

Lampiran

```

#include <WiFi.h>
#include <PubSubClient.h>
#include <WiFiManager.h>

// =====
// PIN SENSOR & KOMPONEN
// =====

#define TRIG_PIN    5
#define ECHO_PIN    18
#define BUZZER_PIN  13
#define LED_PIN     2
#define RELAY_PIN   14
#define BUTTON_PIN  27

// =====
// MQTT CONFIG
// =====

const char* mqtt_server = "broker.hivemq.com";
const char* topicStatus = "restudwiandira/trap/status";
const char* topicCommand = "restudwiandira/trap/command";

// =====
// OBJECT MQTT
// =====

WiFiClient espClient;
PubSubClient client(espClient);

```

```

// =====
// VARIABLE SISTEM
// =====

bool systemActive    = true;
bool alarmActive     = true;
bool trapTriggered   = false;
bool wifiConnected   = false;

// Fase reset: tunggu 4 detik relay ON
// sebelum relay OFF kembali menahan
bool sedangReset     = false;
unsigned long waktuReset = 0;

// Fase trigger: relay ON 2 detik
// lalu relay OFF (pintu sudah jepret)
bool sedangTrigger    = false;
unsigned long waktuTrigger = 0;

unsigned long lastButtonPress = 0;
int buttonCount = 0;

// =====
// SOLENOID HELPER
// =====

void solenoidBuka() {
    digitalWrite(RELAY_PIN, HIGH);
    Serial.println("Solenoid: BUKA");
}

```

```

void solenoidTahan() {
    digitalWrite(RELAY_PIN, LOW);
    Serial.println("Solenoid: TAHAN");
}

// =====
// MULAI RESET (WA atau tombol manual)
// relay ON 4 detik → relay OFF tahan
// =====

void mulaiReset() {
    if (!trapTriggered) return;
    sedangReset = true;
    waktuReset = millis();
    solenoidBuka();
    Serial.println(
        "Reset: relay ON 4 detik, buka pintu..."
    );
}

// =====
// EKSEKUSI RESET setelah 4 detik
// =====

void eksekusiReset() {
    sedangReset = false;
    trapTriggered = false;
    alarmActive = true;

    solenoidTahan();
}

```

```

digitalWrite(BUZZER_PIN, LOW);
digitalWrite(LED_PIN,    LOW);

Serial.println(
  "Reset selesai, perangkat standby."
);

if (wifiConnected && client.connected()) {
  client.publish(
    topicStatus,
    "{\"status\":\"AMAN\"}"
  );
}
}

// =====
// WIFI MANAGER
// =====

void startConfigPortal() {
  WiFiManager wm;
  wm.startConfigPortal(
    "TRAP-IOT-SETUP",
    "12345678"
  );
  Serial.println("WiFi Configured");
}

void connectWiFi() {
  WiFi.mode(WIFI_STA);

```

```

WiFi.begin();

Serial.println("");
Serial.println("Connecting WiFi...");

unsigned long startAttempt = millis();

while (
  WiFi.status() != WL_CONNECTED &&
  millis() - startAttempt < 5000
) {
  delay(500);
  Serial.print(".");
}

if (WiFi.status() == WL_CONNECTED) {
  wifiConnected = true;
  Serial.println("");
  Serial.println("WiFi Connected");
  Serial.println(WiFi.localIP());
} else {
  wifiConnected = false;
  Serial.println("");
  Serial.println("Offline Mode");
}
}

// =====
// CALLBACK MQTT
// =====

```

```

void callback(
    char* topic,
    byte* payload,
    unsigned int length
) {

    String message = "";
    for (int i = 0; i < length; i++) {
        message += (char)payload[i];
    }

    Serial.println(
        "Perintah masuk: " + message
    );

    // Nonaktifkan sistem
    if (
        message.indexOf(
            "\"system\":false"
        ) >= 0
    ) {
        systemActive = false;
        Serial.println("Sistem dinonaktifkan");
    }

    // Aktifkan sistem
    if (
        message.indexOf(
            "\"system\":true"
        ) >= 0
    ) {

```

```

    systemActive = true;
    trapTriggered = false;
    Serial.println("Sistem diaktifkan");
}

// Matikan alarm → mulai reset 4 detik
if (
    message.indexOf(
        "\"alarm\":false"
    ) >= 0
) {
    mulaiReset();
    Serial.println(
        "Alarm Dimatikan Dari WhatsApp"
    );
}

// Hidupkan alarm
if (
    message.indexOf(
        "\"alarm\":true"
    ) >= 0
) {
    alarmActive = true;
    Serial.println("Alarm Dihidupkan");
}
}

// =====
// RECONNECT MQTT
// =====

```

```

void reconnectMQTT() {

    if (!wifiConnected) return;

    while (!client.connected()) {

        Serial.println("Connecting MQTT...");

        String clientId = "ESP32Trap-";
        clientId += String(random(0xffff), HEX);

        if (client.connect(clientId.c_str())) {
            Serial.println("MQTT Connected");
            client.subscribe(topicCommand);
        } else {
            Serial.println("MQTT Failed");
            delay(2000);
        }
    }
}

// =====
// BACA JARAK ULTRASONIK
// =====

float readDistance() {

    digitalWrite(TRIG_PIN, LOW);
    delayMicroseconds(2);

```



```

    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);

    digitalWrite(TRIG_PIN, LOW);

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

// =====
// BLINK ALARM
// =====

void blinkAlarm() {

    static unsigned long lastBlink = 0;
    static bool blinkState = false;

    if (!trapTriggered) return;
    if (!alarmActive) return;

    if (millis() - lastBlink > 300) {
        lastBlink = millis();
        blinkState = !blinkState;

        digitalWrite(BUZZER_PIN, blinkState);
        digitalWrite(LED_PIN, blinkState);
    }
}

// =====

```

```

// CEK PUSH BUTTON
// =====

void checkButton() {

    static bool lastState = HIGH;
    bool currentState = digitalRead(BUTTON_PIN);

    if (
        lastState == HIGH &&
        currentState == LOW
    ) {
        buttonCount++;
        lastButtonPress = millis();
        delay(50);
    }

    lastState = currentState;

    // Double click → buka WiFi Manager
    if (
        buttonCount == 2 &&
        millis() - lastButtonPress < 500
    ) {
        Serial.println("Open WiFi Manager");
        startConfigPortal();
        ESP.restart();
    }

    // Single click → mulai reset 4 detik
    if (

```

```

        buttonCount == 1 &&
        millis() - lastButtonPress > 500
    ) {
        Serial.println("Tombol Manual Ditekan");
        mulaiReset();
        buttonCount = 0;
    }

    if (millis() - lastButtonPress > 1000) {
        buttonCount = 0;
    }
}

// =====
// SETUP
// =====

void setup() {

    Serial.begin(115200);

    Serial.println("");
    Serial.println(
        "=====
    );
    Serial.println(
        " SISTEM PERANGKAP HEWAN OTOMATIS "
    );
    Serial.println(
        "=====
    );

```

```

Serial.println("");
Serial.println(
    "Nama Mahasiswa : RESTU DWI ANDIRA"
);
Serial.println(
    "NPM          : 2210010123"
);
Serial.println("");
Serial.println("ESP32 Starting...");
Serial.println("");

pinMode(TRIG_PIN,    OUTPUT);
pinMode(ECHO_PIN,    INPUT);
pinMode(BUZZER_PIN,  OUTPUT);
pinMode(LED_PIN,     OUTPUT);
pinMode(RELAY_PIN,   OUTPUT);
pinMode(BUTTON_PIN,  INPUT_PULLUP);

digitalWrite(BUZZER_PIN, LOW);
digitalWrite(LED_PIN,    LOW);

// Posisi awal: solenoid buka 3 detik → tahan
Serial.println("Inisialisasi solenoid...");
solenoidBuka();
delay(3000);
solenoidTahan();
Serial.println(
    "Solenoid standby, perangkap siap."
);

connectWiFi();

```

```

    client.setServer(mqtt_server, 1883);
    client.setCallback(callback);
}

// =====
// LOOP
// =====

void loop() {

    checkButton();
    blinkAlarm();

    // =====
    // TIMER TRIGGER: relay ON 2 detik
    // lalu OFF → pintu sudah jepret, standby
    // =====
    if (
        sedangTrigger &&
        millis() - waktuTrigger >= 2000
    ) {
        sedangTrigger = false;
        solenoidTahan();
        Serial.println(
            "Trigger selesai, solenoid menahan."
        );
    }

    // =====
    // TIMER RESET: relay ON 4 detik

```

```

// lalu OFF → pintu terpasang kembali
// =====
if (
    sedangReset &&
    millis() - waktuReset >= 4000
) {
    eksekusiReset();
}

if (
    wifiConnected &&
    !client.connected()
) {
    reconnectMQTT();
}

client.loop();

if (!systemActive) {
    delay(300);
    return;
}

// Baca ultrasonik hanya saat standby
if (
    !trapTriggered &&
    !sedangReset &&
    !sedangTrigger
) {

    float distance = readDistance();

```

```

Serial.print("Distance: ");
Serial.println(distance);

if (distance > 0 && distance <= 10) {

    trapTriggered = true;
    sedangTrigger = true;
    waktuTrigger = millis();
    Serial.println("HEWAN TERDETEKSI");
    // Relay ON → solenoid mundur 2 detik
    // → pintu perangkap jepret
    solenoidBuka();
    // Kirim notif ke bot WA
    if (
        wifiConnected &&
        client.connected()
    ) {
        client.publish(
            topicStatus,
            "{\"status\":\"HEWAN TERDETEKSI\"}"
        );
    }
}
}
delay(300);
}

```