Lampiran 3 Studi Kasus

Lampiran 4 Source Code : Arduino IDE

```
#include <CTBot.h>
#include <CTBotDataStructures.h>
#include <CTBotDefines.h>
#include <CTBotInlineKeyboard.h>
#include <CTBotReplyKeyboard.h>
#include <CTBotSecureConnection.h>
#include <CTBotStatusPin.h>
#include <CTBotWifiSetup.h>
#include <Utilities.h>
#include <ArduinoJson.h>
#include <ArduinoJson.hpp>
#define BLYNK_TEMPLATE_ID "TMPL60I-ohLn1"
#define BLYNK_TEMPLATE_NAME "Air Quality Monitoring"
#define BLYNK_PRINT Serial
#include <Blynk.h>
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <TimeLib.h>
#include <WidgetRTC.h>

// Token dan informasi Blynk dan Telegram
char auth[] = "MxhTQizpPKLN9nEQMtJBx-EID8tHRlEg";
const char* ssid = "satriyo";
const char* password = "suryo123";
char botToken[] = "7351106812:AAFOX7wzD2GPOYqqelMqfosEKcBNaQH-sqw";
char chatId[] = "6312067345";

// Inisialisasi Blynk dan Telegram
WiFiClientSecure client;
UniversalTelegramBot bot(botToken, client);
BlynkTimer timer;
WidgetRTC rtc;

// Pin sensor dan aktuator
#define MQ2 D3
#define MQ7 A0
#define buzz D1
#define ledKuning D6
#define ledMerah D1
```

```
float mq2_ppm = 0;
float mq7_ppm = 0;
double sedikit, sedang, banyak;
double tipis, tengah, tebal;

// Variabel kontrol on/off
bool alatAktif = true;

// Fungsi untuk mendapatkan tanggal dan waktu saat ini
String getCurrentDateTime() {
  char currentDateTime[20];
  snprintf(currentDateTime, sizeof(currentDateTime), "%04d-%02d-%02d %02d:%02d:%02d",
year(), month(), day(), hour(), minute(), second());
  return String(currentDateTime);
}

// Fungsi untuk kalibrasi sensor
void calibrateSensors() {

}

// Fungsi untuk membaca sensor
void readSensors() {
  Serial.println("Membaca sensor...");
  if (alatAktif) {
    Serial.println("Kontrol aktif.");

    // Membaca nilai analog
    mq2_raw = analogRead(MQ2);
    mq7_raw = analogRead(MQ7);

    // Kalibrasi sensor jika perlu
    calibrateSensors();


    mq2_ppm = map(mq2_raw, 0, 1023, 0, 500);
    mq7_ppm = map(mq7_raw, 0, 1023, 0, 600);

    Serial.print("MQ2 PPM: ");
    Serial.println(mq2_ppm);
    Serial.print("MQ7 PPM: ");
    Serial.println(mq7_ppm);

    // Fuzzifikasi
    // MQ2 - Sedikit (Segitiga)
    if (mq2_ppm <= 0 || mq2_ppm > 250) sedikit = 0;
```

```
else if (mq2_ppm <= 150) sedikit = 1;
else sedikit = (250 - mq2_ppm) / (250 - 150);

// MQ2 - Sedang (Segitiga)
if (mq2_ppm <= 150 || mq2_ppm > 350) sedang = 0;
else if (mq2_ppm <= 250) sedang = (mq2_ppm - 150) / (250 - 150);
else sedang = (350 - mq2_ppm) / (350 - 250);

// MQ2 - Banyak (Segitiga)
if (mq2_ppm <= 250 || mq2_ppm > 500) banyak = 0;
else if (mq2_ppm <= 350) banyak = (mq2_ppm - 250) / (350 - 250);
else banyak = 1;

// MQ7 - Tipis (Segitiga)
if (mq7_ppm <= 0 || mq7_ppm > 300) tipis = 0;
else if (mq7_ppm <= 100) tipis = 1;
else tipis = (300 - mq7_ppm) / (300 - 100);

// MQ7 - Tengah (Segitiga)
if (mq7_ppm <= 100 || mq7_ppm > 500) tengah = 0;
else if (mq7_ppm <= 300) tengah = (mq7_ppm - 100) / (300 - 100);
else tengah = (500 - mq7_ppm) / (500 - 300);

// MQ7 - Tebal (Segitiga)
if (mq7_ppm <= 300 || mq7_ppm > 600) tebal = 0;
else if (mq7_ppm <= 450) tebal = (mq7_ppm - 300) / (450 - 300);
else tebal = 1;

// Implementasi aturan fuzzy
float a1 = min(sedikit, tipis);
float z1 = a1 * 150;
float a2 = min(sedang, tipis);
float z2 = a2 * 150;
float a3 = min(banyak, tipis);
float z3 = 150 + a3 * 150;
float a4 = min(sedikit, tengah);
float z4 = a4 * 150;
float a5 = min(sedang, tengah);
float z5 = 150 + a5 * 150;
float a6 = min(banyak, tengah);
float z6 = 150 + a6 * 150;
float a7 = min(sedikit, tebal);
float z7 = 150 + a7 * 150;
float a8 = min(sedang, tebal);
float z8 = 150 + a8 * 150;
float a9 = min(banyak, tebal);
float z9 = 150 + a9 * 150;
```

```
    float hasil_a = (a1 * z1) + (a2 * z2) + (a3 * z3) + (a4 * z4) + (a5 * z5) + (a6 * z6) + (a7 * z7)
+ (a8 * z8) + (a9 * z9);
    float hasil_z = a1 + a2 + a3 + a4 + a5 + a6 + a7 + a8 + a9;

    // Menghindari pembagian dengan nol
    float hasil_akhir = (hasil_z > 0) ? hasil_a / hasil_z : 0;

    Serial.print("Hasil Akhir: ");
    Serial.println(hasil_akhir);

    // Kendali aktuator
    if (hasil_akhir <= 150) {
      digitalWrite(ledKuning, HIGH);
      digitalWrite(ledMerah, LOW);
      digitalWrite(buzz, LOW);
    } else if (hasil_akhir <= 300) {
      digitalWrite(ledKuning, LOW);
      digitalWrite(ledMerah, HIGH);
      tone(buzz, 1000, 500);
    } else {
      digitalWrite(ledKuning, LOW);
      digitalWrite(ledMerah, HIGH);
      digitalWrite(buzz, HIGH);
    }

    // Mengirim hasil ke Blynk
    Blynk.virtualWrite(V1, mq2_ppm);
    Blynk.virtualWrite(V2, mq7_ppm);
    Blynk.virtualWrite(V3, hasil_akhir);

  } else {
    Serial.println("Kontrol nonaktif.");
  }
}

// Fungsi untuk mengontrol tombol on/off dari Blynk
BLYNK_WRITE(V0) {
  alatAktif = param.asInt();
}

void setup() {
  pinMode(MQ2, INPUT);
  pinMode(MQ7, INPUT);
  pinMode(buzz, OUTPUT);
  pinMode(ledKuning, OUTPUT);
  pinMode(ledMerah, OUTPUT);
  Serial.begin(115200);
  Serial.println("Starting");
```

```
  digitalWrite(ledKuning, HIGH);
  digitalWrite(ledMerah, HIGH);
  delay(1000);
  digitalWrite(ledKuning, LOW);
  digitalWrite(ledMerah, LOW);

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Menghubungkan ke WiFi..");
  }
  Serial.println(WiFi.localIP());

  Serial.print("Blynk Auth Token: ");
  Serial.println(auth);

  // Mengatur timer untuk membaca sensor setiap detik
  timer.setInterval(1000L, readSensors);
}

void loop() {
  Blynk.run();
  timer.run();

  if (!alatAktif) {
    digitalWrite(ledMerah, LOW);
    digitalWrite(ledKuning, LOW);
    digitalWrite(buzz, LOW);
  }
}
```