

DOKUMENTASI PENGERJAAN FINAL PROJECT

INTEGRASI SISTEM (B)

Kelompok 8

- **Etha Felisya Br Purba (5027221017)**
 - **Salsabila Amalia Harjanto (5027221023)**
-

Video Penggeraan (Youtube)

- <https://www.youtube.com/watch?v=Hcz8xTzlaG8>

Source Code (Github)

- <https://github.com/zaqueen/FP-Integrasi-Sistem-5027221017-5027221023->
-

PROJECT 1

Perangkat yang Digunakan

- ESP32: Sebagai mikrokontroler.
- Sensor DHT22: Mengukur suhu dan kelembaban.

Fungsionalitas

- Sensor DHT22 mengumpulkan data suhu dan kelembaban.
- Data ini dikirim ke broker MQTT.
- Sebuah halaman web, dibangun dengan JavaScript, mengambil data dari broker MQTT dan menampilkannya secara real-time.

Setup Hardware

- Pin Sensor DHT22:
 1. VCC: Pin daya
 2. DATA: Pin data
 3. NC: Tidak terhubung (tidak digunakan)
 4. GND: Ground

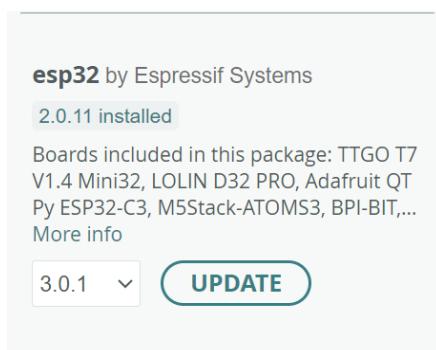
- Pin ESP32:
 1. 3V3 (atau 5V): Pin daya untuk sensor DHT22
 2. GPIO Pin: Pin data untuk sensor DHT22
 3. GND: Ground

Langkah-langkah Koneksi

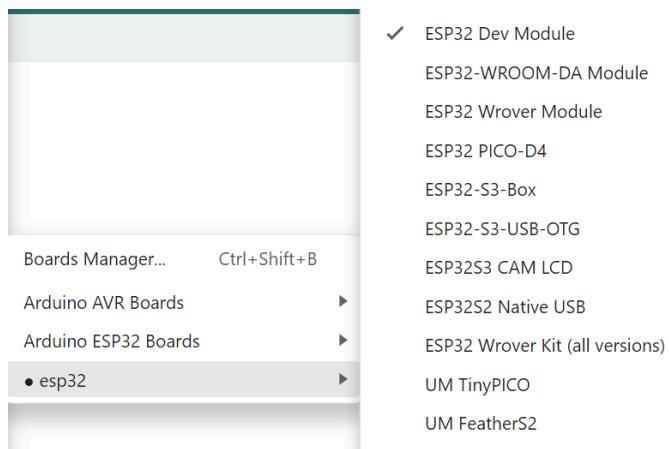
1. Hubungkan pin VCC pada sensor DHT22 ke pin 3V3 (atau 5V) pada ESP32
2. Hubungkan pin DATA pada sensor DHT22 ke pin GPIO yang sesuai pada ESP32
3. Hubungkan pin GND pada sensor DHT22 ke pin GND pada ESP32
4. Pin NC pada sensor DHT22 tidak perlu dihubungkan:

Arduino IDE Setup

1. Instalasi Arduino IDE
2. Mengatur Board di Arduino IDE
 - a. Buka Arduino IDE.
 - b. Klik pada Board Manager.
 - c. Di dalam Board Manager, cari dan instal board ESP32 oleh Espressif Systems.

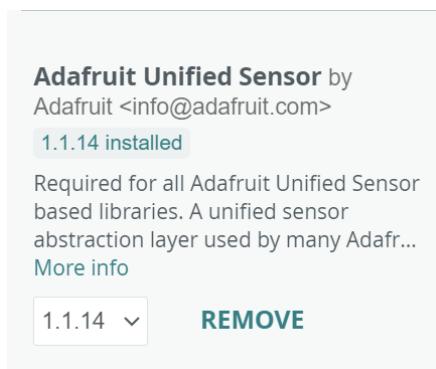


- d. Masuk ke tools, klik board esp32, pilih ESP32 DEV Module.

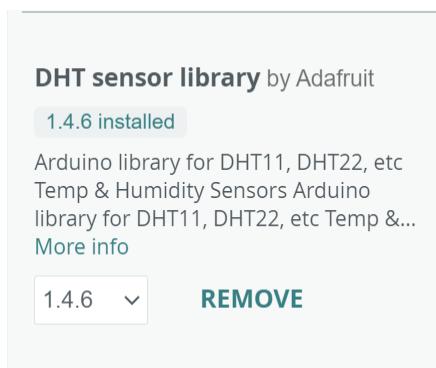


3. Menginstal Library yang Diperlukan

- Buka menu Sketch > Include Library > Manage Libraries.
- Install DHT sensor library yang dibuat oleh Adafruit.



- Install PubSubClient yang dibuat oleh Nick O'Leary.



Source Code

<https://github.com/zaqueen/FP-Integrasi-Sistem-5027221017-5027221023-/tree/main/Project-1>

Implementasi

1. Setup mosquitto.conf

```
# =====
# Listeners
# =====

# Listen on a port/ip address combination. By using this variable
# multiple times, mosquitto can listen on more than one port. If
# this variable is used and neither bind_address nor port given,
# then the default listener will not be started.
# The port number to listen on must be given. Optionally, an ip
# address or host name may be supplied as a second argument. In
# this case, mosquitto will attempt to bind the listener to that
# address and so restrict access to the associated network and
# interface. By default, mosquitto will listen on all interfaces.
# Note that for a websockets listener it is not possible to bind to a host
# name.
#
# On systems that support Unix Domain Sockets, it is also possible
# to create a Unix socket rather than opening a TCP socket. In
# this case, the port number should be set to 0 and a unix socket
# path must be provided, e.g.
# listener 0 /tmp/mosquitto.sock
#
# listener port-number [ip address/host name/unix socket path]
listener 1883
listener 9000
protocol websockets

# =====
# Persistence
# =====

# If persistence is enabled, save the in-memory database to disk
# every autosave_interval seconds. If set to 0, the persistence
# database will only be written when mosquitto exits. See also
# autosave_on_changes.
# Note that writing of the persistence database can be forced by
# sending mosquitto a SIGUSR1 signal.
#autosave_interval 1800

# If true, mosquitto will count the number of subscription changes, retained
# messages received and queued messages and if the total exceeds
# autosave_interval then the in-memory database will be saved to disk.
# If false, mosquitto will save the in-memory database to disk by treating
# autosave_interval as a time in seconds.
#autosave_on_changes false

# Save persistent message data to disk (true/false).
# This saves information about all messages, including
# subscriptions, currently in-flight messages and retained
# messages.
# retained_persistence is a synonym for this option.
persistence true

# The filename to use for the persistent database, not including
# the path.
#persistence_file mosquitto.db

# Location for persistent database.
# Default is an empty string (current directory).
# Set to e.g. /var/lib/mosquitto if running as a proper service on Linux or
# similar.
persistence_location /var/lib/mosquitto
```

```

# =====
# Security
# =====

# If set, only clients that have a matching prefix on their
# clientid will be allowed to connect to the broker. By default,
# all clients may connect.
# For example, setting "secure-" here would mean a client "secure-
# client" could connect but another with clientid "mqtt" couldn't.
#clientid_prefixes

# Boolean value that determines whether clients that connect
# without providing a username are allowed to connect. If set to
# false then a password file should be created (see the
# password_file option) to control authenticated client access.
#
# Defaults to false, unless there are no listeners defined in the configuration
# file, in which case it is set to true, but connections are only allowed from
# the local machine.
allow_anonymous true

pid_file /run/mosquitto/mosquitto.pid
log_dest file /var/log/mosquitto/mosquitto.log
include_dir /etc/mosquitto/conf.d

```

2. Setup file.ino untuk esp32

Menyesuaikan ssid, passwd, mqtt_server, topic, username, dan password

```

const char* ssid = "masihspotpanas";
const char* pswd = "mbokarep";

const char* mqtt_server = "152.42.194.14"; //Broker IP/URL
const char* topic = "/lab-kcks/room/temperature"; //Topic
const char* username="admin";
const char* password="kcks1029";

```

3. Setup html code untuk tampilan web

Menyesuaikan server, port, topic, mqtt username dan password

```

/** Write Your MQTT Settings Here Start*/
Server="152.42.194.14";
Port="9000";
Topic="/lab-kcks/room/temperature";
MQTTUserName="admin";
MQTTPassword="kcks1029";
/** Write Your MQTT Settings Here End **/

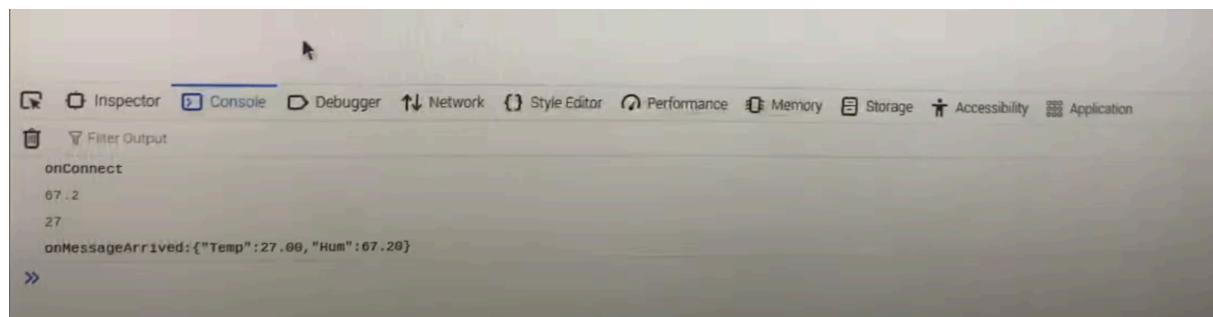
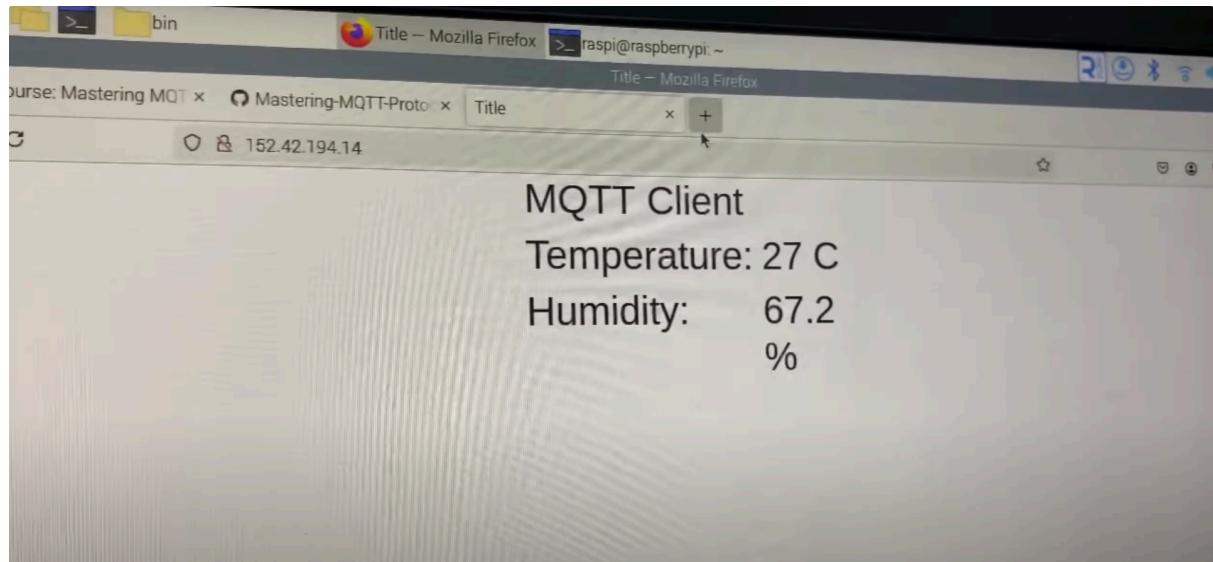

// Generate a random client ID
clientID = "clientID_" + parseInt(Math.random() * 100);

// Create a client instance
client=new Paho.MQTT.Client(Server,Number(Port),clientID);

```

Hasil

Setelah melakukan semua setup dan menjalankan code, berikut hasil tampilan pada halaman web:



Project 1 berhasil diimplementasikan dengan menggunakan sensor DHT22 dan mikrokontroler ESP32. Sensor berhasil membaca suhu dan kelembapan ruangan yang nilainya ditampilkan di halaman web HTML dan berkomunikasi menggunakan protokol MQTT.

PROJECT 2

Project ini melanjutkan implementasi yang sudah dilakukan pada project 1 dengan menambahkan fitur on/off LED.

Perangkat yang Digunakan

1. LED: Untuk memberikan indikator visual dan dapat dikendalikan melalui halaman web.

Fungsionalitas

- Halaman web akan memiliki tombol untuk menghidupkan dan mematikan LED.
- Perubahan status LED dikirim ke Raspberry Pi melalui broker MQTT.

Setup Hardware

- Pin LED
 1. Anoda (Panjang): Positif (+)
 2. Katoda (Pendek): Negatif (Ground)

Langkah-langkah koneksi

Koneksi LED ke Raspberry Pi:

- Anoda LED ke pin GPIO yang sesuai
- Katoda LED ke pin GND pada Raspberry Pi

Source Code

<https://github.com/zaqueen/FP-Integrasi-Sistem-5027221017-5027221023-/tree/main/Project-2>

Implementasi

1. Memastikan setup mosquito.conf yang sama dengan project 1 sudah diimplementasikan
2. Setup mqtt_sub.py

Menyesuaikan client.subscribe, username, password, dan connect

```
def on_connect(client, userdata, flags, rc):  
    # subscribe, which need to put into on_connect  
    # if reconnect after losing the connection with the broker, it will continue to subscribe to the raspberry/topic topic  
    client.subscribe("/lab-kcks/room/led")  
  
client = mqtt.Client()  
client.on_connect = on_connect  
client.on_message = on_message  
# create connection, the three parameters are broker address, broker port number, and keep-alive time respectively  
client.username_pw_set(username="admin",password="kcks1029")  
client.connect("152.42.194.14", 1883, 60)  
# set the network loop blocking, it will not actively end the program before calling disconnect() or the program crash  
client.loop_forever()
```

3. Memastikan setup file.ino yang sama dengan project 1 sudah diimplementasikan
4. Setup html code untuk tampilan web

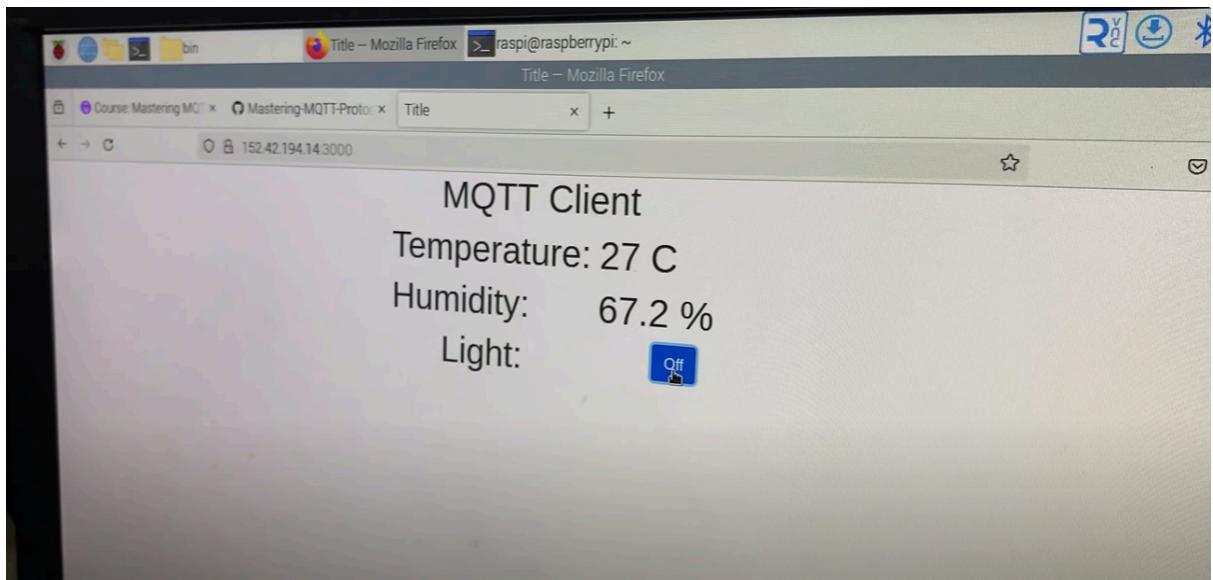
Menyesuaikan server, port, topic, subtopic, mqtt username dan password

```
/** Write Your MQTT Settings Here Start**/  
Server="152.42.194.14";  
Port="9000";  
Topic="/lab-kcks/room/temperature";  
LedTopic="/lab-kcks/room/led";  
MQTTUserName="admin";  
MQTTPassword="kcks1029";  
Connected=false;  
/** Write Your MQTT Settings Here End **/  
  
// Generate a random client ID  
clientID = "clientID_" + parseInt(Math.random() * 100);  
  
// Create a client instance  
client=new Paho.MQTT.Client(Server,Number(Port),clientID);
```

Hasil

Setelah melakukan semua setup dan menjalankan code, berikut hasil tampilan pada halaman web dan LED:

1. Ketika button “On” diklik



2. Ketika button "Off" diklik

