); // Tampilkan
NAMA ALAT
  dma_display->setCursor(2, 17);
  dma_display->print(" ------------------- ");
  dma_display->setCursor(37, 23);
  dma_display->print(timeStr);

}

// --- Fungsi untuk membaca semua sensor ---
void datautama(){
  unsigned long now = millis();

  float temp = sht20.readTemperature();
  float hum = sht20.readHumidity();

  if (!mq135_ready && now - mq135_start_time >=
MQ135_PREHEAT_TIME) {
    MQ135_Ro = getMQ135Resistance() / 2.29;
    mq135_ready = true;
    Serial.println("Kalibrasi MQ135 selesai.");
  }

  if (mq7_isPreheating && now - mq7_previousMillis >=
MQ7_PREHEAT_TIME) {
    mq7_isPreheating = false;
    mq7_previousMillis = now;
    ledcWrite(MQ7_PWM_CHANNEL, MQ7_HIGH_DUTY);
  }

  if (!mq7_isPreheating) {
    if (mq7_isHighPhase && now - mq7_previousMillis >=
MQ7_HIGH_TIME) {
```

```
      mq7_isHighPhase = false; mq7_previousMillis =
now;
      ledcWrite(MQ7_PWM_CHANNEL, MQ7_LOW_DUTY);
    } else if (!mq7_isHighPhase && now -
mq7_previousMillis >= MQ7_LOW_TIME) {
      mq7_isHighPhase = true; mq7_previousMillis = now;
      ledcWrite(MQ7_PWM_CHANNEL, MQ7_HIGH_DUTY);
    }
    if (!mq7_isHighPhase) bacaMQ7();
  }

  if (mq135_ready && mq7_isHighPhase) bacaMQ135();

  if (now - mq2_start_time >= MQ2_PREHEAT_TIME) {
    if (mq2_preheating) {
      Serial.println("MQ2 preheat selesai.");
      mq2_preheating = false;
    }
    bacaMQ2();
  } else {
    Serial.println("MQ2 sedang preheat...");
  }

  Serial.printf("CO: %.2f ppm | CO2: %.2f ppm | LPG:
%.2f ppm\n", mq7_last_ppm, mq135_last_ppm,
mq2_last_ppm);
  Serial.printf("Temp: %.2f C | Humidity: %.2f %%\n",
temp, hum);

if (Firebase.ready()) {
    if (Firebase.RTDB.setFloat(&fbdo, "PPM/HK",
mq2_last_ppm)) {
      Serial.print("HK: ");
      Serial.println(mq2_last_ppm);
    } else {
      Serial.println("FAILED HK");
    }
    if (Firebase.RTDB.setFloat(&fbdo, "PPM/CO2",
mq135_last_ppm)) {
      Serial.print("CO2: ");
      Serial.println(mq135_last_ppm);
    } else {
      Serial.println("FAILED CO2");
    }
    if (Firebase.RTDB.setFloat(&fbdo, "PPM/CO",
mq7_last_ppm)) {
      Serial.print("CO: ");
```

```
      Serial.println(mq7_last_ppm);
    } else {
      Serial.println("FAILED CO");
    }
    if (Firebase.RTDB.setFloat(&fbdo, "PPM/Temp",
temp)) {
      Serial.print("Temp: ");
      Serial.println(temp);
    } else {
      Serial.println("FAILED Temp");
    }
    if (Firebase.RTDB.setFloat(&fbdo, "PPM/Hum", hum))
{
      Serial.print("Hum: ");
      Serial.println(hum);
    } else {
      Serial.println("FAILED Hum");
    }

  if (Firebase.ready()) {
  if (Firebase.RTDB.getInt(&fbdo, "PPM/relay", &relay))
{
    Serial.printf("Get int ref... %s\n",
String(relay).c_str());

    if (relay == 1) {
      digitalWrite(RELAY_PIN, HIGH); // pastikan ini
pakai huruf besar
      Serial.println("Relay MENYALA");
    } else {
      digitalWrite(RELAY_PIN, LOW);
      Serial.println("Relay MATI");
    }

    xrelay = relay;

  } else {
    Serial.print("Firebase error: ");
    Serial.println(fbdo.errorReason());
  }
} else {
  Serial.println("Firebase not ready.");
}

  sendToDisplay(temp, hum, mq7_last_ppm, mq2_last_ppm,
mq135_last_ppm);
}
```

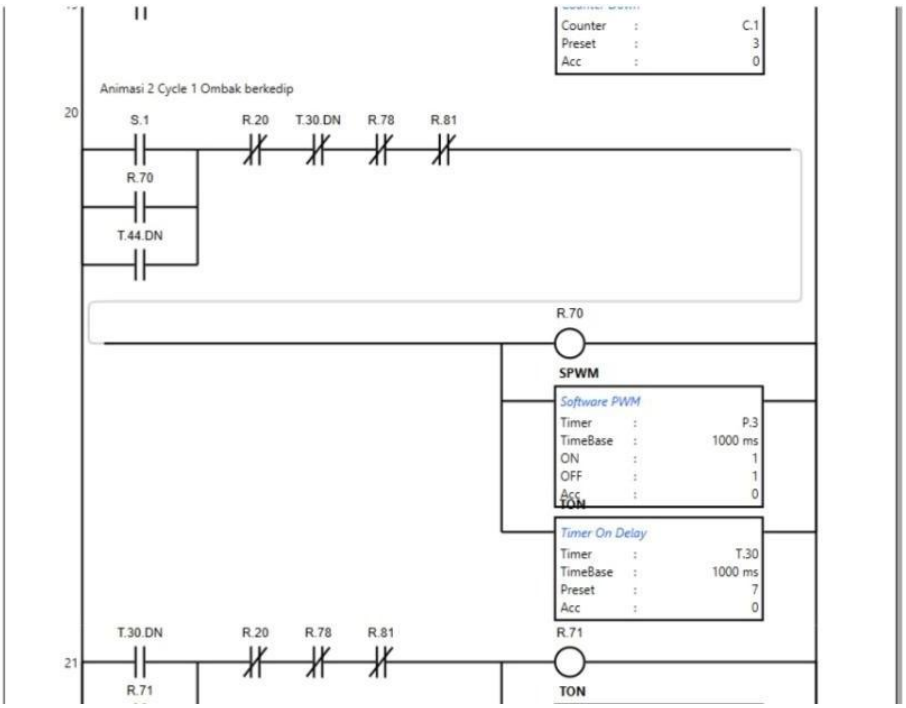
```
}

// --- Display Function (Gabungan) ---
void sendToDisplay(float temp, float hum, float
last_ppm7, float last_ppm2, float last_ppm135) {
 unsigned long now = millis();

  if (now - lastSwitch < switchInterval) {
    // Perbarui tampilan jam setiap detik, jika mode
tampilan adalah jam.
  if (displayMode == 2) {
      displayClock();
    }
    return;
  }
 lastSwitch = now;

 dma_display->fillScreen(0);
 dma_display->setTextSize(1);

 displayMode = (displayMode + 1) % 3;

  if (displayMode == 0) {
    // Tampilan 1: Data Sensor
    dma_display->setTextColor(dma_display-
>color565(255, 0, 255));
    dma_display->setCursor(2, 0);    dma_display-
>print("CO :"); dma_display->print(last_ppm7);
    dma_display->setCursor(78, 0);   dma_display-
>print("ppm");
    dma_display->setCursor(2, 8);    dma_display-
>print("HC :"); dma_display->print(last_ppm2);
    dma_display->setCursor(78, 8);   dma_display-
>print("ppm");
    dma_display->setCursor(2, 16);   dma_display-
>print("CO2:"); dma_display->print(last_ppm135);
    dma_display->setCursor(78, 16);  dma_display-
>print("ppm");
    dma_display->setTextColor(dma_display-
>color565(255, 0, 0));
    dma_display->setCursor(10, 24);  dma_display-
>print("T");
    dma_display->setTextColor(dma_display-
>color565(255, 0, 255));
    dma_display->setCursor(8, 24);   dma_display-
>print("  :"); dma_display->print(temp, 1);
    dma_display->setCursor(52, 24);  dma_display-
```

```
>print("C");
    dma_display->setTextColor(dma_display->color565(0,
0, 255));
    dma_display->setCursor(68, 24);  dma_display-
>print("H");
    dma_display->setTextColor(dma_display-
>color565(255, 0, 255));
    dma_display->setCursor(61, 24);  dma_display-
>print("  :");
    dma_display->setCursor(78, 24); dma_display-
>print(hum, 1);
    dma_display->setCursor(103, 24); dma_display-
>print("%");
  } else if (displayMode == 1) {
    // Tampilan 2: Tulisan
    tulisan();
  } else if (displayMode == 2) {
    // Tampilan 3: Jam
    displayClock();
  }
}

// --- Sensor Functions ---
void bacaMQ7() {
 float total = 0;
 for (int i = 0; i < MQ7_SAMPLES; i++) {
  total += analogRead(MQ7_SENSOR_PIN);
  delay(50);
 }
 float adc = total / MQ7_SAMPLES;
 float Vout = (adc / ADC_RES) * MQ7_VCC;
 float Rs = (MQ7_VCC - Vout) * MQ7_RL / Vout;
 float rs_ro = Rs / MQ7_Ro;
 mq7_last_ppm = pow(10, MQ7_a * log10(rs_ro) + MQ7_b);
}

float getMQ135Resistance() {
 int adc = analogRead(MQ135_PIN);
 float volt = (adc / ADC_RES) * VREF;
 return (VREF - volt) * MQ135_RL / volt;
}

void bacaMQ135() {
 float rs = getMQ135Resistance();
 float rs_ro = rs / MQ135_Ro;
 mq135_last_ppm = pow(rs_ro / MQ135_a, 1.0 / MQ135_b);
}
```

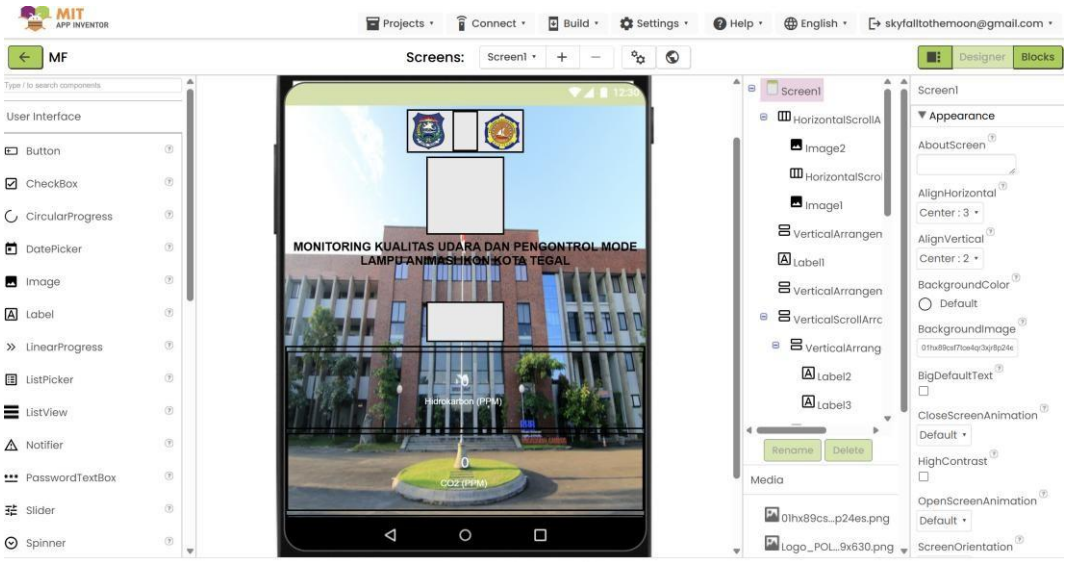
```
float readFilteredADC(int pin) {
 const int samples = 10;
 float total = 0;
 for (int i = 0; i < samples; i++) {
  total += analogRead(pin);
  delay(10);
 }
 return total / samples;
}

void bacaMQ2() {
 float adc = readFilteredADC(MQ2_PIN);
 float volt = (adc / ADC_RES) * VREF;
 float Rs = (VREF - volt) * MQ2_RL / volt;
 float rs_ro = Rs / MQ2_R0_CALIBRATED;
 mq2_last_ppm = pow(10, MQ2_lpgSlope * log10(rs_ro) +
MQ2_lpgIntercept);
}

// --- Setup ---
void setup() {
 Serial.begin(115200);
 Wire.begin();
 sht20.initSHT20();
 delay(100);
 Serial.println("Sensor init finish!");
 pinMode(RELAY_PIN, OUTPUT);
 pinMode(tombol,INPUT);
 digitalWrite(RELAY_PIN, HIGH);

 int datatombol = digitalRead(tombol);
  if(datatombol == 1){
    WiFi.begin(ssid, password);
    Serial.print("Menghubungkan ke WiFi");
    while (WiFi.status() != WL_CONNECTED) {
      delay(500); Serial.print(".");
    }
    Serial.println("\nWiFi connected!");

    config.api_key = API_KEY;
    config.database_url = DATABASE_URL;
    if (Firebase.signUp(&config, &auth, "", "")) {
      Serial.println("Firebase OK\nFree Palestine");
    } else {
      Serial.printf("%s\n",
config.signer.signupError.message.c_str());
```

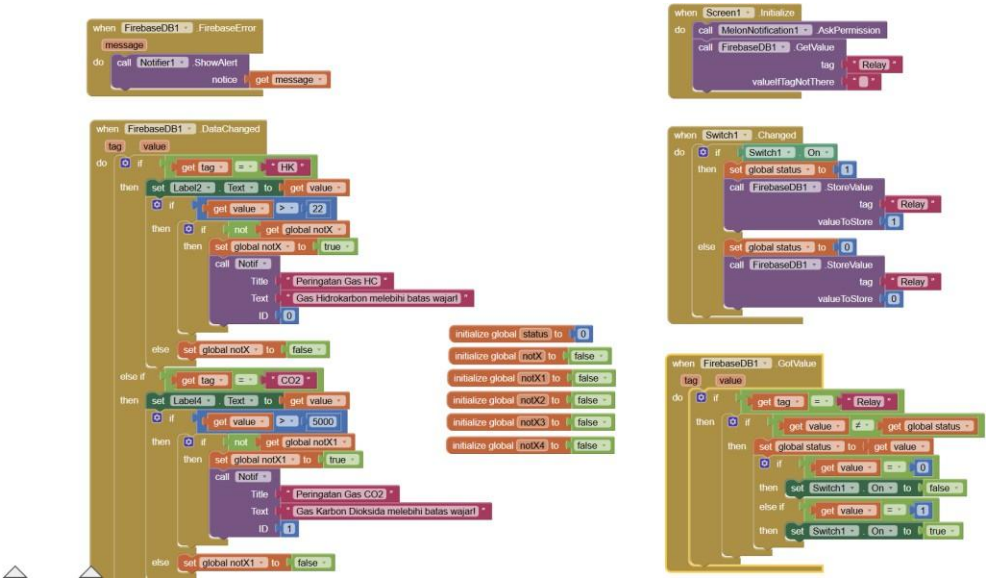
```
    }
    config.token_status_callback = tokenStatusCallback;
    // Koneksi Firebase
    Firebase.begin(&config, &auth);
    Firebase.reconnectWiFi(true);

    // Konfigurasi NTP untuk sinkronisasi waktu
    configTime(gmtOffset_sec, daylightOffset_sec,
ntpServer);
    Serial.println("Waktu disinkronkan dari NTP
server.");
  } else {
    Serial.println("Running tanpa WiFi & layanan
eksternal.");
  }

 ledcSetup(MQ7_PWM_CHANNEL, MQ7_PWM_FREQ, MQ7_PWM_RES);
 ledcAttachPin(MQ7_PWM_PIN, MQ7_PWM_CHANNEL);
 ledcWrite(MQ7_PWM_CHANNEL, MQ7_HIGH_DUTY);
 mq7_previousMillis = millis();

 mq2_start_time = millis();
 mq135_start_time = millis();

 HUB75_I2S_CFG::i2s_pins _pins =
{R1_PIN,G1_PIN,B1_PIN,R2_PIN,G2_PIN,B2_PIN,A_PIN,B_PIN,
C_PIN,D_PIN,E_PIN,LAT_PIN,OE_PIN,CLK_PIN};
 HUB75_I2S_CFG mxcfg(PANEL_RES_X, PANEL_RES_Y,
PANEL_CHAIN, _pins);
 mxcfg.i2sspeed = HUB75_I2S_CFG::HZ_10M;

 dma_display = new MatrixPanel_I2S_DMA(mxcfg);
 dma_display->begin();
 dma_display->setBrightness8(65);
}

// --- Loop ---
void loop() {
  datautama();
  delay(1000);
}ss
```

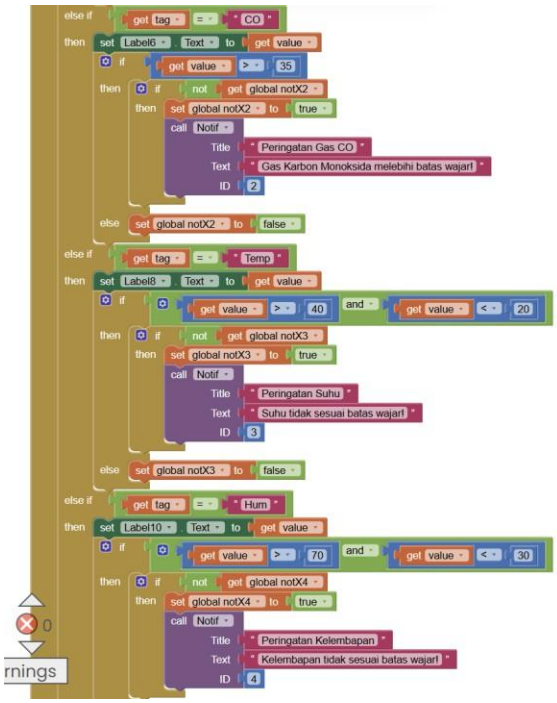
Lampiran 2. Ledder Diagram

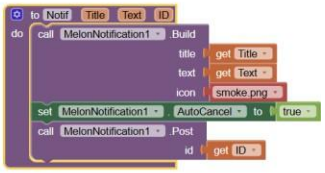


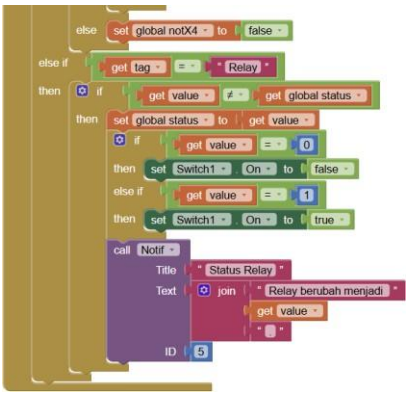
Lampiran 3. App Inventor






else if get tag = " CO "
then set Label6 . Text to get value
if get value > 35
then if not get global notX2
then set global notX2 to true
call Notif
Title " Peringatan Gas CO "
Text " Gas Karbon Monoksida melebihi batas wajar! "
ID 2
else set global notX2 to false
else if get tag = " Temp "
then set Label8 . Text to get value
if get value > 40 and get value < 20
then if not get global notX3
then set global notX3 to true
call Notif
Title " Peringatan Suhu "
Text " Suhu tidak sesuai batas wajar! "
ID 3
else set global notX3 to false
else if get tag = " Hum "
then set Label10 . Text to get value
if get value > 70 and get value < 30
then if not get global notX4
then set global notX4 to true
call Notif
Title " Peringatan Kelembapan "
Text " Kelembapan tidak sesuai batas wajar! "
ID 4
0
rnings


to Notif Title Text ID
do call MelonNotification1 .Build
title get Title
text get Text
icon smoke.png
set MelonNotification1 . AutoCancel to true
call MelonNotification1 .Post
id get ID


else set global notX4 to false
else if get tag = " Relay "
then if get value ≠ get global status
then set global status to get value
if get value = 0
then set Switch1 . On to false
else if get value = 1
then set Switch1 . On to true
call Notif
Title " Status Relay "
Text join " Relay berubah menjadi "
get value
" "
ID 5

Lampiran 4. Penilaian Bimbingan Tugas Akhir Individu

PENILAIAN BIMBINGAN TUGAS AKHIR INDIVIDU

Judul : Sistem Monitoring Kualitas Udara dn Pengontrol Lampu Hias Ikonik Kota Tegal Menggunakan Teknologi Internet Of Things

Nama : Muhammad Faruq Elhaq

NIM : 22010014

Kelas : 6A / D3 Teknik Elektronika

I. Nilai Bimbingan Tugas Akhir (Pembimbing I)

No	Unsur Yang Dinilai	Nilai
1	Kedisiplinan dalam bimbingan	90
2	Kreativitas pemecahan dalam bimbingan	85
3	Penguasaan materi tugas akhir	90
4	Kelengkapan dan referensi tugas akhir	85
Total Nilai = (Jumlah Nilai / 4)		87,5

II. Nilai Bimbingan Tugas Akhir (Pembimbing II)

No	Unsur Yang Dinilai	Nilai
1	Kedisiplinan dalam bimbingan	90
2	Kreativitas pemecahan dalam bimbingan	85
3	Penguasaan materi tugas akhir	85
4	Kelengkapan dan referensi tugas akhir	90
Total Nilai = (Jumlah Nilai / 4)		87,5

Nilai Bimbingan $= \frac{TotalNilaiPembimbing1 + TotalNilaiPembibing2}{2}$

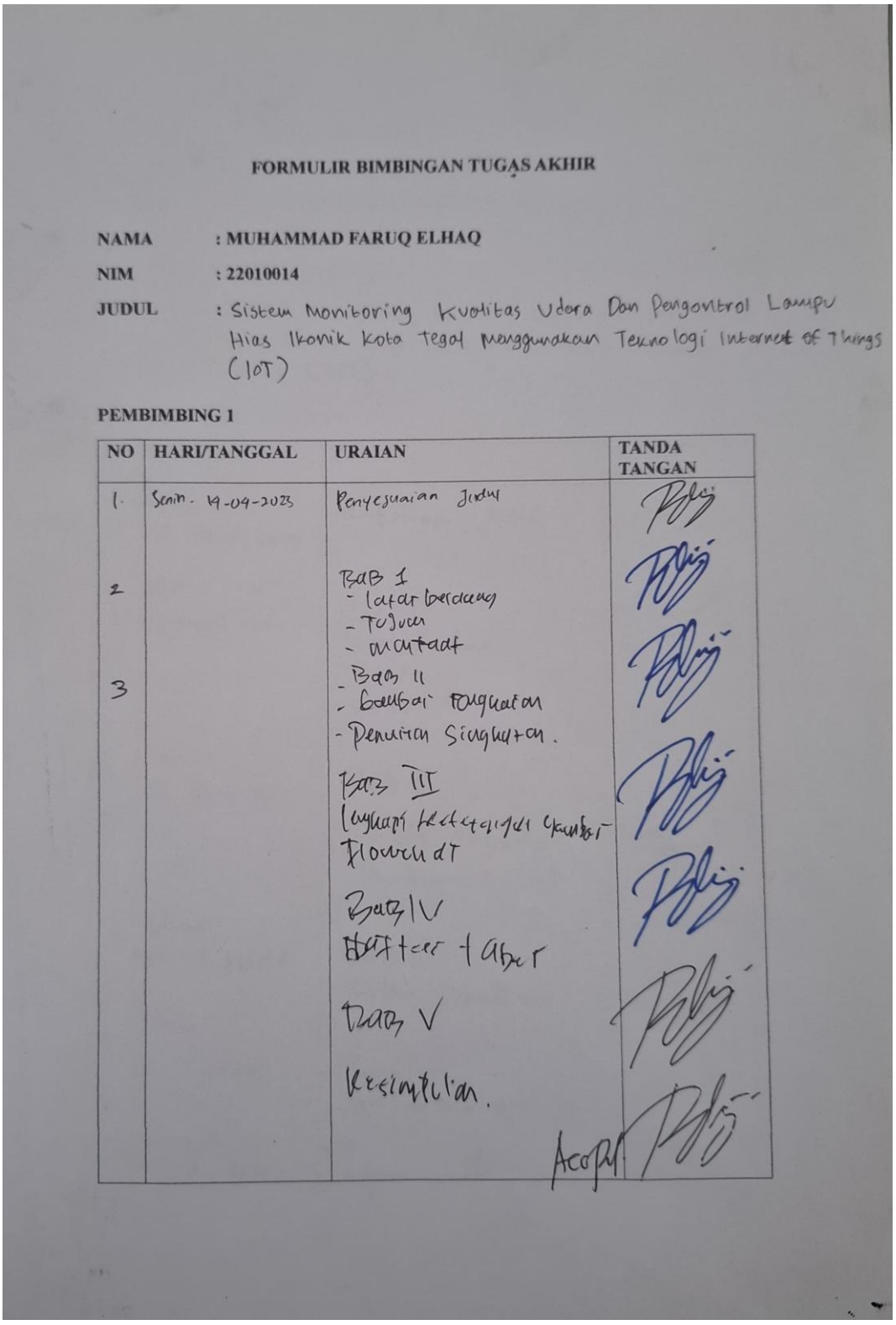
$= \frac{87,5 + 87,5}{2} = 87,5$

Tegal, Juli 2025

Mengetahui,

Pembimbing 1	Pembimbing 2
Rony Darpono M.T	Bahrun Niam M.T

Lampiran 5. Formulir Bimbingan Tugas Akhir



FORMULIR BIMBINGAN TUGAS AKHIR

NAMA : **MUHAMMAD FARUQ ELHAQ**

NIM : **22010014**

JUDUL : Sistem Monitoring Kualitas Udara Dan Pengontrol Lampu Hias Ikonik Kota Tegal menggunakan Teknologi Internet of Things (IoT)

PEMBIMBING 1

NO	HARI/TANGGAL	URAIAN	TANDA TANGAN
1.	Senin - 17-04-2023	Penyesuaian Judul	
2		BAB I - Latar Belakang - Tujuan - Manfaat	
3		- BAB II - Gambar Rangkaian - Penulisan Singkatan.	
		BAB III Lengkapi Keterangan Gambar Flowchart	
		BAB IV Hasil dan Pembahasan	
		BAB V	
		Kesimpulan.	
		Acc	

FORMULIR BIMBINGAN TUGAS AKHIR

NAMA : **MUHAMMAD FARUQ ELHAQ**

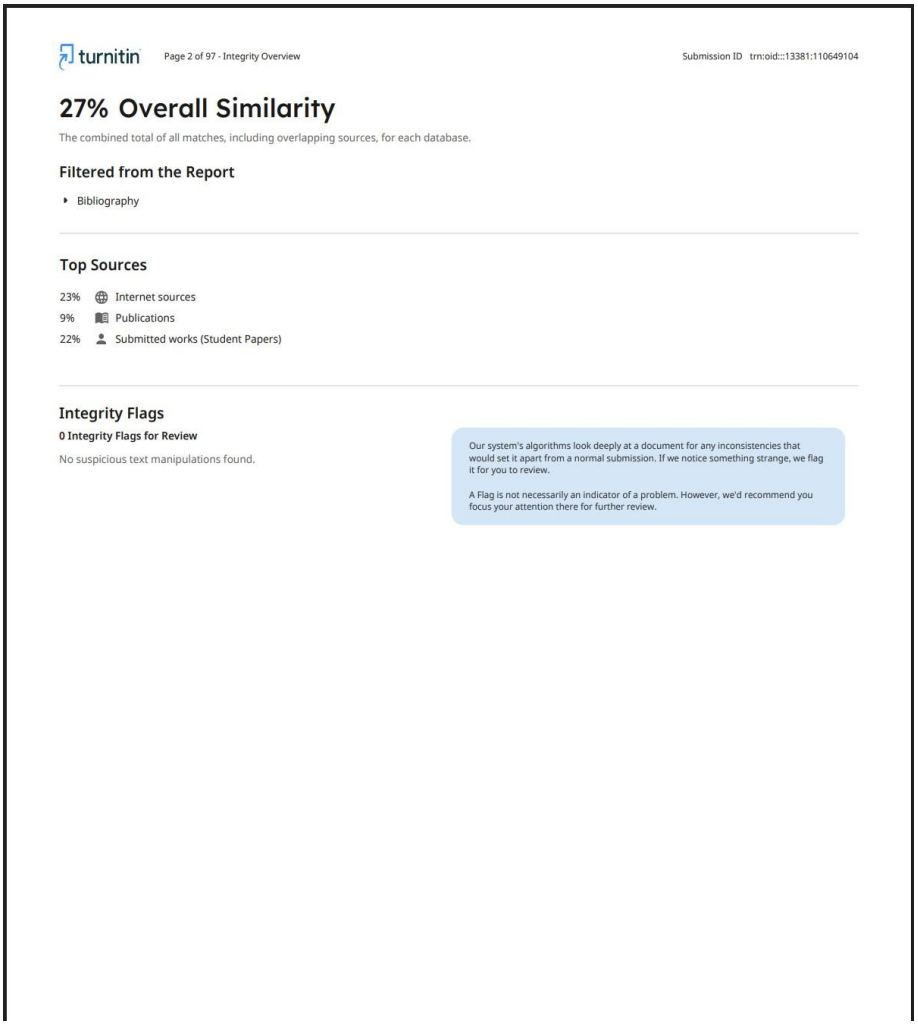
NIM : **22010014**

JUDUL : Sistem Monitoring kualitas Udara Dan pengontrol Lampu Hias Ikonik kota Tegal Menggunakan Teknologi Internet of Things (IoT)

PEMBIMBING 2

NO	HARI/TANGGAL	URAIAN	TANDA TANGAN
1	Senin 14 March 2025	Pembahasan Judul.	
2.	Kamis 17 March 2025	Bab I. - Latar belakang. - Tujuan - Manfaat	
3.	Senin 28 April 2025	Bab II - Gambar Rangkaian - Penulisan sintaks.	
4.	Selasa 8 - 7 - 2025	BAB III - sistem penulisan.	
5	Rabu 16 - 7 - 2025	Bab IV - Tambahkan waktu.	
6.	Jumat 18 - 7 - 2025	Laporan Ace.	

Lampiran 6. *Originally Report by Turnitin*



turnitin Page 2 of 97 - Integrity Overview Submission ID trn:oid:::13381:110649104

27% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Top Sources

23% Internet sources
9% Publications
22% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Lampiran 7. Dokumentasi