LAMPIRAN SOURCECODE

Kode Program Perangkat *Internet of Things*

```
#include <ESP8266HTTPClient.h>
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>

// Definisi pin untuk sensor dan perangkat
#define TRIGGER_PIN  4
#define ECHO_PIN     0
#define RAIN_SENSOR_PIN A0
#define WATER_LEVEL_PIN 13
#define RELAY1_PIN  16
#define RELAY2_PIN  12
#define LED_RED_PIN 14
#define LED_GREEN_PIN 2
#define BUZZER_PIN  15

// Konfigurasi WiFi
const char* ssid = "Galaxy A34 5G 5335";
const char* password = "012345654";
const char* server = "192.168.84.238";

// Konfigurasi Telegram Bot
const char* botToken =
"6793426512:AAFTDvPhZw67sPtaIRaGWxQy_3tqBRiw7xg";
const char* chatID = "6324139953";

WiFiClientSecure client;
UniversalTelegramBot bot(botToken, client);

// Variabel status
bool prevWaterLevelHigh = false;
bool prevIsRaining = false;

void setup() {
  Serial.begin(921600);

  // Inisialisasi pin
  pinMode(RELAY1_PIN, OUTPUT);
  pinMode(RELAY2_PIN, OUTPUT);
  pinMode(LED_RED_PIN, OUTPUT);
  pinMode(LED_GREEN_PIN, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);
```

```
  pinMode(WATER_LEVEL_PIN, INPUT);
  pinMode(TRIGGER_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);

  // konfigurasi semua perangkat di awal
  digitalWrite(RELAY1_PIN, LOW);
  digitalWrite(RELAY2_PIN, HIGH);
  digitalWrite(LED_RED_PIN, LOW);
  digitalWrite(LED_GREEN_PIN, LOW);
  digitalWrite(BUZZER_PIN, LOW);

  // Koneksi WiFi
  WiFi.hostname("nodeMCU");
  WiFi.begin(ssid, password);


  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }

  client.setInsecure(); // Menonaktifkan verifikasi sertifikat SSL/TLS

  Serial.println("WiFi connected");
}

void sendTelegramMessage(String message) {
  if (WiFi.status() == WL_CONNECTED) {
  bot.sendMessage(chatID, message, "");
  } else {
    Serial.println("WiFi not connected");
  }
}

long readUltrasonicDistance() {
  digitalWrite(TRIGGER_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIGGER_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIGGER_PIN, LOW);

  return pulseIn(ECHO_PIN, HIGH) * 0.034 / 2;
}


void loop() {
```

```
  // Membaca sensor hujan
  int rainValue = analogRead(RAIN_SENSOR_PIN);
  bool isRaining = rainValue < 500; // Threshold untuk mendeteksi hujan
  Serial.println("Sensor Hujan: " + String(rainValue));

  // sensor water level
  bool waterLevelHigh = digitalRead(WATER_LEVEL_PIN) == LOW; // Asumsi
LOW saat air mencapai sensor
  Serial.println("Sensor Water Level: " + String(waterLevelHigh));

  // sensor HC-SR04
  long distance = readUltrasonicDistance();
  Serial.println("Jarak: " + String(distance) + " cm");

  // Kirim ke database
  if (WiFi.status() == WL_CONNECTED) {
    WiFiClient client;
    HTTPClient http;
    String link = "http://" + String(server) + "/waterlevel/kirimdata.php?tinggi=" +
String(distance) + "&rain=" + bool(isRaining);
    http.begin(client, link);
    int httpCode = http.GET();
    if (httpCode > 0) {
      Serial.println("Data sent successfully");
    } else {
      Serial.println("Error sending data: " +
String(http.errorToString(httpCode).c_str()));
    }
    http.end();
  } else {
    Serial.println("WiFi not connected");
  }

  // Logika kontrol
  if (isRaining) {
    digitalWrite(LED_GREEN_PIN, HIGH); // Nyalakan LED hijau
    delay(2000); // Tahan selama 2 detik
    digitalWrite(LED_GREEN_PIN, LOW); // Matikan LED hijau

    if (!prevIsRaining) {
      sendTelegramMessage("Hujan terdeteksi!");
      prevIsRaining = true;
    }
  } else {
    prevIsRaining = false;
  }
```

```
  if (waterLevelHigh) {
    digitalWrite(RELAY1_PIN, HIGH); // Aktifkan relay 1 dan mini pump
    digitalWrite(LED_RED_PIN, HIGH); // Nyalakan LED merah
    digitalWrite(BUZZER_PIN, HIGH); // Nyalakan buzzer

    if (!prevWaterLevelHigh) {
      sendTelegramMessage("Banjir terdeteksi! Pump menyala.");
      prevWaterLevelHigh = true;
    }


    if (distance < 5) {
      digitalWrite(RELAY2_PIN, LOW); // Matikan arus listrik dari LED merah,
buzzer, dan HC-SR04
      digitalWrite(LED_RED_PIN, LOW);
      digitalWrite(BUZZER_PIN, LOW);

      sendTelegramMessage("Banjir melebihi batas biasanya, arus listrik terputus.");
    } else if (distance > 8) {
      digitalWrite(RELAY2_PIN, HIGH); // Aktifkan kembali relay 2
    }
  } else {
    digitalWrite(RELAY1_PIN, LOW);
    digitalWrite(RELAY2_PIN, HIGH);
    digitalWrite(LED_RED_PIN, LOW);
    digitalWrite(BUZZER_PIN, LOW);

    prevWaterLevelHigh = false;
  }

  // Interval untuk pengukuran berikutnya
  delay(1000);
}
```

Kode program *interface* website

Index.php

```
<!DOCTYPE html>
<html>
<head>

        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>Monitoring tester</title>

        <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">
```

```
        <link rel="stylesheet" type="text/css" href="css/rainy.css">
        <script type="text/javascript" src="jquery/jquery.min.js"></script>
        <script type="text/javascript" src="js/bootstrap.min.js"></script>
<script type="text/javascript" src="js/rainy.js"></script>
        <style type="text/css">
                .tangki {
                        border-style: solid;
                        width: 300px;
                        height:  300px;
                        left: 50%;
                        transform: translate(-50%);
                        position: sticky;
                        border-bottom-left-radius: 20px;
                        border-bottom-right-radius: 20px;
                }
                .air {
                        left: 50%;
                        bottom: 0px;
                        transform: translate(-50%);
                        position: absolute;
                        border-bottom-left-radius: 20px;
                        border-bottom-right-radius: 20px;
                        background-color: blue;
                }
                .penutup {
                        border-style: solid;
                        width: 300px;
                        height: 40px;
                        left: 50%;
                        transform: translate(-50%);
                        position: sticky;
                        border-top-left-radius: 20px;
                        border-top-right-radius: 20px;
                }
                .pegangan {
                        border-style: solid;
                        width: 40px;
                        height: 20px;
                        left: 50%;
                        transform: translate(-50%);
                        position: sticky;
                        border-top-left-radius: 20px;
                        border-top-right-radius: 20px;
                }

        </style>
```

```
        <!-- ajax untuk realtime -->
        <script type="text/javascript">
                $(document).ready(function() {
                        setInterval(function(){
                                $("#data").load('data.php')
                        }, 1000);
                                                setInterval(function(){
                                $("#logo-container").load('logo.php')
                        }, 1000);
                });
        </script>

</head>
<body>

        <!--Tampilan aplikasi -->
        <div class="container" style="text-align: center;">

                <!-- <img src="images/rainy-2.png" style="width: 180px; margin-top:
25px; object-position: top;">
                                <img src="https://w7.pngwing.com/pngs/487/562/png-
transparent-sun-logo-sunlight-silhouette-thumbnail.png" style="width: 180px;
margin-top: 25px; object-position: top;"> -->
                <div id="logo-container">

                </div>
                <h2>Sistem monitoring deteksi banjir <br> Berbasis Web</h2>
                <div style="font-size: 20px;">jarak tinggi air banjir</div>

                <div id="data"></div>

        </div>
</body>
</html>
```

Lampiran 3 SourceCode