

Source Code

```
#define BLYNK_TEMPLATE_ID "TMPL6A6HcoYj7"
#define BLYNK_TEMPLATE_NAME "Alat Yesa"
#define BLYNK_AUTH_TOKEN "5f4j-ma5qm8kDtAhSLObj_tTQ8TAUzGG"

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DHT.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <PubSubClient.h>

#define DHTPIN 5
#define DHTTYPE DHT22
#define RELAY_PIN 26

char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Gass";
char pass[] = "085269527272";

const char* mqtt_server = "broker.emqx.io";
const char* mqtt_topic = "topic/temperature";
WiFiClient espClient;
PubSubClient client(espClient);

DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
  pinMode(RELAY_PIN, OUTPUT);
  digitalWrite(RELAY_PIN, HIGH); // Pastikan relay mati saat dinyalakan

  dht.begin();
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Suhu:");
  lcd.setCursor(0, 1);
  lcd.print("Kelembapan:");

  Blynk.begin(auth, ssid, pass);

  Serial.begin(115200);
  Serial.println("Connecting to WiFi...");
```

```

// Connect to WiFi
WiFi.begin(ssid, pass); // Menghubungkan ke jaringan Wi-Fi
while (WiFi.status() != WL_CONNECTED) {
  delay(500); // Tunggu setengah detik dan coba lagi
  Serial.print(".");
}
Serial.println(" Connected to WiFi!");
Serial.print("IP Address: ");
Serial.println(WiFi.localIP()); // Tampilkan IP ESP32 yang terkoneksi

// Connect to MQTT
client.setServer(mqtt_server, 1883); // Set your Mosquitto broker IP address
connectMQTT();
}

void loop() {
  if (!Blynk.connected()) {
    Blynk.connect();
  }
  Blynk.run();

  if (WiFi.status() != WL_CONNECTED) {
    reconnectWiFi();
  }

  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();

  if (isnan(humidity) || isnan(temperature)) {
    return;
  }

  lcd.setCursor(5, 0);
  lcd.print(temperature);
  lcd.print("C");
  lcd.setCursor(11, 1);
  lcd.print(humidity);
  lcd.print("%");

  Blynk.virtualWrite(V1, temperature);
  Blynk.virtualWrite(V2, humidity);

  publishDataToMQTT(temperature);

  if (temperature > 40.0) {

```

```

    digitalWrite(RELAY_PIN, HIGH);
  } else {
    digitalWrite(RELAY_PIN, LOW);
  }

  delay(1000);
}

void reconnectWiFi() {
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println(" Reconnected to WiFi!");
  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP()); // Show new IP address after reconnecting
}

void connectMQTT() {
  while (!client.connected()) {
    Serial.print("Connecting to MQTT... ");
    if (client.connect("ESP32Client")) {
      Serial.println("connected");
      // Once connected, subscribe to a topic (if needed)
      // client.subscribe("home/temperature"); // Example subscription
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      delay(2000); // Wait 2 seconds before retrying
    }
  }
}

// Function to publish temperature and humidity data to MQTT
void publishDataToMQTT(float temperature) {
  if (!client.connected()) {
    connectMQTT();
  }

  // Create a JSON payload
  String payload = "{\"temp\":";
  payload += String(temperature, 2);
  payload += "}";

  Serial.print("Payload to MQTT: ");

```

```
Serial.println(payload);

// Publish temperature and humidity data to MQTT broker
if (client.publish(mqtt_topic, payload.c_str())) {
    Serial.println("Data published to MQTT");
} else {
    Serial.println("Failed to publish data to MQTT");
}
}
```

LAMPIRAN

Dokumentasi Kegiatan





