

LAMPIRAN

SCRIPT ARDUINO

```
#include <Arduino.h>

#include <WiFi.h>

#include <ESP32Servo.h>

#include <Firebase_ESP_Client.h>

// Insert your network credentials

#define WIFI_SSID "NierJur"

#define WIFI_PASSWORD "kucingberanak"

// Insert Firebase project API Key

#define API_KEY "AIzaSyB46dwY1BrRl80WEVAnvhIqiShNUVdNWmY"

// Insert RTDB URL

#define DATABASE_URL "https://kontrol-irigasi-v1-default-rtdb.asia-southeast1.firebaseio.com"

// PIN

#define PIN_PINTU1 27 // Pin servo pintu 1

#define PIN_PINTU2 14 // Pin servo pintu 2

#define relayPin 25 // Relay

const int sensorPin = 26; // Define pin untuk sensor YF-S201 atau sensor aliran air

const int pinSensor = 33; // Pin sensor water level

// Variabel global untuk sensor flow

volatile int pulseCount = 0;

float debit = 0.0; // Debit awal

const float calibrationFactor = 0.00265; // Faktor kalibrasi (L/pulse)

float totalVolume = 0.0; // Total volume air yang telah mengalir

float totalVolumeHarian = 0.0; // Total volume per hari

unsigned long lastPulseTime = 0; // Waktu dari detik terakhir terdeteksi pulsa

unsigned long resetTime = 0; // Waktu untuk reset totalVolume

const unsigned long resetInterval = 5 * 60 * 1000; // 5 menit dalam milidetik

// Variabel global untuk sensor water level

int tinggiAir = 0;

// Define Firebase Data object

FirebaseData fbdo, fbdo_s1, fbdo_s2, fbdo_s3;

FirebaseAuth auth;

FirebaseConfig config;

// Define servo

Servo pintu1;

Servo pintu2;

// Define manual override flag

bool manual = false;

int status1;

// Variables for volume edit

float volumeEdit = 0; // Total volume air perubahan data

float initialTotalVolume = 0;

float initialVolumeEdit = 0;

// Timer for sending data
```

```

unsigned long sendDataPrevMillis = 0;

bool signupOK = false;

void IRAM_ATTR pulseCounter() {
    pulseCount++;
}

void openGate() {
    Serial.println("Opening gate...");
    pintu1.write(20);
}

void closeGate() {
    Serial.println("Closing gate...");
    pintu1.write(180); // Rotate servo to 180 degrees
}

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

    // Initialize Wi-Fi
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    Serial.print("Connecting to Wi-Fi");
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print(".");
        delay(300);
    }
    Serial.println();
    Serial.print("Connected with IP: ");
    Serial.println(WiFi.localIP());
    Serial.println();

    // Initialize Firebase
    config.api_key = API_KEY;
    config.database_url = DATABASE_URL;

```

```

    if (Firebase.signUp(&config, &auth, "", "")) {
        Serial.println("Firebase signup successful");
        signupOK = true;
    } else {
        Serial.printf("Firebase signup failed: %s\n",
config.signer.signupError.message.c_str());
    }

    Firebase.begin(&config, &auth);
    Firebase.reconnectWiFi(true);

    pintu1.attach(PIN_PINTU1);
    pintu1.write(180); // Titik awal servo
    pintu2.attach(PIN_PINTU2);
    pintu2.write(0);

    // Set pin sensor sebagai input dan aktifkan internal
pull-up resistor
    pinMode(sensorPin, INPUT_PULLUP);
    pinMode(pinSensor, INPUT);
    pinMode(relayPin, OUTPUT);

    // Reset variabel
    pulseCount = 0;
    lastPulseTime = millis();
    resetTime = millis();

    // Attach interrupt untuk sensor YF-S201
    attachInterrupt(digitalPinToInterrupt(sensorPin),
pulseCounter, FALLING);

    // Stream volume air dan manualOverride
    if (!Firebase.RTDB.beginStream(&fbdo_s1,
"SensorWaterFlow/volumeEdit")) {

```

```

        Serial.printf("Stream 1 begin error, %s\n\n",
fbdo_s1.errorReason().c_str());
    }

    if (!Firebase.RTDB.beginStream(&fbdo_s2,
"Control/manualOverride")) {
        Serial.printf("Stream manualOverride begin error,
%s\n\n", fbdo_s2.errorReason().c_str());
    }
    if (!Firebase.RTDB.beginStream(&fbdo_s3,
"Pintu1/Status")) {
        Serial.printf("Stream Pintu 1 begin error, %s\n\n",
fbdo_s3.errorReason().c_str());
    }
}

void loop() {
    unsigned long currentTime = millis();

    // Stream manualOverride
    if (Firebase.RTDB.readStream(&fbdo_s2)) {
        if (fbdo_s2.streamAvailable()) {
            if (fbdo_s2.dataPath() == "/") {
                bool newmanual = fbdo_s2.boolData();
                Serial.print("Status Manual (boolean): ");
                Serial.println(newmanual);
                manual = newmanual;
            }
        }
    } else {
        Serial.printf("Stream read error, %s\n",
fbdo_s2.errorReason().c_str());
    }
}

```

```

// Stream manualOverride
if (Firebase.RTDB.readStream(&fbdo_s3)) {
    if (fbdo_s3.streamAvailable()) {
        if (fbdo_s3.dataPath() == "/") {
            int newstatus = fbdo_s3.boolData();
            Serial.print("Status Pintu : ");
            Serial.println(newstatus);
            status1 = newstatus;
        }
    }
} else {
    Serial.printf("Stream read error, %s\n",
fbdo_s3.errorReason().c_str());
}

// Sensor Water Level
int nilaiAir = analogRead(pinSensor);
Serial.print("Sensor Air Value = ");
Serial.println(nilaiAir);

// Kontrol servo pintu berdasarkan status manual dan
level air
if (manual) {
    Serial.println("Manual Override is ON");
    if (status1 == 1) {
        openGate();
    } else if (status1 == 0) {
        closeGate();
    }
} else {
    if (nilaiAir >= 3200) {
        pintu1.write(20); // Fully open position
        status1 = 1;
        Serial.println("Pintu Terbuka");
    }
}

```

```

    } else if (nilaiAir > 2800 && nilaiAir < 3200) {

        pintu1.write(80); // Half open position

        status1 = 2;

        Serial.println("Pintu Terbuka Setengah");
    } else {

        pintu1.write(180); // Closed position

        status1 = 0;

        Serial.println("Pintu Tertutup");
    }

    // Update status to Firebase

    if (Firebase.RTDB.setInt(&fbdo, "/Pintu1/Status",
status1)) {

        Serial.println("Status Pintu1 berhasil diperbarui
di Firebase");

        } else {

            Serial.printf("Gagal memperbarui status Pintu1:
%s\n", fbdo.errorReason().c_str());

        }

    }

    if (nilaiAir >= 3200) {

        Serial.println("Relay OFF");

        digitalWrite(relayPin, HIGH);

        tinggiAir = 1;

        Serial.println("Status Level Air Tinggi");

    } else if (nilaiAir > 2800 && nilaiAir < 3200) {

        tinggiAir = 2;

        Serial.println("Status Level Air Sedang");

    } else {

        Serial.println("Relay Nyala");

        digitalWrite(relayPin, LOW);

        tinggiAir = 0;

        Serial.println("Status Level Air Rendah");

    }

}

// Sensor Waterflow

if (millis() - lastPulseTime > 2000) {

    detachInterrupt(digitalPinToInterrupt(sensorPin));

    debit = calibrationFactor * pulseCount / (2000 /
1000.0); // Menggunakan calibrationFactor, hasil dalam
liter per detik

    float volume = debit * (2000 / 1000.0); // Volume
dalam interval waktu

    totalVolumeHarian += volume;

    totalVolume += volume;

    Serial.print("Debit: ");

    Serial.print(debit, 4);

    Serial.println(" L/s");

    Serial.print("Total Volume Harian: ");

    Serial.print(totalVolumeHarian, 4);

    Serial.println(" L");

    Serial.print("Total Volume: ");

    Serial.print(totalVolume, 4);

    Serial.println(" L");

    pulseCount = 0;

    lastPulseTime = millis();

    attachInterrupt(digitalPinToInterrupt(sensorPin),
pulseCounter, FALLING);

}

// Reset totalVolumeHarian

if ((millis() - resetTime) > resetInterval) {

    totalVolumeHarian = initialTotalVolume;

```

```

        Serial.println("5 menit berlalu. Reset totalVolume
ke nilai awal.");

        resetTime = millis(); // Set reset time baru
    }

    // Mengambil data volumeEdit dari Firebase
    if (Firebase.RTDB.readStream(&fbdo_s1)) {
        if (fbdo_s1.streamAvailable()) {
            if (fbdo_s1.dataPath() == "/") {
                float newVolumeEdit = fbdo_s1.floatData();

                Serial.print("Volume Edit Value (float): ");
                Serial.println(newVolumeEdit);

                if (volumeEdit != newVolumeEdit) {
                    volumeEdit = newVolumeEdit;
                }
            }
        }
    }

    Serial.print("Volume Edit: ");
    Serial.println(volumeEdit);

    // Kontrol servo pintu kedua
    int status2;

    if (totalVolume < volumeEdit) {
        pintu2.write(135); // Open position

        status2 = 1; // Pintu terbuka

        Serial.println("Pintu 2 terbuka.");
    } else {
        pintu2.write(0); // Closed position

        status2 = 0; // Pintu tertutup

        Serial.println("Pintu 2 tertutup.");

        totalVolume = initialTotalVolume;
        volumeEdit = initialVolumeEdit;

```

```

        if (Firebase.RTDB.setFloat(&fbdo,
"SensorWaterFlow/volumeEdit", volumeEdit)) {
            Serial.println("PASSED");

            Serial.println("PATH: " + fbdo.dataPath());
            Serial.println("TYPE: " + fbdo.dataType());
        } else {
            Serial.println("FAILED");

            Serial.println("REASON: " +
fbdo.errorReason());
        }

        Serial.println("Total volume tercapai. Reset
totalVolume ke nilai awal.");
    }

    // Mengirim data ke Firebase
    if (Firebase.ready() && signupOK && (millis() -
sendDataPrevMillis > 2000 || sendDataPrevMillis == 0))
    {
        sendDataPrevMillis = millis();

        if (Firebase.RTDB.setInt(&fbdo, "Sensor/Value",
nilaiAir)) {
            Serial.println("PASSED");

            Serial.println("PATH: " + fbdo.dataPath());
            Serial.println("TYPE: " + fbdo.dataType());
        } else {
            Serial.println("FAILED");

            Serial.println("REASON: " +
fbdo.errorReason());
        }

        if (Firebase.RTDB.setInt(&fbdo,
"Sensor/TinggiAir", tinggiAir)) {
            Serial.println("PASSED");

            Serial.println("PATH: " + fbdo.dataPath());

```

```

        Serial.println("TYPE: " + fbdo.dataType());
    } else {
        Serial.println("FAILED");
        Serial.println("REASON: " +
fbdo.errorReason());
    }

    if (Firebase.RTDB.setInt(&fbdo, "Pintu1/Status",
status1)) {
        Serial.println("PASSED");
        Serial.println("PATH: " + fbdo.dataPath());
        Serial.println("TYPE: " + fbdo.dataType());
    } else {
        Serial.println("FAILED");
        Serial.println("REASON: " +
fbdo.errorReason());
    }

    if (Firebase.RTDB.setInt(&fbdo, "Pintu2/Status",
status2)) {
        Serial.println("PASSED");
        Serial.println("PATH: " + fbdo.dataPath());
        Serial.println("TYPE: " + fbdo.dataType());
    } else {
        Serial.println("FAILED");
        Serial.println("REASON: " +
fbdo.errorReason());
    }
}

        if (Firebase.RTDB.setFloat(&fbdo,
"SensorWaterFlow/debit", debit)) {
            Serial.println("PASSED");
            Serial.println("PATH: " + fbdo.dataPath());
            Serial.println("TYPE: " + fbdo.dataType());
        } else {
            Serial.println("FAILED");
            Serial.println("REASON: " +
fbdo.errorReason());
        }

        if (Firebase.RTDB.setFloat(&fbdo,
"SensorWaterFlow/volume", totalVolumeHarian)) {
            Serial.println("PASSED");
            Serial.println("PATH: " + fbdo.dataPath());
            Serial.println("TYPE: " + fbdo.dataType());
        } else {
            Serial.println("FAILED");
            Serial.println("REASON: " +
fbdo.errorReason());
        }

        delay(100); // Mengurangi delay untuk loop yang lebih
cepat
    }
}

```