LAPORAN PRAKTIKUM

Kelompok	**:**	5
Nama Anggota	**:**	1. Alif Rizki Ananta Harahap 2. Andita Farah Salsabila 3. Hakim Asrori 4. Mali Nur Al Isthifa 5. Muhammad Ridzal Maulana 6. Rantika
Kelas	**:**	D3TI.2C
Mata Kuliah	**:**	**Internet Of Things**
Praktikum ke / Judul	**:**	4/ Komunikasi antar perangkat(Node)menggunakan komunikasi protocol MMQ
Tanggal Praktikum	**:**	20 September 2021
Dosen Pengampu	**:**	Ahmad Rifai, S.Tr.Kom., M.Tr.Kom

MODUL 5 – KOMUNIKASI ANTAR PERANGKAT (NODE) MENGGUNAKAN KOMUNIKASI PROTOKOL MQTT

1.1. Tujuan

- Mahasiswa mampu menginstall Broker MQTT.
- Mahasiswa mampu untuk mengimplementasikan protocol komunikasi MQTT
- Mahasiswa mampu untuk membuat dua perangkat NodeMCU untuk saling berkomunikasi menggunakan protokol komunikasi M2M.

1.2. Praktikum

1.2.1. Install Broker

Broker yang banyak digunakan dan free license yaitu Mosca Broker dan Mosquitto Broker.

- Mosca Broker dapat diperoleh di laman https://github.com/moscajs/mosca
- Mosquitto Broker dapat diperoleh di laman https://mosquitto.org/download

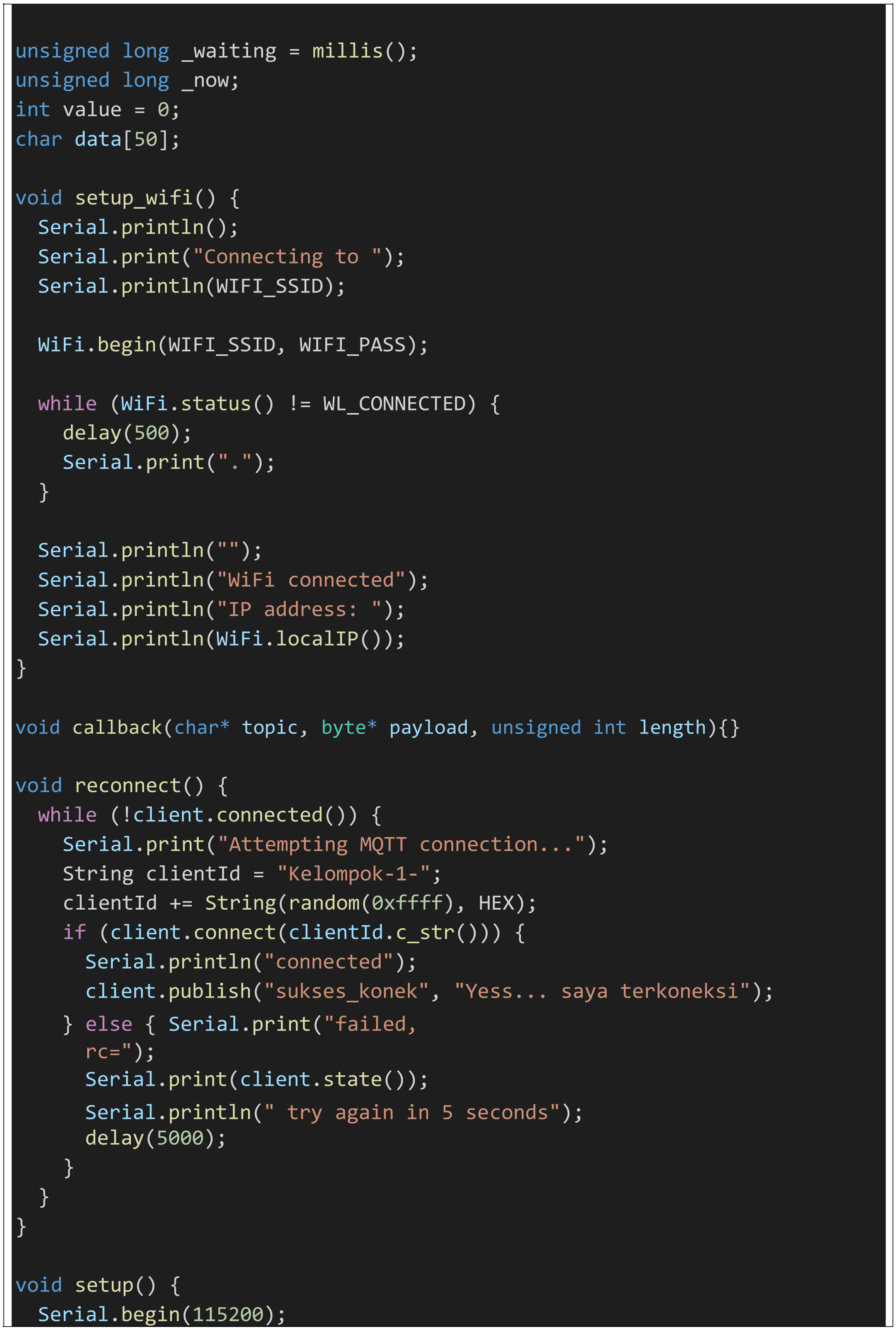
Silahkan anda lakukan instalasi salah satu dari dua pilihan tersebut dan jangan lupa untuk dituliskan setiap langkahnya berserta screenshoot.

1.2.2. Publish data

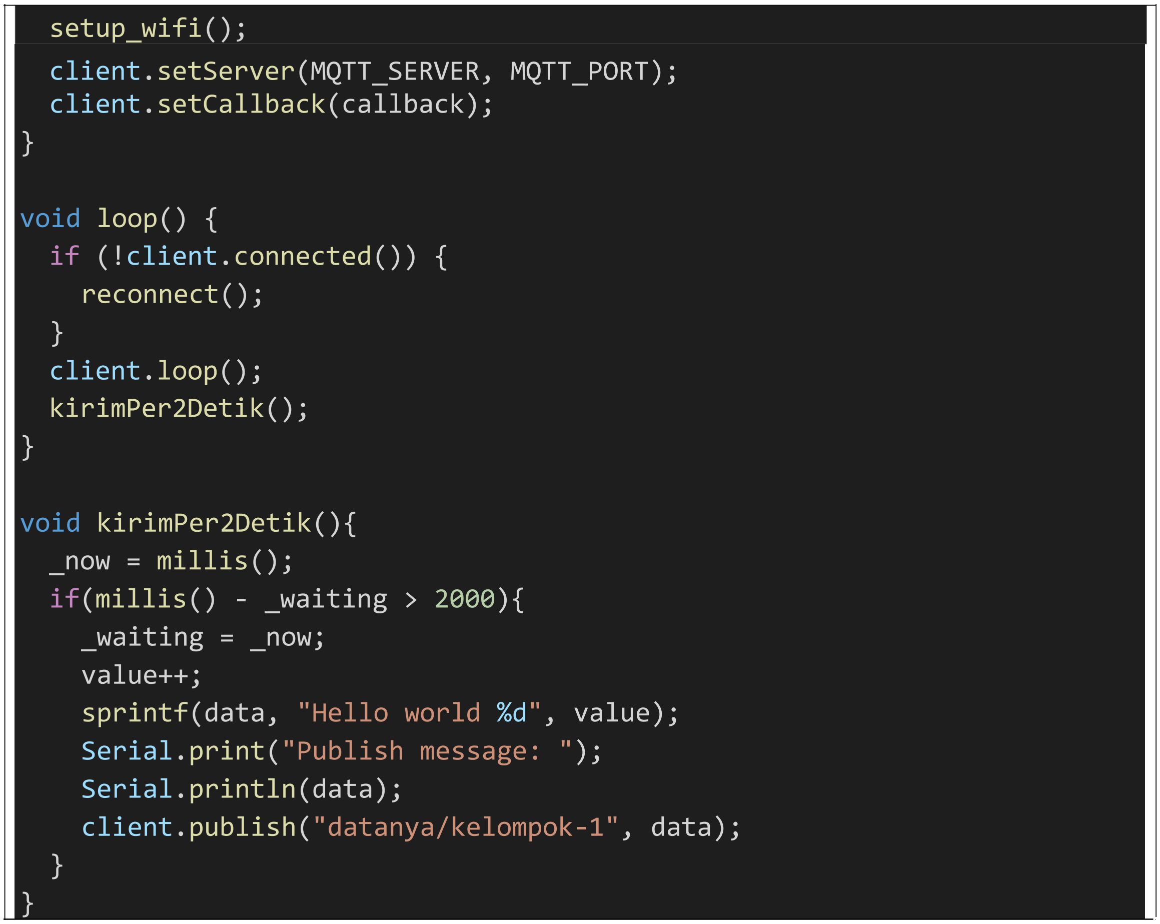
Dalam konsep komunikasi protokol MQTT mengirimkan suatu data dari perangkat Node 1 ke Node lainnya disebut publish data. Dalam konsep publish data terdapat dua istilah yaitu topik dan payload. Topik digunakan untuk mengidentifikasi identitas dari data. Sedangkan payload merupakan isi dari datanya. Berikut ini adalah contoh program mem-publish data.

Program Publish

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

WiFiClient espClient;
PubSubClient client(espClient);

#define WIFI_SSID     "NAMA_WIFI"
#define WIFI_PASS     "PASS_WIFI"
#define MQTT_SERVER   "broker.hivemq.com"
#define MQTT_PORT     1883
```

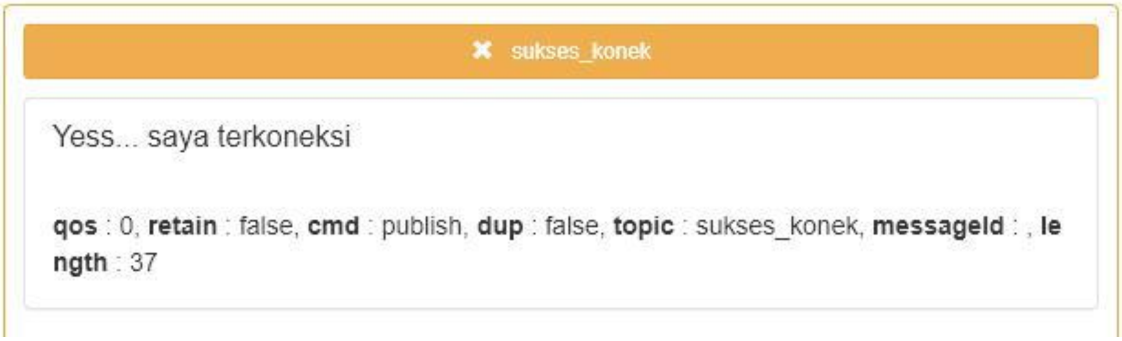


```
unsigned long _waiting = millis();
unsigned long _now;
int value = 0;
char data[50];

void setup_wifi() {
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(WIFI_SSID);

  WiFi.begin(WIFI_SSID, WIFI_PASS);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

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

void reconnect() {
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    String clientId = "Kelompok-1-";
    clientId += String(random(0xffff), HEX);
    if (client.connect(clientId.c_str())) {
      Serial.println("connected");
      client.publish("sukses_konek", "Yess... saya terkoneksi");
    } else { Serial.print("failed,
      rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}

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



```
  setup_wifi();
  client.setServer(MQTT_SERVER, MQTT_PORT);
  client.setCallback(callback);
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
  kirimPer2Detik();
}

void kirimPer2Detik(){
  _now = millis();
  if(millis() - _waiting > 2000){
    _waiting = _now;
    value++;
    sprintf(data, "Hello world %d", value);
    Serial.print("Publish message: ");
    Serial.println(data);
    client.publish("datanya/kelompok-1", data);
  }
}
```

Berikut ini adalah tampilan apabila berhasil terhubung ke MQTT Broker.

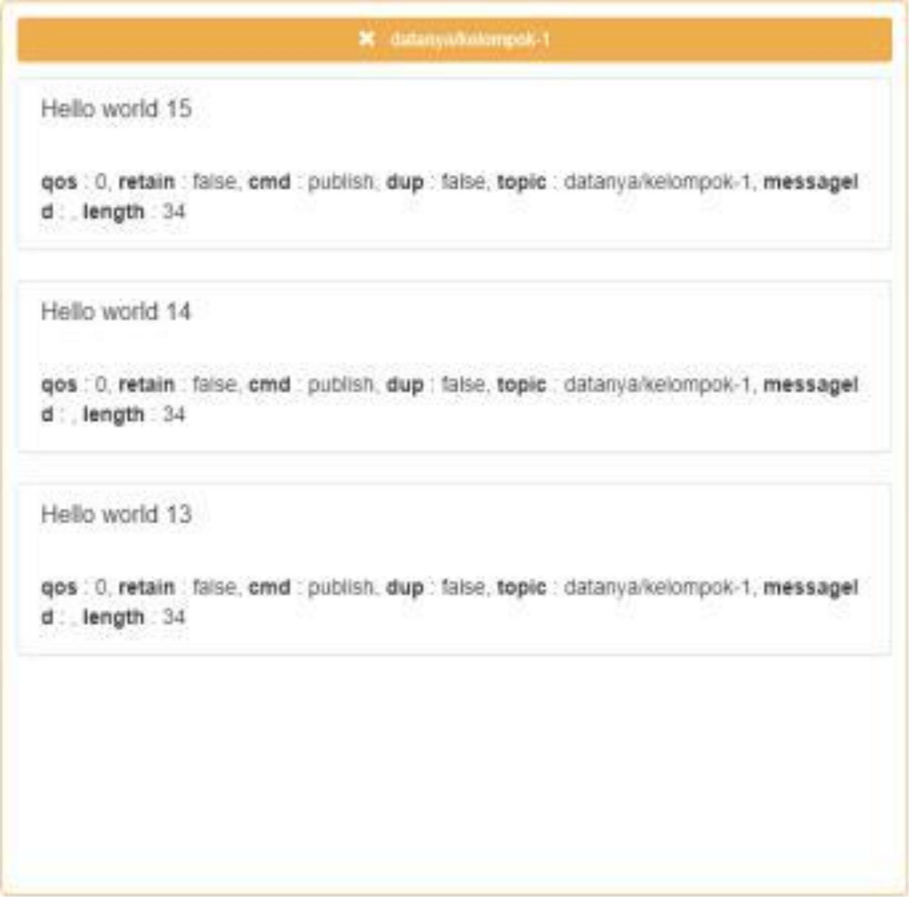


Yess... saya terkoneksi

qos : 0, **retain** : false, **cmd** : publish, **dup** : false, **topic** : sukses_konek, **messageId** : , **le
ngth** : 37

Gambar 1. Berhasil terhubung ke MQTT Broker

Berikut ini adalah tampilan di software MQTT Box dari data yang publish oleh NodeMCU.

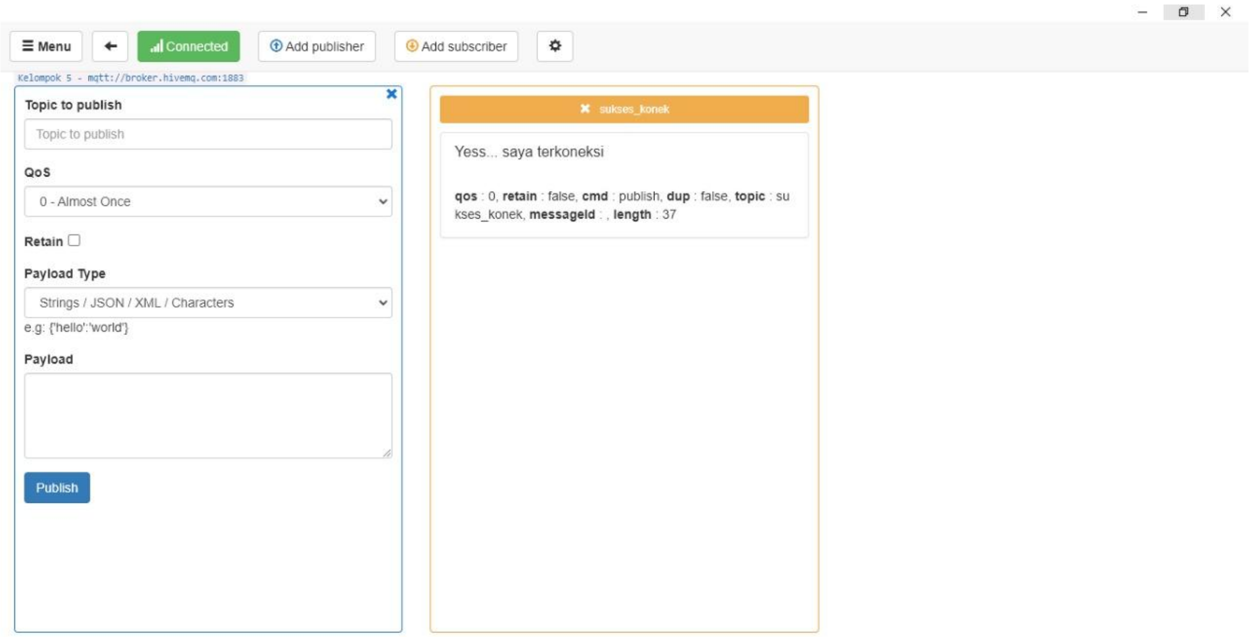


```
Hello world 15

qos : 0, retain : false, cmd : publish, dup : false, topic : datanya/kelompok-1, messageId : , length : 34

Hello world 14

qos : 0, retain : false, cmd : publish, dup : false, topic : datanya/kelompok-1, messageId : , length : 34

Hello world 13

qos : 0, retain : false, cmd : publish, dup : false, topic : datanya/kelompok-1, messageId : , length : 34
```

Gambar 2. Data yang dipublish oleh NodeMCU

LATIHAN 1.22

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

WiFiClient espClient;
PubSubClient client(espClient);

#define WIFI_SSID "METHA"
#define WIFI_PASS "kupukupu"
#define MQTT_SERVER "broker.hivemq.com"
#define MQTT_PORT 1883

unsigned long _waiting = millis();
unsigned long _now;
int value = 0;
char data[50];

void setup_wifi() {
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(WIFI_SSID);
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);

    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

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

void reconnect() {
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    String clientId = "Kelompok-1-";
    clientId += String(random(0xffff), HEX);
    if (client.connect(clientId.c_str())) {
      Serial.println("connected");
      client.publish("sukses_konek", "Yess... saya terkoneksi");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}

void setup() {
  Serial.begin(115200);
  setup_wifi();
  client.setServer(MQTT_SERVER, MQTT_PORT);
  client.setCallback(callback);
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
  kirimPer2Detik();
}

void kirimPer2Detik(){
  _now = millis();
  if(millis() - _waiting > 2000){
    _waiting = _now;
    value++;
    sprintf(data, "Hello world %d", value);
    Serial.print("Publish message: ");
    Serial.println(data);
    client.publish("datanya/kelompok-5", data);
  }
}
```

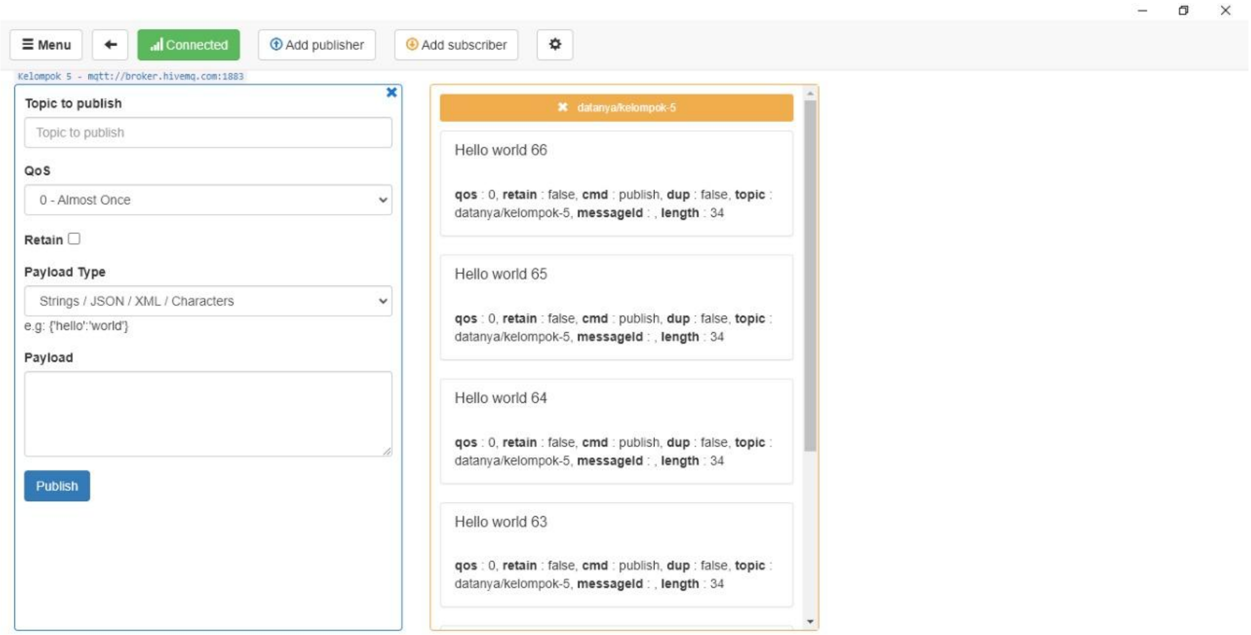


Kelompok 5 - mqtt://broker.hivemq.com:1883

Topic to publish: Topic to publish
QoS: 0 - Almost Once
Retain □
Payload Type: Strings / JSON / XML / Characters
e.g: {'hello':'world'}
Payload

sukses_konek

```
Yess... saya terkoneksi

qos : 0, retain : false, cmd : publish, dup : false, topic : sukses_konek, messageId : , length : 37
```

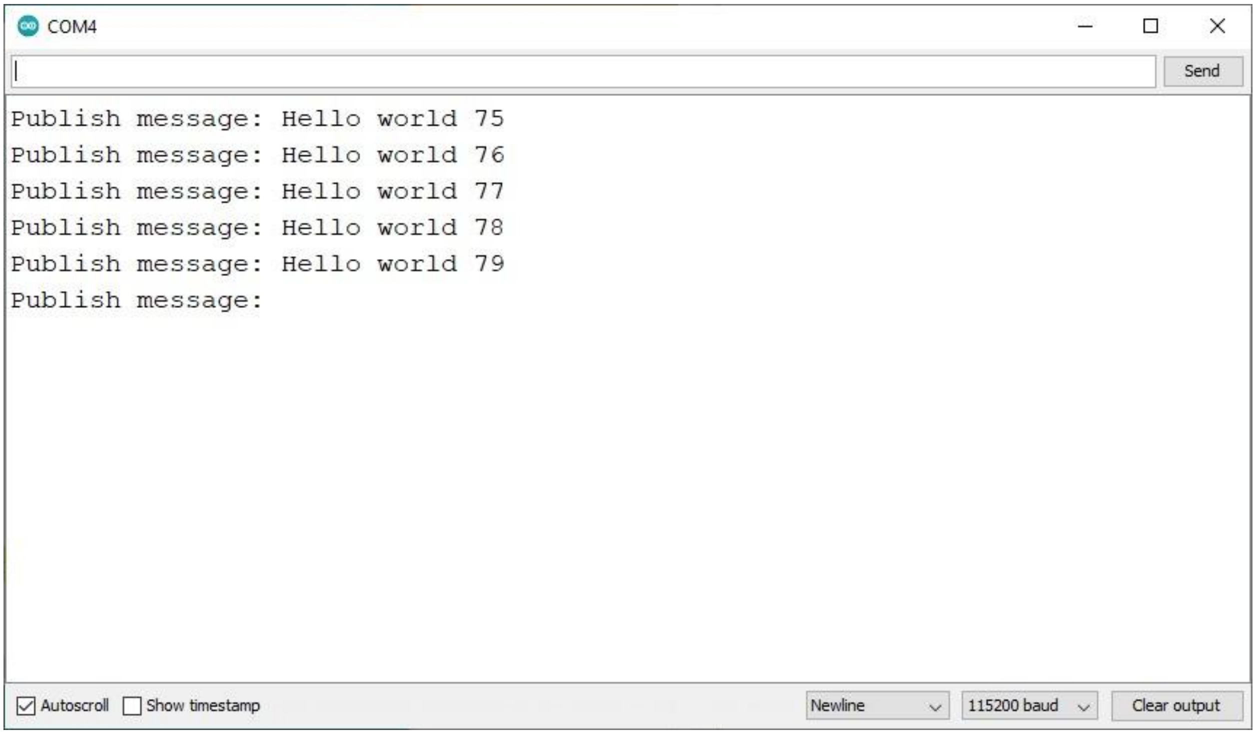


Kelompok 5 - mqtt://broker.hivemq.com:1883

Topic to publish: Topic to publish
QoS: 0 - Almost Once
Retain □
Payload Type: Strings / JSON / XML / Characters
e.g: {'hello':'world'}
Payload

datanya/kelompok-5

```
Hello world 66

qos : 0, retain : false, cmd : publish, dup : false, topic : datanya/kelompok-5, messageId : , length : 34

Hello world 65

qos : 0, retain : false, cmd : publish, dup : false, topic : datanya/kelompok-5, messageId : , length : 34

Hello world 64

qos : 0, retain : false, cmd : publish, dup : false, topic : datanya/kelompok-5, messageId : , length : 34

Hello world 63

qos : 0, retain : false, cmd : publish, dup : false, topic : datanya/kelompok-5, messageId : , length : 34
```



```
Publish message: Hello world 75
Publish message: Hello world 76
Publish message: Hello world 77
Publish message: Hello world 78
Publish message: Hello world 79
Publish message:
```

Kesimpulan : nodemcu akan mengirimkan atau publish data ke mqtt public seperti contoh yaitu mqtt milik hivemq. Dengan topic “sukses_konek” dan juga “datanya/kelompok5”.

1.2.3. Subscribe data

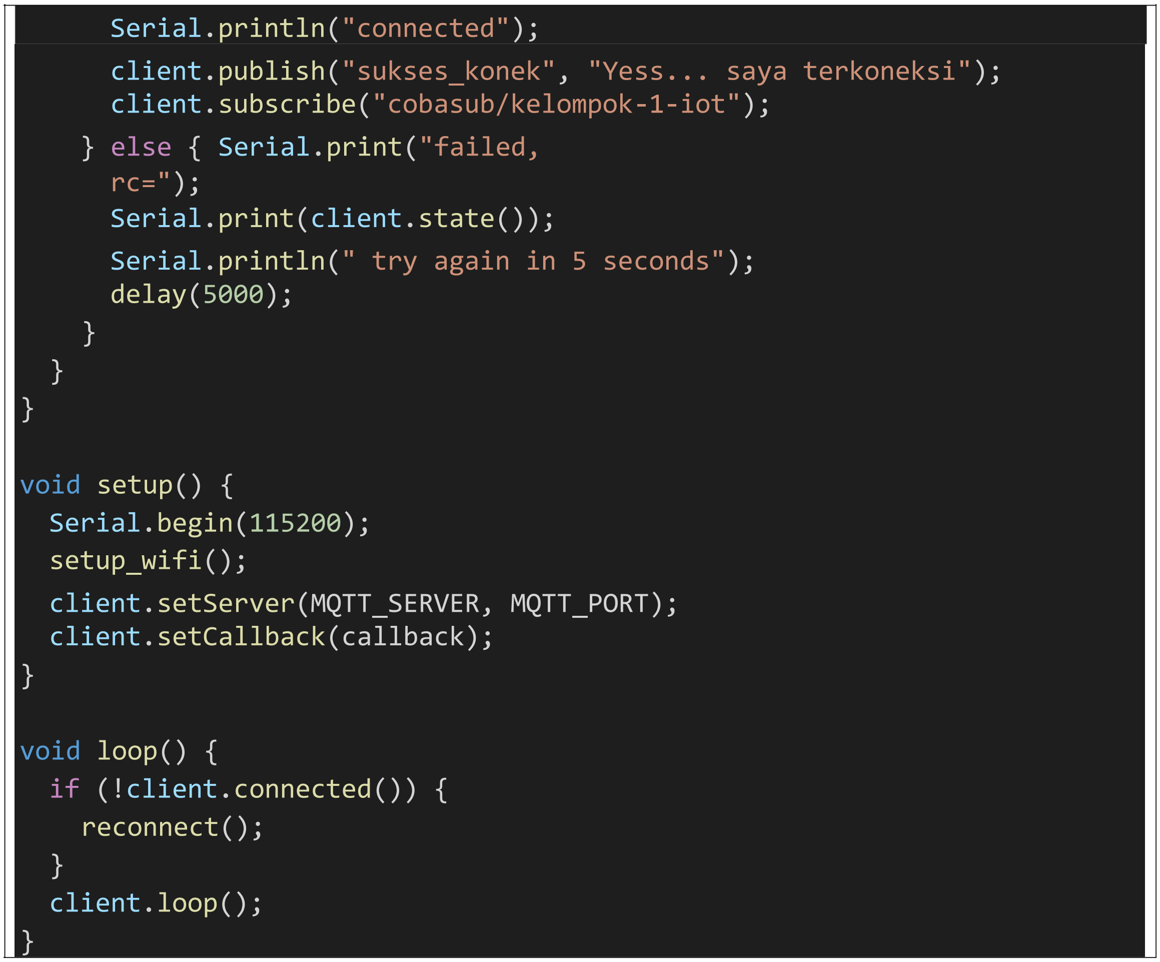
Subscribe merupakan proses menerima pesan/data yang disukai (sesuai topic) dari perangkat Node lainnya. Proses ini akan mendefinisikan Node yang melakukan subscribe akan secara realtime mendapatkan data yang dikirimkan oleh Node publisher selama kedua perangkat tersebut masih terhubung ke Broker yang sama. Keuntungannya disini Node subscriber tidak perlu melakukan request terlebih dahulu ke pengirim data seperti halnya yang telah dilakukan dengan menggunakan protocol HTTP. Berikut ini adalah program NodeMCU sebagai subscriber.

Program Subscribe

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
```

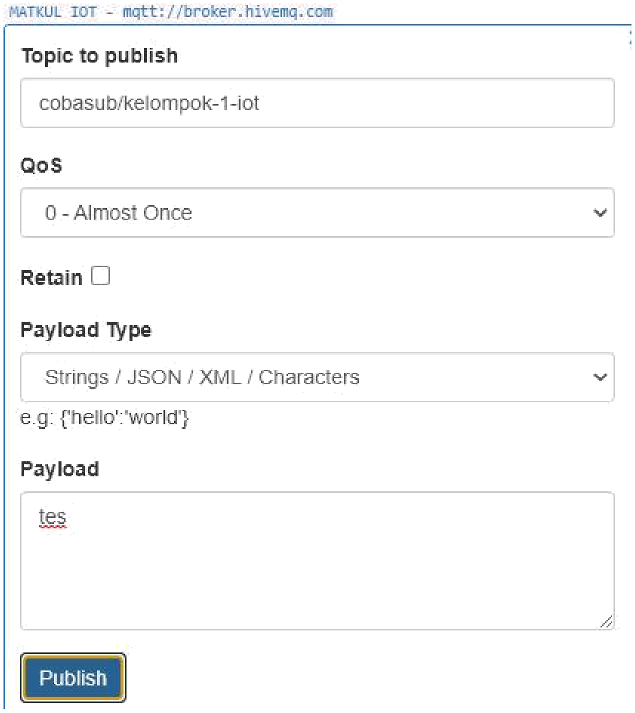
[redacted]

```
WiFiClient espClient;
PubSubClient client(espClient);

#define WIFI_SSID     "NAMA_WIFI"
#define WIFI_PASS     "PASS_WIFI"
#define MQTT_SERVER   "broker.hivemq.com"
#define MQTT_PORT     1883

String buffData;

void setup_wifi() {
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(WIFI_SSID);

  WiFi.begin(WIFI_SSID, WIFI_PASS);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* payload, unsigned int
  length){ buffData = "";
  for(int i=0; i<length; i++){
    buffData += (char) payload[i];
  }

  Serial.print(topic);
  Serial.print(" ==> ");
  Serial.println(buffData);
}

void reconnect() {
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    String clientId = "Kelompok-1-";
    clientId += String(random(0xffff), HEX);
    if (client.connect(clientId.c_str())) {
```



```
      Serial.println("connected");
      client.publish("sukses_konek", "Yess... saya terkoneksi");
      client.subscribe("cobasub/kelompok-1-iot");
    } else { Serial.print("failed,
      rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}

void setup() {
  Serial.begin(115200);
  setup_wifi();
  client.setServer(MQTT_SERVER, MQTT_PORT);
  client.setCallback(callback);
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
}
```

Berikut ini percobaan publish data dari MQTT Box dan diterima oleh NodeMCU.



MATKUL IOT - mqtt://broker.hivemq.com

Topic to publish: cobasub/kelompok-1-iot

QoS: 0 - Almost Once

Retain □

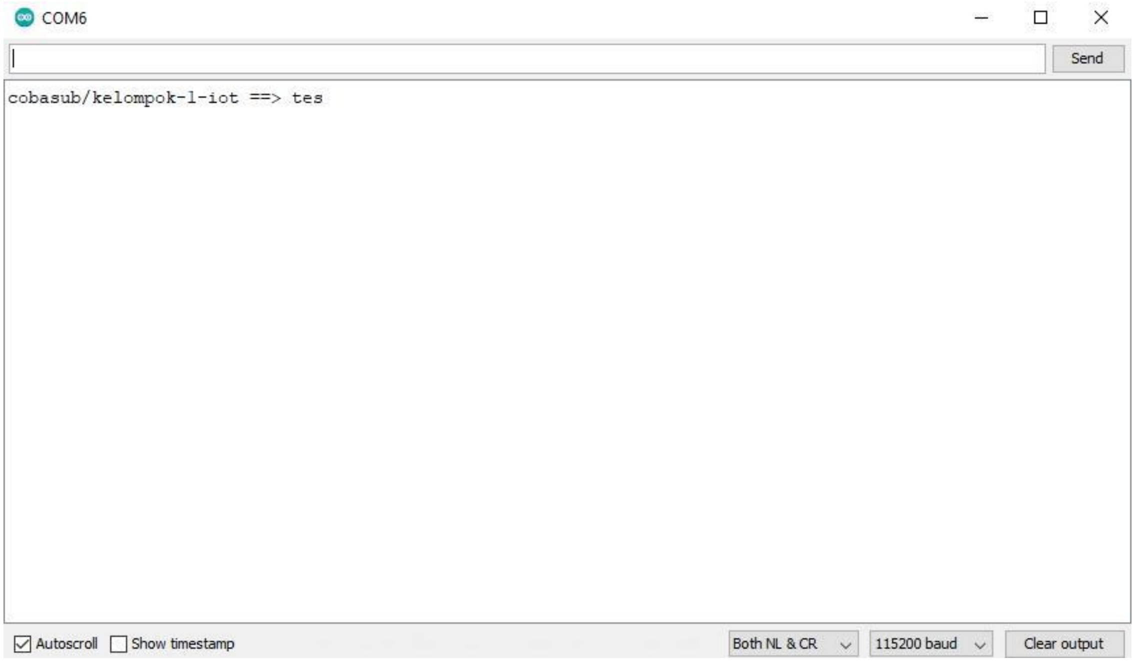
Payload Type: Strings / JSON / XML / Characters

e.g: {'hello':'world'}

Payload: tes

Gambar 3. Publish data melalui MQTT Box

Data yang diterima oleh NodeMCU dan ditempilkan pada serial monitor.



```
cobasub/kelompok-1-iot ==> tes
```

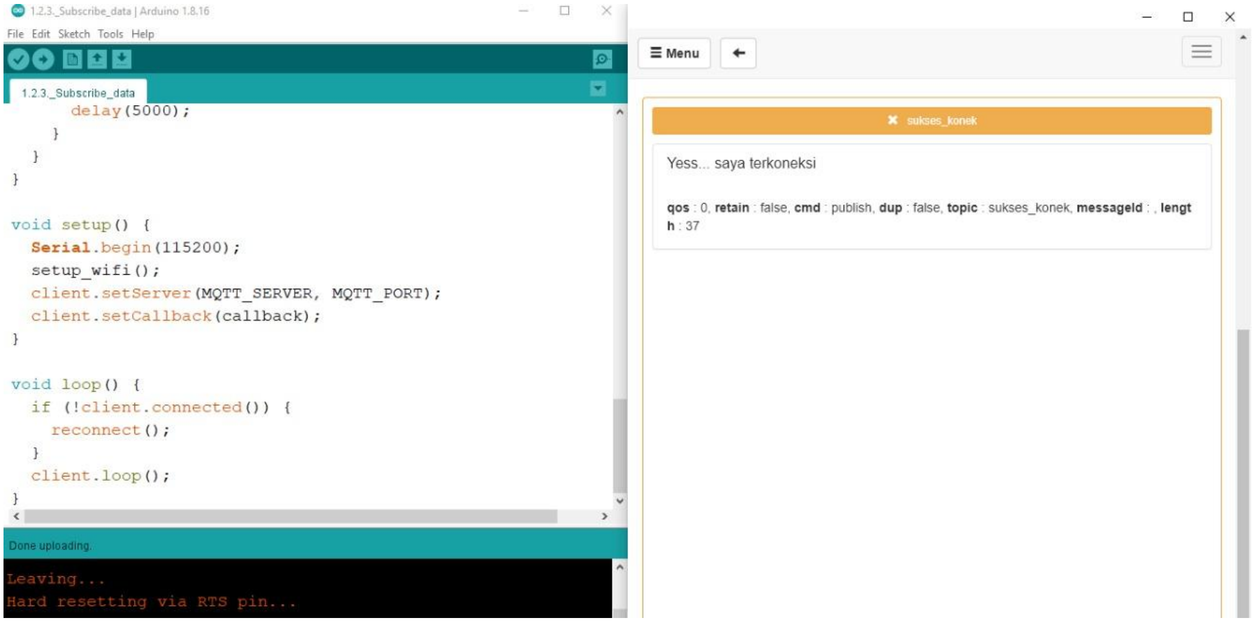
☑ Autoscroll □ Show timestamp

Both NL & CR

115200 baud

Gambar 4. NodeMCU menerima data

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

WiFiClient espClient;
PubSubClient client(espClient);

#define WIFI_SSID "METHA"
#define WIFI_PASS "kupukupu"
#define MQTT_SERVER "broker.hivemq.com"
#define MQTT_PORT 1883

String buffData;

void setup_wifi() {
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(WIFI_SSID);
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* payload, unsigned int length){
  buffData = "";
  for(int i=0; i<length; i++){
    buffData += (char) payload[i];
  }

  Serial.print(topic);
  Serial.print(" ==> ");
  Serial.println(buffData);
}

void reconnect() {
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    String clientId = "Kelompok-5-";
    clientId += String(random(0xffff), HEX);
    if (client.connect(clientId.c_str())) {
      Serial.println("connected");
      client.publish("sukses_konek", "Yess... saya terkoneksi");
      client.subscribe("cobasub/kelompok-5-iot");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}

void setup() {
  Serial.begin(115200);
  setup_wifi();
  client.setServer(MQTT_SERVER, MQTT_PORT);
  client.setCallback(callback);
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
}
```


1.2.3._Subscribe_data | Arduino 1.8.16
File Edit Sketch Tools Help
1.2.3._Subscribe_data
≡ Menu
sukses_konek

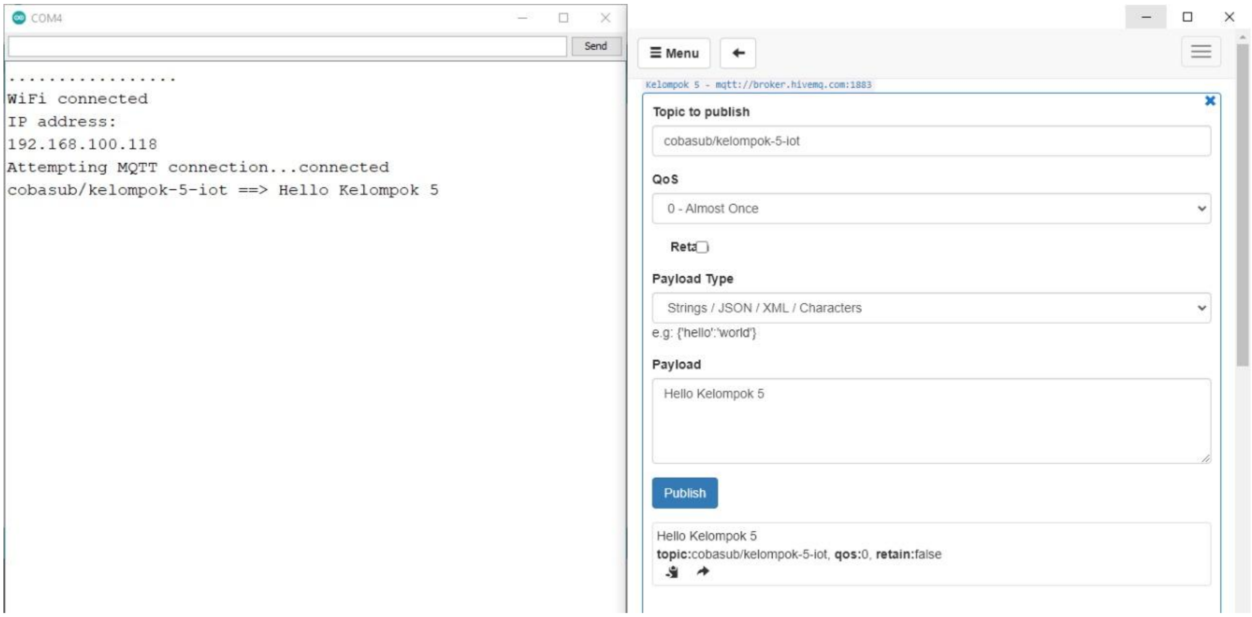
```
      delay(5000);
    }
  }
}

void setup() {
  Serial.begin(115200);
  setup_wifi();
  client.setServer(MQTT_SERVER, MQTT_PORT);
  client.setCallback(callback);
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
}
```

Done uploading.

```
Leaving...
Hard resetting via RTS pin...
```

Yess... saya terkoneksi

qos : 0, **retain** : false, **cmd** : publish, **dup** : false, **topic** : sukses_konek, **messageId** : , **length** : 37



```
.................
WiFi connected
IP address:
192.168.100.118
Attempting MQTT connection...connected
cobasub/kelompok-5-iot ==> Hello Kelompok 5
```

Topic to publish: cobasub/kelompok-5-iot

QoS: 0 - Almost Once

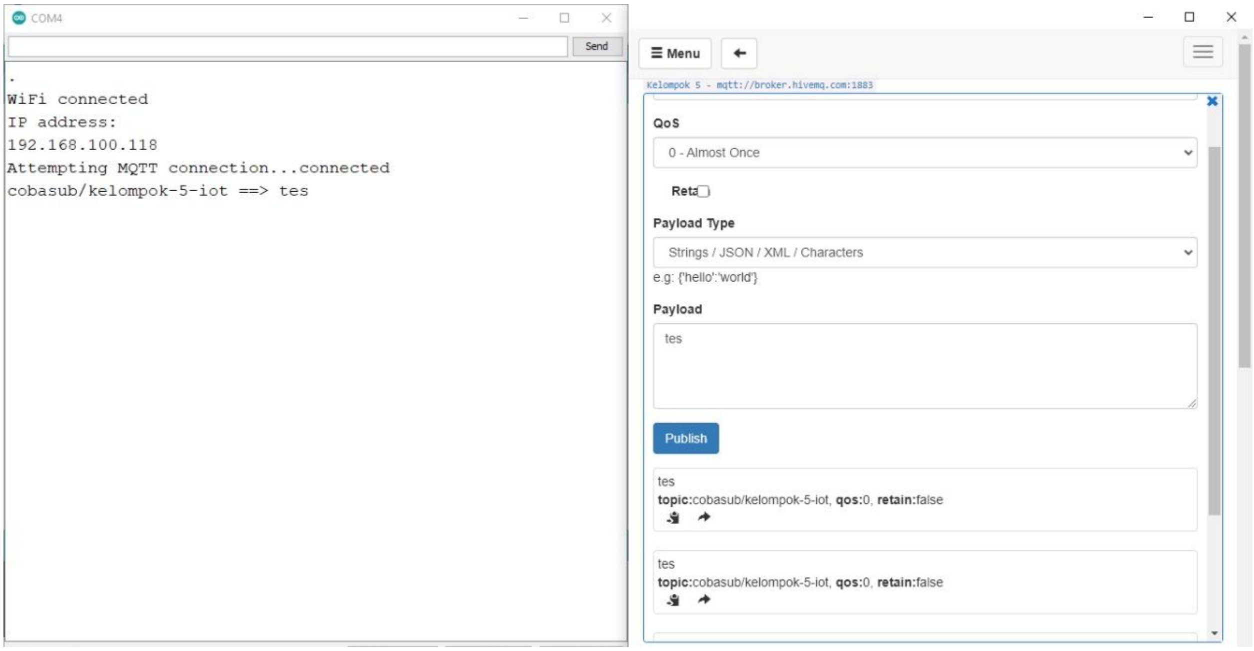
Reta □

Payload Type: Strings / JSON / XML / Characters

e.g: {'hello':'world'}

Payload: Hello Kelompok 5

Hello Kelompok 5
topic:cobasub/kelompok-5-iot, **qos:**0, **retain:**false



```
.
WiFi connected
IP address:
192.168.100.118
Attempting MQTT connection...connected
cobasub/kelompok-5-iot ==> tes
```

QoS: 0 - Almost Once

Reta □

Payload Type: Strings / JSON / XML / Characters

e.g: {'hello':'world'}

Payload: tes

tes
topic:cobasub/kelompok-5-iot, **qos:**0, **retain:**false

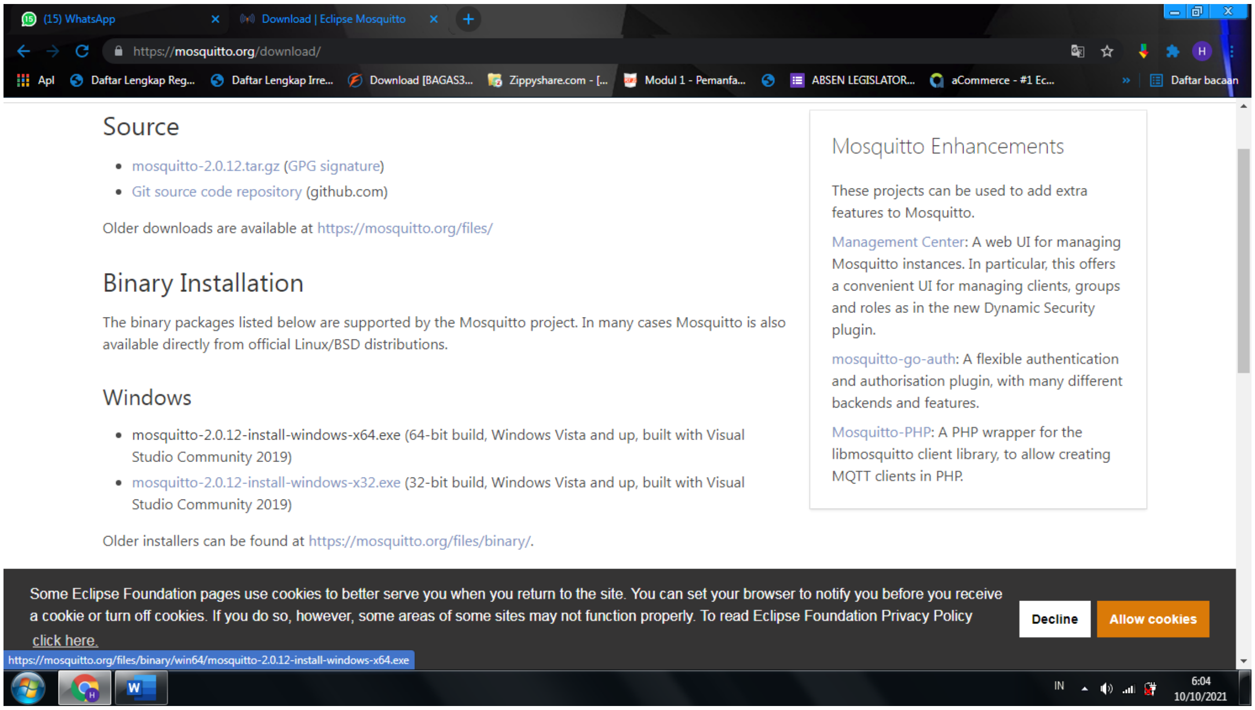
tes
topic:cobasub/kelompok-5-iot, **qos:**0, **retain:**false

Kesimpulan : Nodemcu akan menerrima data atau subscribe contojnya seperti diatas, melalui mqtt broker publik dengan topik “cobasub/kelompok-5-iot”

1.3. Tugas

1. Screenshoot setiap langkah-langkah installasi/persiapan Broker MQTT pilih salah satu (Mosca, Mosquitto, EMQX) secara lokal (di komputer/PC/Laptop masing-masing).

   1.1. Disini kami menggunakan aplikasi mosquitto, jadi yang pertama download mosquitto terlebih dahulu di https://mosquitto.org/download/



Source

- mosquitto-2.0.12.tar.gz (GPG signature)
- Git source code repository (github.com)

Older downloads are available at https://mosquitto.org/files/

Binary Installation

The binary packages listed below are supported by the Mosquitto project. In many cases Mosquitto is also available directly from official Linux/BSD distributions.

Windows

- mosquitto-2.0.12-install-windows-x64.exe (64-bit build, Windows Vista and up, built with Visual Studio Community 2019)
- mosquitto-2.0.12-install-windows-x32.exe (32-bit build, Windows Vista and up, built with Visual Studio Community 2019)

Older installers can be found at https://mosquitto.org/files/binary/.

Mosquitto Enhancements

These projects can be used to add extra features to Mosquitto.

Management Center: A web UI for managing Mosquitto instances. In particular, this offers a convenient UI for managing clients, groups and roles as in the new Dynamic Security plugin.

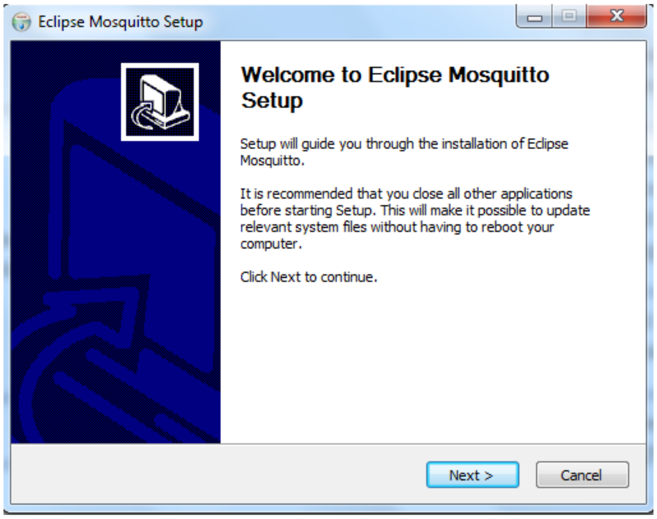
mosquitto-go-auth: A flexible authentication and authorisation plugin, with many different backends and features.

Mosquitto-PHP: A PHP wrapper for the libmosquitto client library, to allow creating MQTT clients in PHP.

Some Eclipse Foundation pages use cookies to better serve you when you return to the site. You can set your browser to notify you before you receive a cookie or turn off cookies. If you do so, however, some areas of some sites may not function properly. To read Eclipse Foundation Privacy Policy click here.

   1.2. Setelah didownload, lalu kita double klik untuk menjalankan installernya.

   1.3. Setelah itu pilih next



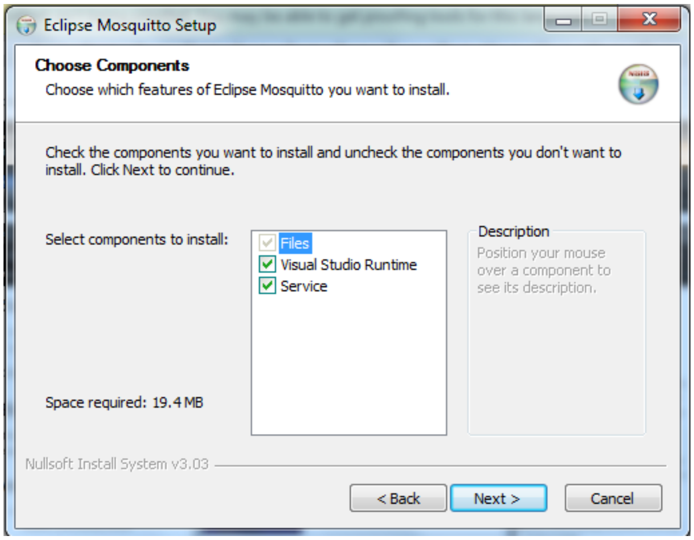
Welcome to Eclipse Mosquitto Setup

Setup will guide you through the installation of Eclipse Mosquitto.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

   1.4. Centang semua komponen, Setelah itu pilih next



Choose Components

Choose which features of Eclipse Mosquitto you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

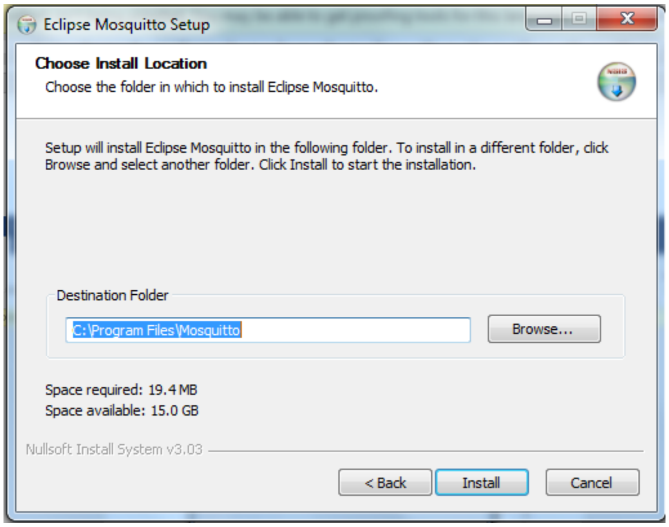
☑ Files
☑ Visual Studio Runtime
☑ Service

Description
Position your mouse over a component to see its description.

Space required: 19.4 MB

Nullsoft Install System v3.03

1.5. Setelah itu pilih next



Choose Install Location

Choose the folder in which to install Eclipse Mosquitto.

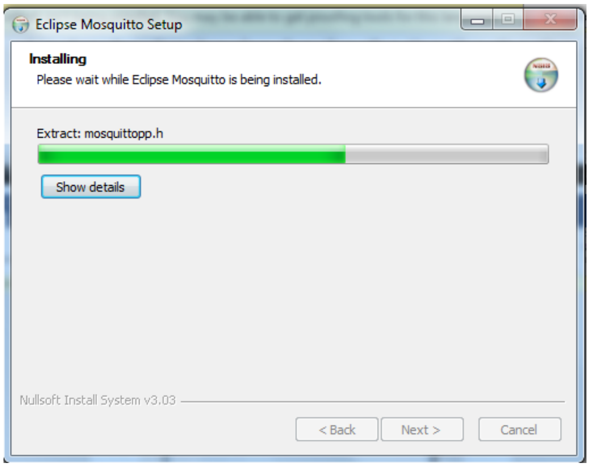
Setup will install Eclipse Mosquitto in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder
C:\Program Files\Mosquitto

Space required: 19.4 MB
Space available: 15.0 GB

Nullsoft Install System v3.03

1.6. Berikut proses installasi nya



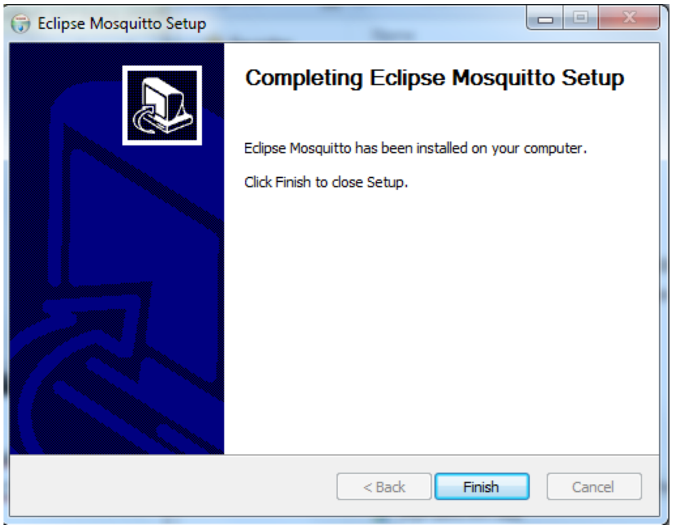
Installing

Please wait while Eclipse Mosquitto is being installed.

Extract: mosquittopp.h

Nullsoft Install System v3.03

1.7. Setelah itu pilih finish



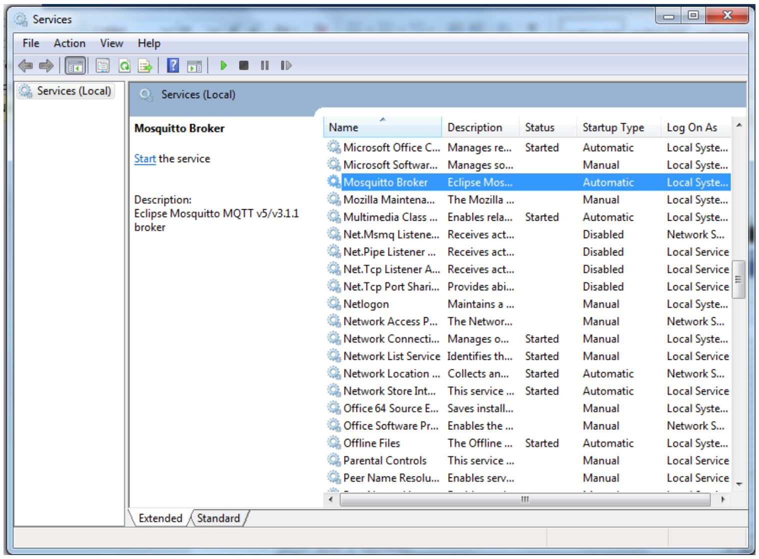
Eclipse Mosquitto Setup

Completing Eclipse Mosquitto Setup

Eclipse Mosquitto has been installed on your computer.

Click Finish to close Setup.

1.8. Setelah itu start mosquitto broker di services.msc



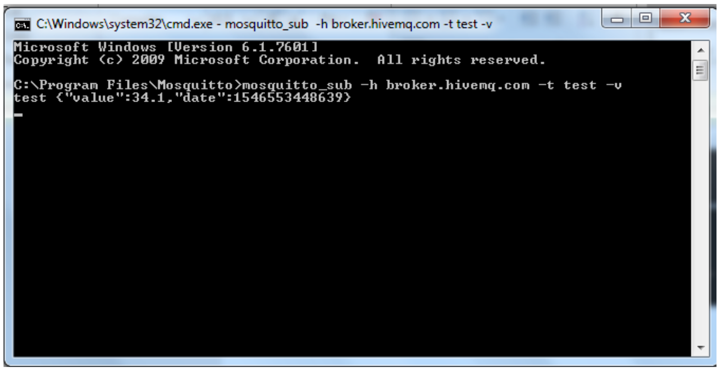
Mosquitto Broker

Start the service

Description:
Eclipse Mosquitto MQTT v5/v3.1.1 broker

Name	Description	Status	Startup Type	Log On As
Microsoft Office C...	Manages re...	Started	Automatic	Local Syste...
Microsoft Softwar...	Manages so...		Manual	Local Syste...
Mosquitto Broker	Eclipse Mos...		Automatic	Local Syste...
Mozilla Maintena...	The Mozilla ...		Manual	Local Syste...
Multimedia Class ...	Enables rela...	Started	Automatic	Local Syste...
Net.Msmq Listene...	Receives act...		Disabled	Network S...
Net.Pipe Listener ...	Receives act...		Disabled	Local Service
Net.Tcp Listener A...	Receives act...		Disabled	Local Service
Net.Tcp Port Shari...	Provides abi...		Disabled	Local Service
Netlogon	Maintains a ...		Manual	Local Syste...
Network Access P...	The Networ...		Manual	Network S...
Network Connecti...	Manages o...	Started	Manual	Local Syste...
Network List Service	Identifies th...	Started	Manual	Local Service
Network Location ...	Collects an...	Started	Automatic	Network S...
Network Store Int...	This service ...	Started	Automatic	Local Service
Office 64 Source E...	Saves install...		Manual	Local Syste...
Office Software Pr...	Enables the ...		Manual	Network S...
Offline Files	The Offline ...	Started	Automatic	Local Syste...
Parental Controls	This service ...		Manual	Local Service
Peer Name Resolu...	Enables serv...		Manual	Local Service

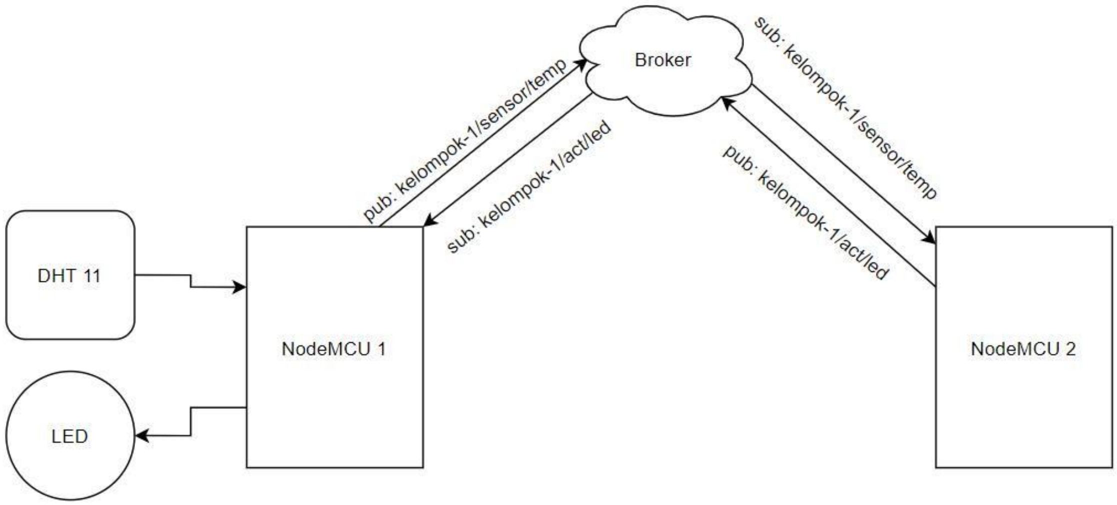
1.9. Hasil Installasinya



C:\Windows\system32\cmd.exe - mosquitto_sub -h broker.hivemq.com -t test -v

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Program Files\Mosquitto>mosquitto_sub -h broker.hivemq.com -t test -v
test {"value":34.1,"date":1546553448639}
_
```

2. Melakukan hal yang sama seperti praktikum sebelumnya yaitu membuat dua perangkat NodeMCU untuk saling berkomunikasi. Berikut adalah diagram bloknya.



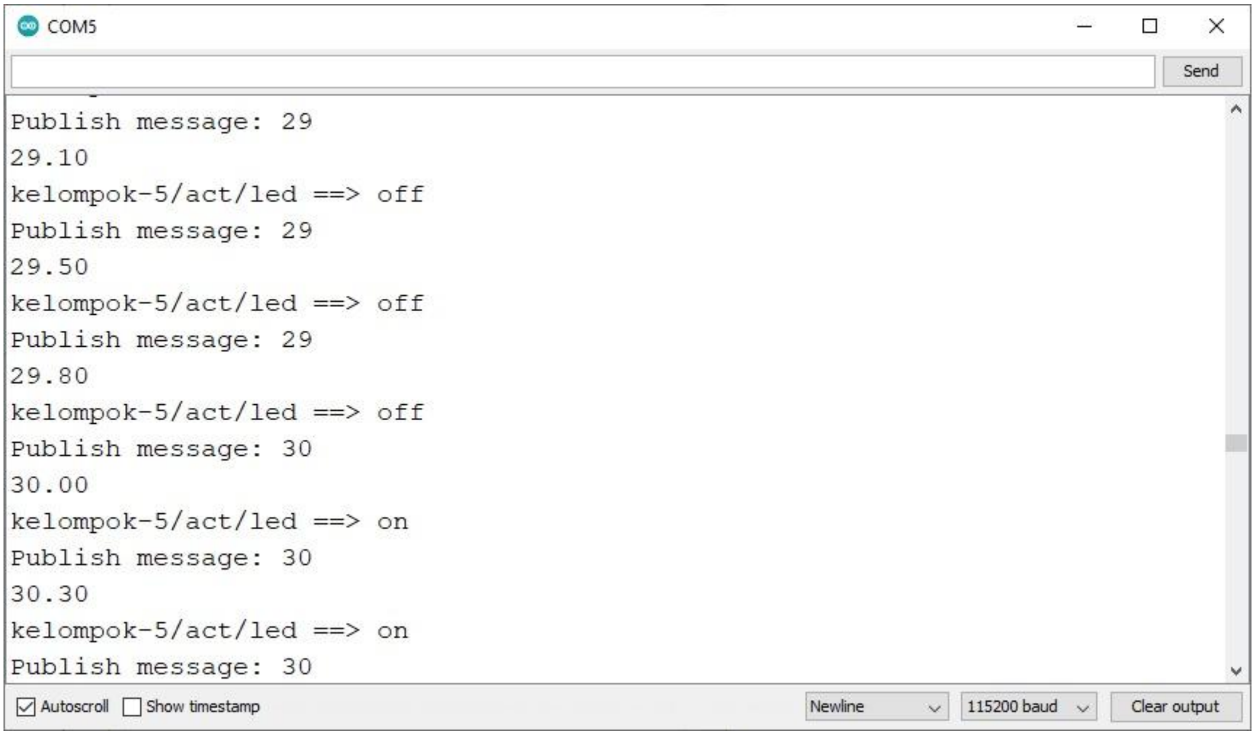
NodeMCU 1 bertugas sebagai pengambil data sensor dan terhubung dengan LED. Sedangkan NodeMCU 2 bertugas sebagai pemroses data yang dikirimkan oleh NodeMCU 1 melalui jaringan dengan protokol MQTT. Berikut ini adalah Rule dari kasus tersebut.

No	Sensor DHT	LED
1	Temperature > 29	ON
2	Temperature <= 29	OFF

Jawab

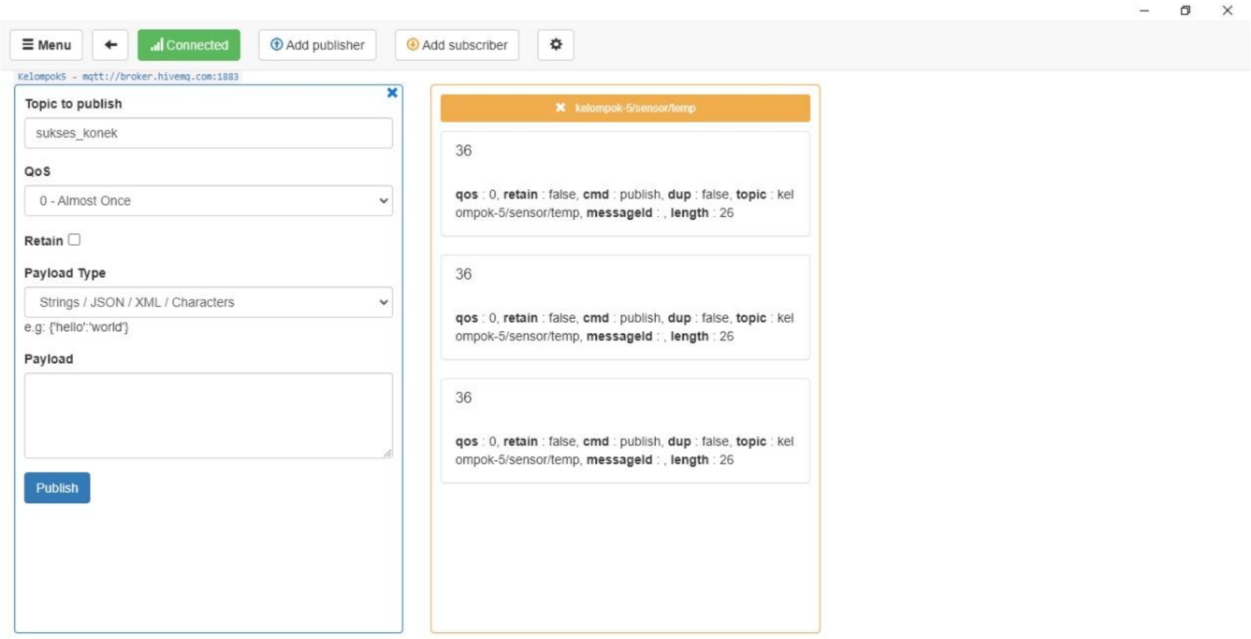
```
#SERVER
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include "DHT.h"

WiFiClient espClient;
PubSubClient client(espClient);

#define WIFI_SSID "METHA"
#define WIFI_PASS "kupukupu"
#define MQTT_SERVER "broker.hivemq.com"
#define MQTT_PORT 1883
#define DHTPIN D1
#define DHTTYPE DHT11
#define LED D2

unsigned long _waiting = millis();
unsigned long _now;
int value = 0;
char data[50];

DHT dht(DHTPIN, DHTTYPE);

String buffData;

void setup_wifi() {
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(WIFI_SSID);
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* payload, unsigned int length){
  buffData = "";
  for(int i=0; i<length; i++){
    buffData += (char) payload[i];
  }

  Serial.print(topic);
  Serial.print(" ==> ");
  Serial.println(buffData);

  if (buffData == "on") {
    digitalWrite(LED, HIGH);
  }

  if (buffData == "off") {
    digitalWrite(LED, LOW);
  }
}

void reconnect() {
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    String clientId = "Kelompok-5-";
    clientId += String(random(0xffff), HEX);
    if (client.connect(clientId.c_str())) {
      Serial.println("connected");
      client.publish("sukses_konek", "Yess... saya terkoneksi");
      client.subscribe("kelompok-5/act/led");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}

void setup() {
  Serial.begin(115200);
  setup_wifi();
  pinMode(LED, OUTPUT);
  client.setServer(MQTT_SERVER, MQTT_PORT);
  client.setCallback(callback);
  dht.begin();
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
  kirimPer2Detik();
}

void kirimPer2Detik(){
  _now = millis();
  if(millis() - _waiting > 2000){
    _waiting = _now;
    float temperature = dht.readTemperature();
    itoa(temperature, data, 10);
    Serial.print("Publish message: ");
    Serial.println(data);
    Serial.println(temperature);
    client.publish("kelompok-5/sensor/temp", data);
  }
}
```

#Sisi Server



```
Publish message: 29
29.10
kelompok-5/act/led ==> off
Publish message: 29
29.50
kelompok-5/act/led ==> off
Publish message: 29
29.80
kelompok-5/act/led ==> off
Publish message: 30
30.00
kelompok-5/act/led ==> on
Publish message: 30
30.30
kelompok-5/act/led ==> on
Publish message: 30
```

#Sisi server



Kelompok5 - mqtt://broker.hivemq.com:1883

Topic to publish: sukses_konek

QoS: 0 - Almost Once

Retain ☐

Payload Type: Strings / JSON / XML / Characters

e.g: {'hello':'world'}

Payload

Publish

kelompok-5/sensor/temp

36

qos : 0, **retain** : false, **cmd** : publish, **dup** : false, **topic** : kelompok-5/sensor/temp, **messageId** : , **length** : 26

36

qos : 0, **retain** : false, **cmd** : publish, **dup** : false, **topic** : kelompok-5/sensor/temp, **messageId** : , **length** : 26

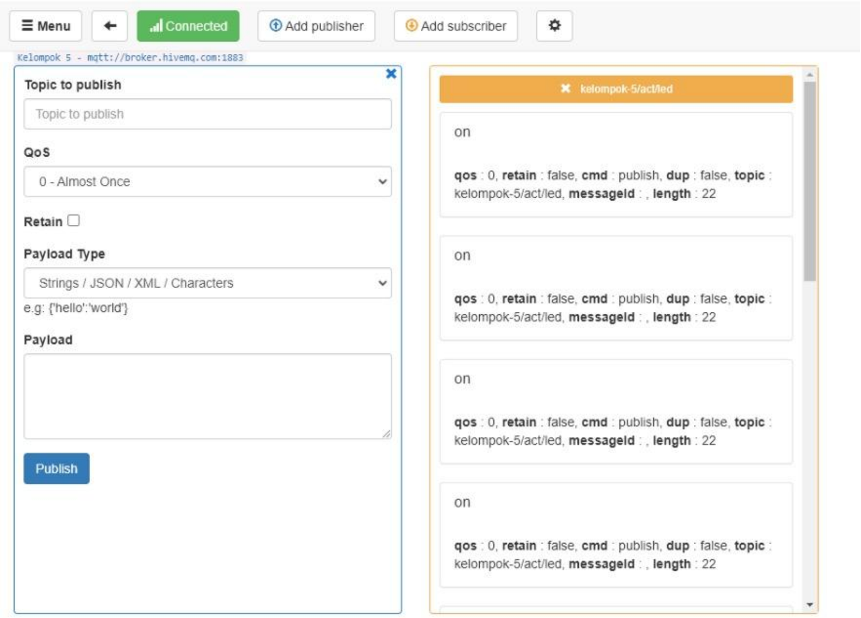
36

qos : 0, **retain** : false, **cmd** : publish, **dup** : false, **topic** : kelompok-5/sensor/temp, **messageId** : , **length** : 26

```
#Client
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include "DHT.h"

WiFiClient espClient;
PubSubClient client(espClient);

#define WIFI_SSID "METHA"
#define WIFI_PASS "kupukupu"
#define MQTT_SERVER "broker.hivemq.com"
#define MQTT_PORT 1883
#define DHTPIN D1
#define DHTTYPE DHT11
#define LED D2

unsigned long _waiting = millis();
unsigned long _now;
int value = 0;
char data[50];

DHT dht(DHTPIN, DHTTYPE);

String buffData;

void setup_wifi() {
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(WIFI_SSID);
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
```

```
void callback(char* topic, byte* payload, unsigned int length){
  buffData = "";
  for(int i=0; i<length; i++){
    buffData += (char) payload[i];
  }

  Serial.print(topic);
  Serial.print(" ==> ");
  Serial.println(buffData);

  if (buffData == "on") {
    digitalWrite(LED, HIGH);
  }

  if (buffData == "off") {
    digitalWrite(LED, LOW);
  }
}

void reconnect() {
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    String clientId = "Kelompok-5-";
    clientId += String(random(0xffff), HEX);
    if (client.connect(clientId.c_str())) {
      Serial.println("connected");
      client.publish("sukses_konek", "Yess... saya terkoneksi");
      client.subscribe("kelompok-5/act/led");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}

void setup() {
  Serial.begin(115200);
  setup_wifi();
  pinMode(LED, OUTPUT);
  client.setServer(MQTT_SERVER, MQTT_PORT);
  client.setCallback(callback);
  dht.begin();
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
  kirimPer2Detik();
}

void kirimPer2Detik(){
  _now = millis();
  if(millis() - _waiting > 2000){
    _waiting = _now;
    float temperature = dht.readTemperature();
    itoa(temperature, data, 10);
    Serial.print("Publish message: ");
    Serial.println(data);
    Serial.println(temperature);
    client.publish("kelompok-5/sensor/temp", data);
  }
}
```

#Sisi Client



Kelompok 5 - mqtt://broker.hivemq.com:1883

Topic to publish: Topic to publish

QoS: 0 - Almost Once

Retain □

Payload Type: Strings / JSON / XML / Characters

e.g: {'hello':'world'}

Payload

on

qos : 0, **retain** : false, **cmd** : publish, **dup** : false, **topic** : kelompok-5/act/led, **messageId** : , **length** : 22

on

qos : 0, **retain** : false, **cmd** : publish, **dup** : false, **topic** : kelompok-5/act/led, **messageId** : , **length** : 22

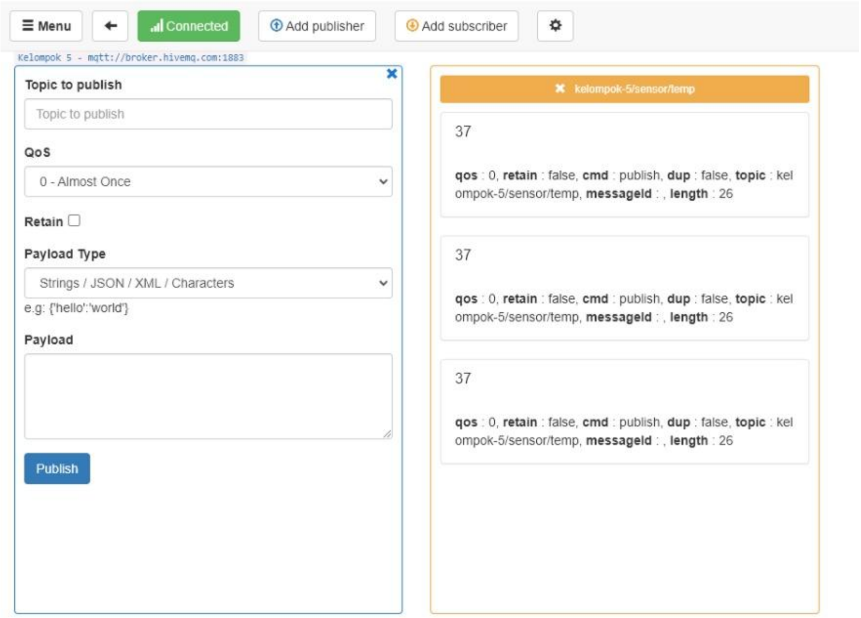
on

qos : 0, **retain** : false, **cmd** : publish, **dup** : false, **topic** : kelompok-5/act/led, **messageId** : , **length** : 22

on

qos : 0, **retain** : false, **cmd** : publish, **dup** : false, **topic** : kelompok-5/act/led, **messageId** : , **length** : 22

#Sisi Client



Kelompok 5 - mqtt://broker.hivemq.com:1883

Topic to publish: Topic to publish

QoS: 0 - Almost Once

Retain □

Payload Type: Strings / JSON / XML / Characters

e.g: {'hello':'world'}

Payload

37

qos : 0, **retain** : false, **cmd** : publish, **dup** : false, **topic** : kelompok-5/sensor/temp, **messageId** : , **length** : 26

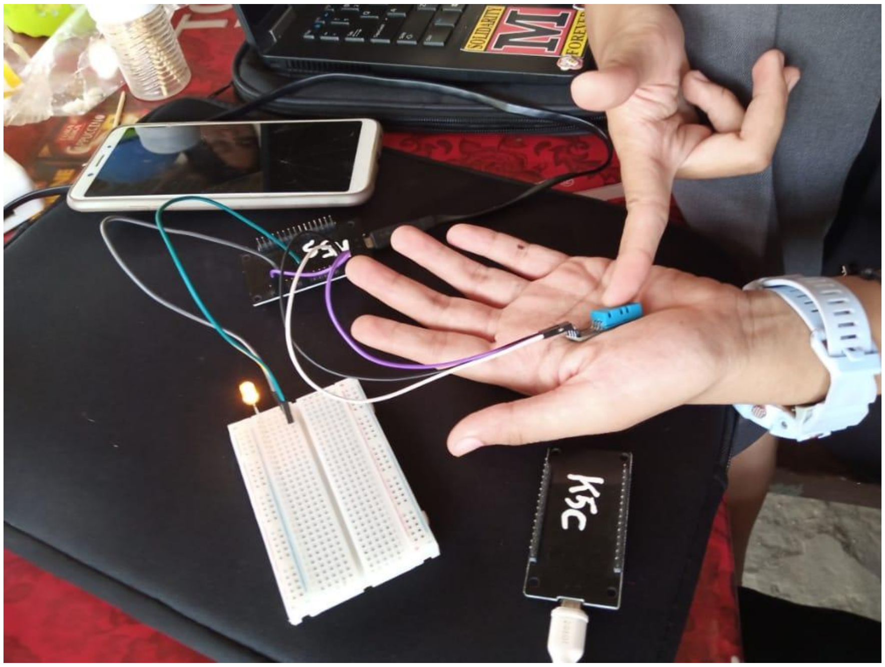
37

qos : 0, **retain** : false, **cmd** : publish, **dup** : false, **topic** : kelompok-5/sensor/temp, **messageId** : , **length** : 26

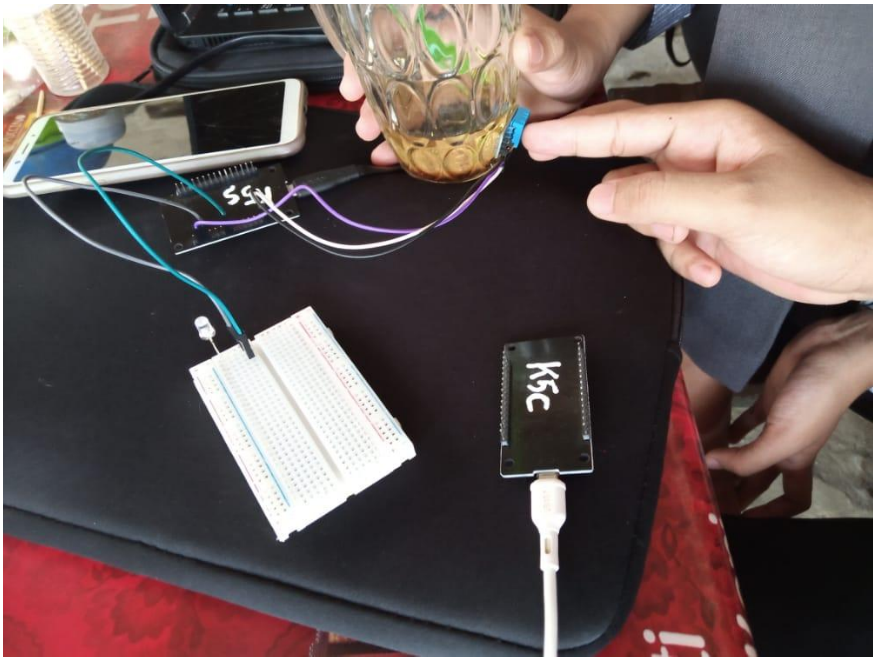
37

qos : 0, **retain** : false, **cmd** : publish, **dup** : false, **topic** : kelompok-5/sensor/temp, **messageId** : , **length** : 26

#Mode ON



#Mode OFF



Kesimpulan:

NodeMCU 1 bertugas sebagai pengambil data sensor dan terhubung dengan LED. Sedangkan NodeMCU 2 bertugas sebagai pemroses data yang dikirimkan oleh NodeMCU 1 melalui jaringan dengan protokol MQTT temperature.

Jika Temperature > 29 maka led akan menyala
Jika temperature <29 maka led akan mati

Nodemcu 1 itu berfungsi sebagai server.ada sensor dht yang terhubung ke dalam sensor dht dan juga led sebagi indikator bahwa suhunya diatas 29 led dakan menyala jidad kurang dari 29 akan off lednya.di noodemcu 1 juga akan mempublis datanya ke mqtt public seperti contohnya yang ada di atas dengan topik kelompok-5/sensor/temp dan node mcu 1 juga dia akan menerima data dari nodemcu 2 melalui mqtt broker public dengan topic kelompok-5/act/led.
Sedangkan nodemcu 2 nya berfungsi sebagai pengelola data ,data yanng diambil ialah data suhu dari nodemcu 1dengan menggunakan mqtt broker public dengan topic kelompok-5/sensor/temp jika data yang diambil >29 maka nodemcu2 akan mengirimkan data on ke mqtt broker public dengan topic kelompok-5/act/led

3. Lakukan hal yang sama seperti No. 2 namun menggunakan local Broker MQTT yang sudah 7di install pada No. 1.

   Jawaban

   SERVER

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include "DHT.h"

WiFiClient espClient;
PubSubClient client(espClient);

#define WIFI_SSID "Hakim Asrori"
#define WIFI_PASS ""
#define MQTT_SERVER "192.168.43.37"
#define MQTT_PORT 1883
#define DHTPIN D1
#define DHTTYPE DHT11
#define LED D2

unsigned long _waiting = millis();
unsigned long _now;
int value = 0;
char data[50];
```

```
DHT dht(DHTPIN, DHTTYPE);

String buffData;

void setup_wifi() {
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(WIFI_SSID);
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* payload, unsigned int length){
  buffData = "";
  for(int i=0; i<length; i++){
    buffData += (char) payload[i];
  }

  Serial.print(topic);
  Serial.print(" ==> ");
  Serial.println(buffData);

  if (buffData == "on") {
    digitalWrite(LED, HIGH);
  }
  if (buffData == "off") {
    digitalWrite(LED, LOW);
  }
}

void reconnect() {
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    String clientId = "Kelompok-5-";
    clientId += String(random(0xffff), HEX);
    if (client.connect(clientId.c_str())) {
      Serial.println("connected");
      client.publish("sukses_konek", "Yess... saya terkoneksi");
      client.subscribe("kelompok-5/act/led");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}

void setup() {
  Serial.begin(115200);
  setup_wifi();
  pinMode(LED, OUTPUT);
  client.setServer(MQTT_SERVER, MQTT_PORT);
  client.setCallback(callback);
  dht.begin();
}
```

```
void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
  kirimPer2Detik();
}

void kirimPer2Detik(){
  _now = millis();
  if(millis() - _waiting > 2000){
    _waiting = _now;
    float temperature = dht.readTemperature();
    itoa(temperature, data, 10);
    Serial.print("Publish message: ");
    Serial.println(data);
    Serial.println(temperature);
    client.publish("kelompok-5/sensor/temp", data);
  }
}
```

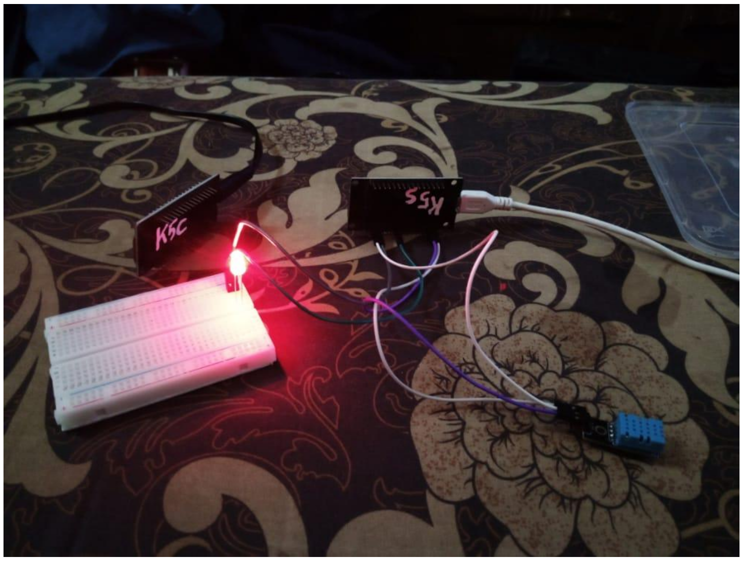
CLIENT

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

WiFiClient espClient;
PubSubClient client(espClient);

#define WIFI_SSID "Hakim Asrori"
#define WIFI_PASS ""
#define MQTT_SERVER "192.168.43.37"
```

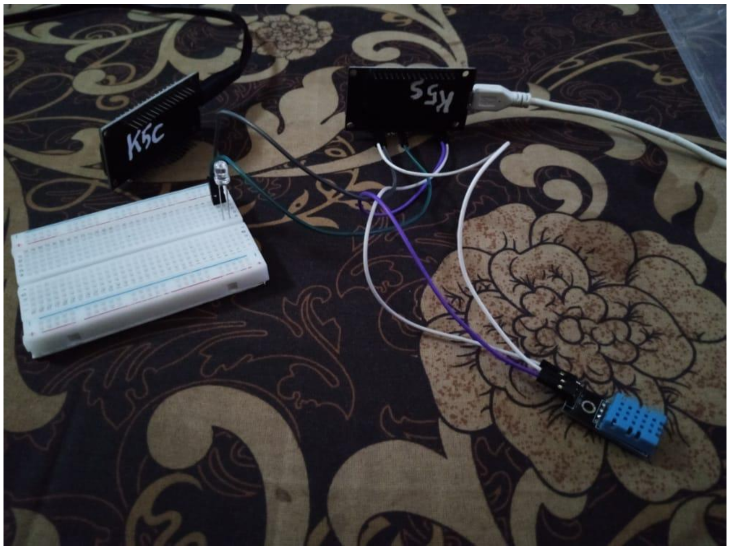
```
#define MQTT_PORT 1883

unsigned long _waiting = millis();
unsigned long _now;
int value = 0;
char data[50];

String buffData;

void setup_wifi() {
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(WIFI_SSID);
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* payload, unsigned int length){
  buffData = "";
  for(int i=0; i<length; i++){
    buffData += (char) payload[i];
  }

  Serial.print(topic);
  Serial.print(" ==> ");
  Serial.println(buffData);

  if (buffData > "29") {
    client.publish("kelompok-5/act/led", "on");
  } else {
    client.publish("kelompok-5/act/led", "off");
  }
}

void reconnect() {
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    String clientId = "Kelompok-5-";
    clientId += String(random(0xffff), HEX);
    if (client.connect(clientId.c_str())) {
      Serial.println("connected");
      client.publish("sukses_konek", "Yess... saya terkoneksi");
      client.subscribe("kelompok-5/sensor/temp");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}

void setup() {
  Serial.begin(115200);
  setup_wifi();
  client.setServer(MQTT_SERVER, MQTT_PORT);
  client.setCallback(callback);
```

```
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
}
```

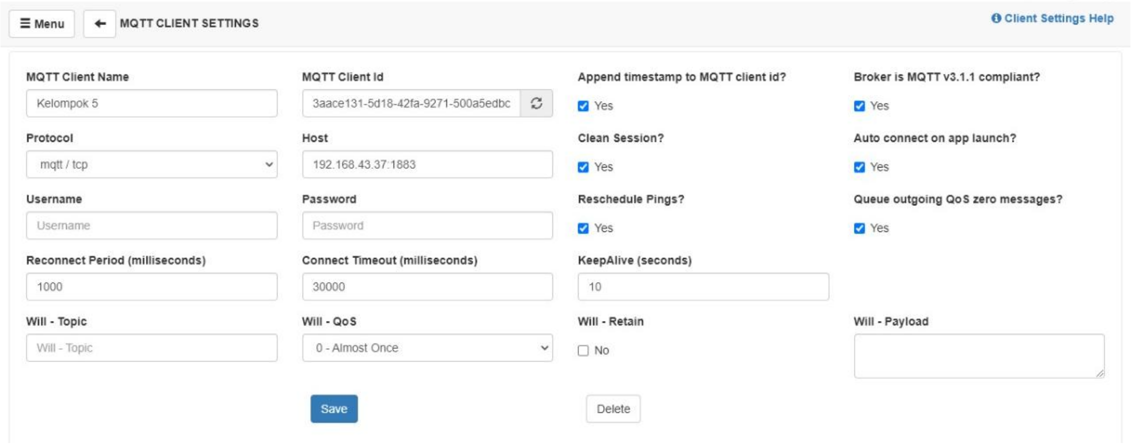
Suhu > 29



Suhu <29



Settingan MQTTBox local



MQTT Client Name: Kelompok 5

MQTT Client Id: 3aace131-5d18-42fa-9271-500a5edbc

Append timestamp to MQTT client id? ☑ Yes

Broker is MQTT v3.1.1 compliant? ☑ Yes

Protocol: mqtt / tcp

Host: 192.168.43.37:1883

Clean Session? ☑ Yes

Auto connect on app launch? ☑ Yes

Username: Username

Password: Password

Reschedule Pings? ☑ Yes

Queue outgoing QoS zero messages? ☑ Yes

Reconnect Period (milliseconds): 1000

Connect Timeout (milliseconds): 30000

KeepAlive (seconds): 10

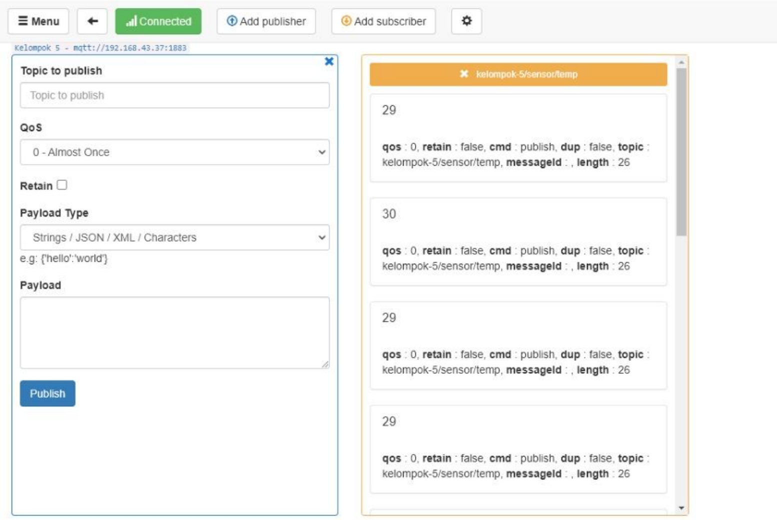
Will - Topic: Will - Topic

Will - QoS: 0 - Almost Once

Will - Retain: □ No

Will - Payload:

Hasil



Topic to publish: Topic to publish

QoS: 0 - Almost Once

Retain □

Payload Type: Strings / JSON / XML / Characters

e.g: {'hello':'world'}

Payload:

29

qos : 0, **retain** : false, **cmd** : publish, **dup** : false, **topic** : kelompok-5/sensor/temp, **messageId** : , **length** : 26

30

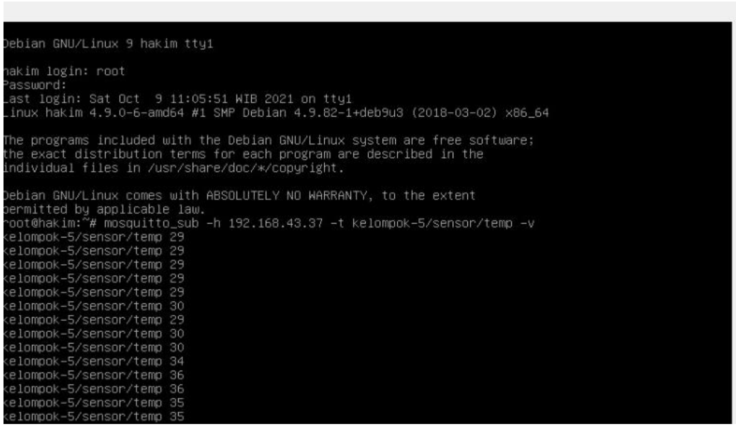
qos : 0, **retain** : false, **cmd** : publish, **dup** : false, **topic** : kelompok-5/sensor/temp, **messageId** : , **length** : 26

29

qos : 0, **retain** : false, **cmd** : publish, **dup** : false, **topic** : kelompok-5/sensor/temp, **messageId** : , **length** : 26

29

qos : 0, **retain** : false, **cmd** : publish, **dup** : false, **topic** : kelompok-5/sensor/temp, **messageId** : , **length** : 26



```
Debian GNU/Linux 9 hakim tty1

hakim login: root
Password:
Last login: Sat Oct  9 11:05:51 WIB 2021 on tty1
Linux hakim 4.9.0-6-amd64 #1 SMP Debian 4.9.82-1+deb9u3 (2018-03-02) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@hakim:~# mosquitto_sub -h 192.168.43.37 -t kelompok-5/sensor/temp -v
kelompok-5/sensor/temp 29
kelompok-5/sensor/temp 29
kelompok-5/sensor/temp 29
kelompok-5/sensor/temp 29
kelompok-5/sensor/temp 29
kelompok-5/sensor/temp 30
kelompok-5/sensor/temp 29
kelompok-5/sensor/temp 30
kelompok-5/sensor/temp 30
kelompok-5/sensor/temp 34
kelompok-5/sensor/temp 36
kelompok-5/sensor/temp 36
kelompok-5/sensor/temp 35
kelompok-5/sensor/temp 35
```

Behubungan Laptopnya eror menggunakan mosquito windows jadi kelompok kami menggunakan oprasi system lain yaitu debian 9 untuk menjalankan mosquito broker atau mqtt broker lokalnya.

Kesimpulan : Yaitu nodemcu sebagai server yang memiliki data suhu dan juga terhubung ke led sebagai Indikator suhu diatas 29.

Nodemcu 1 itu berfungsi sebagai server.ada sensor dht yang terhubung ke dalam sensor dht dan juga led sebagi indikator bahwa suhunya diatas 29 led dakan menyala jidad kurang dari 29 akan off lednya.di noodemcu 1 juga akan mempublis datanya ke mqtt public seperti contohnya yang ada di atas dengan topik kelompok-5/sensor/temp dan node mcu 1 juga dia akan menerima data dari nodemcu 2 melalui mqtt broker local dengan topic kelompok-5/act/led.

Sedangkan nodemcu 2 nya berfungsi sebagai pengelola data ,data yanng diambil ialah data suhu dari nodemcu 1dengan menggunakan mqtt broker public dengan topic kelompok-5/sensor/temp jika data yang diambil >29 maka nodemcu2 akan mengirimkan data on ke mqtt broker public dengan topic kelompok-5/act/led