

## LAMPIRAN

```
#define BLYNK_TEMPLATE_ID "TMPL6yF3kYOJW"
#define BLYNK_TEMPLATE_NAME "Kotak Obat"
#define BLYNK_AUTH_TOKEN "-iprbIuhMYVtbHCqJ_-L8liyKBZ-QaY8"

#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SoftwareSerial.h>
#include <DFRobotDFPlayerMini.h>
#include <TimeLib.h>
#include <WidgetRTC.h>
#include <Servo.h>

// ===== WIFI =====
char ssid[] = "LAB 1";
char pass[] = "1122334455";

// ===== PIN =====
#define SERVO_PIN D5
#define BUTTON_PIN D2
#define TRIG_PIN D3
#define ECHO_PIN D4

// ===== DFPLAYER =====
SoftwareSerial mySerial(D6, D7);
DFRobotDFPlayerMini dfplayer;
```

```

// ===== SERVO =====
Servo servo;

// ===== RTC =====
WidgetRTC rtc;

// ===== JADWAL =====
int jam[3] = {0,0,0};
int menit[3] = {0,0,0};
bool sudahTrigger[3] = {false, false, false};

// ===== SYSTEM =====
unsigned long lastTriggerTime = 0;
bool sedangJalan = false;

// =====

// ===== BLYNK MANUAL =====
BLYNK_WRITE(V3) {
  if (param.asInt() == 1) {
    Serial.println("[BLYNK] Manual trigger dari app");
    jalankanManual();
  }
}

// =====

void kirimStatus(String status) {
  Serial.print("[STATUS] ");
  Serial.println(status);
  Blynk.virtualWrite(V4, status);
}

```

```

}

void kirimLog(String pesan) {
  Serial.print("[LOG] ");
  Serial.print(pesan);
  Blynk.virtualWrite(V5, pesan);
}

// =====

long bacaJarak() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  long durasi = pulseIn(ECHO_PIN, HIGH, 30000);
  long jarak = durasi * 0.034 / 2;

  Serial.print("[SENSOR] Jarak: ");
  Serial.print(jarak);
  Serial.println(" cm");

  return jarak;
}

// =====

void bukaTungguAmbil(int index) {

```

```

unsigned long startTime = millis();
unsigned long detectTime = 0;
bool tanganTerdeteksi = false;
bool obatDiambil = false;

Serial.println("[SERVO] Terbuka");
servo.write(90);

 kirimStatus("Menunggu diambil...");

 while (millis() - startTime < 15000) {

    long jarak = bacaJarak();

    if (jarak > 0 && jarak < 10) {

        if (!tanganTerdeteksi) {
            Serial.println("[INFO] Tangan terdeteksi");
            detectTime = millis();
            tanganTerdeteksi = true;
        }

        if (millis() - detectTime >= 5000) {
            Serial.println("[RESULT] Obat diambil");
            obatDiambil = true;
            break;
        }

    } else {
        tanganTerdeteksi = false;
    }
}

```

```

    delay(100);
}

Serial.println("[SERVO] Tertutup");
servo.write(0);

String waktuStr = String(hour()) + ":" + String(minute());

if (obatDiambil) {

    kirimStatus("Obat diambil");

    String logMsg;
    if (index == 0) logMsg = "PAGI - DIAMBIL " + waktuStr;
    else if (index == 1) logMsg = "SIANG - DIAMBIL " + waktuStr;
    else if (index == 2) logMsg = "MALAM - DIAMBIL " + waktuStr;
    else logMsg = "MANUAL - DIAMBIL " + waktuStr;

    kirimLog(logMsg + "\n");

} else {

    Serial.println("[WARNING] Obat tidak diambil!");

    kirimStatus("Obat belum diambil!");

    String logMsg;
    if (index == 0) logMsg = "PAGI - TIDAK DIAMBIL " + waktuStr;
    else if (index == 1) logMsg = "SIANG - TIDAK DIAMBIL " + waktuStr;
    else if (index == 2) logMsg = "MALAM - TIDAK DIAMBIL " + waktuStr;

```

```

else logMsg = "MANUAL - TIDAK DIAMBIL " + waktuStr;

    kirimLog(logMsg + "\n");

    Blynk.logEvent("obat_tidak_diambil", logMsg);
}

delay(2000);
kirimStatus("Siap");
}

// =====

// ===== JADWAL =====
BLYNK_WRITE(V0) {
    int t = param.asInt();
    jam[0] = t / 3600;
    menit[0] = (t % 3600) / 60;

    Serial.print("[SET] PAGI = ");
    Serial.print(jam[0]);
    Serial.print(":");
    Serial.println(menit[0]);

    sudahTrigger[0] = false;
}

BLYNK_WRITE(V1) {
    int t = param.asInt();
    jam[1] = t / 3600;
    menit[1] = (t % 3600) / 60;
}

```

```

Serial.print("[SET] SIANG = ");
Serial.print(jam[1]);
Serial.print(":");
Serial.println(menit[1]);

sudahTrigger[1] = false;
}

BLYNK_WRITE(V2) {
  int t = param.asInt();
  jam[2] = t / 3600;
  menit[2] = (t % 3600) / 60;

  Serial.print("[SET] MALAM = ");
  Serial.print(jam[2]);
  Serial.print(":");
  Serial.println(menit[2]);

  sudahTrigger[2] = false;
}

// =====

void setup() {
  Serial.begin(9600);

  Serial.println("=====");
  Serial.println("[BOOT] Sistem mulai...");
  Serial.println("=====");
}

```

```

pinMode(BUTTON_PIN, INPUT_PULLUP);
pinMode(TRIG_PIN, OUTPUT);
pinMode(ECHO_PIN, INPUT);

servo.attach(SERVO_PIN);
servo.write(0);

mySerial.begin(9600);
if (dfplayer.begin(mySerial)) {
  Serial.println("[DFPLAYER] Ready");
  dfplayer.volume(25);
} else {
  Serial.println("[DFPLAYER] ERROR");
}

Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
Serial.println("[WIFI] Terhubung");

rtc.begin();
setSyncInterval(600);
Serial.println("[RTC] Sinkron OK");

 kirimStatus("Siap");
}

// =====

void loop() {
  Blynk.run();

  if (year() < 2020) {

```

```

Serial.println("[RTC] Menunggu waktu...");
delay(1000);
return;
}

int h = hour();
int m = minute();

Serial.print("[TIME] ");
Serial.print(h);
Serial.print(":");
Serial.println(m);

// tombol fisik
if (digitalRead(BUTTON_PIN) == LOW && !sedangJalan) {
  delay(50);
  if (digitalRead(BUTTON_PIN) == LOW) {
    Serial.println("[BUTTON] Manual trigger");
    jalankanManual();
  }
}

// jadwal
for (int i = 0; i < 3; i++) {
  if (h == jam[i] && m == menit[i] && !sudahTrigger[i]) {

    Serial.print("[JADWAL] Trigger ke-");
    Serial.println(i + 1);

    jalankanDispenser(i);
  }
}

```

```

String msg;
if (i == 0) msg = "PAGI";
else if (i == 1) msg = "SIANG";
else msg = "MALAM";

Blynk.logEvent("minum_obat", msg);

    sudahTrigger[i] = true;
}
}

// reset harian
if (h == 0 && m == 0) {
    Serial.println("[RESET] Reset harian");
    for (int i = 0; i < 3; i++) {
        sudahTrigger[i] = false;
    }
}

delay(1000);
}

// =====

void jalankanDispenser(int index) {

    if (millis() - lastTriggerTime < 5000) return;

    sedangJalan = true;
    lastTriggerTime = millis();
}

```

```

Serial.print("[AUDIO] Play: ");
Serial.println(index + 1);

dfplayer.playMp3Folder(index + 1);

delay(4000); // □ SESUAIKAN DURASI AUDIO

bukaTungguAmbil(index);

sedangJalan = false;
}

// =====

void jalankanManual() {

if (millis() - lastTriggerTime < 5000) return;

sedangJalan = true;
lastTriggerTime = millis();

Serial.println("[AUDIO] Play: 4 (Manual)");

dfplayer.playMp3Folder(4);

delay(4000); // □ SESUAIKAN

bukaTungguAmbil(3);

sedangJalan = false;
}

```