**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.

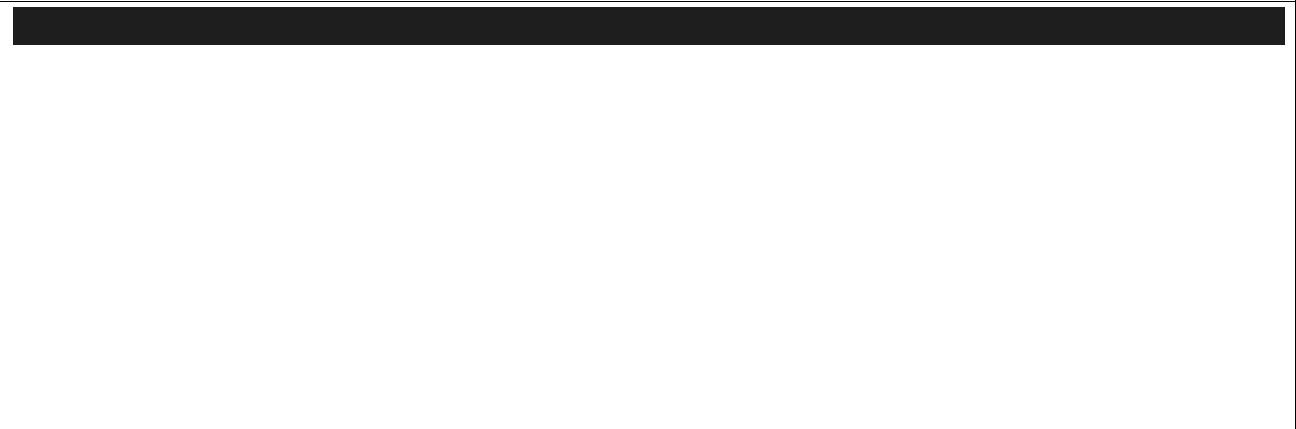
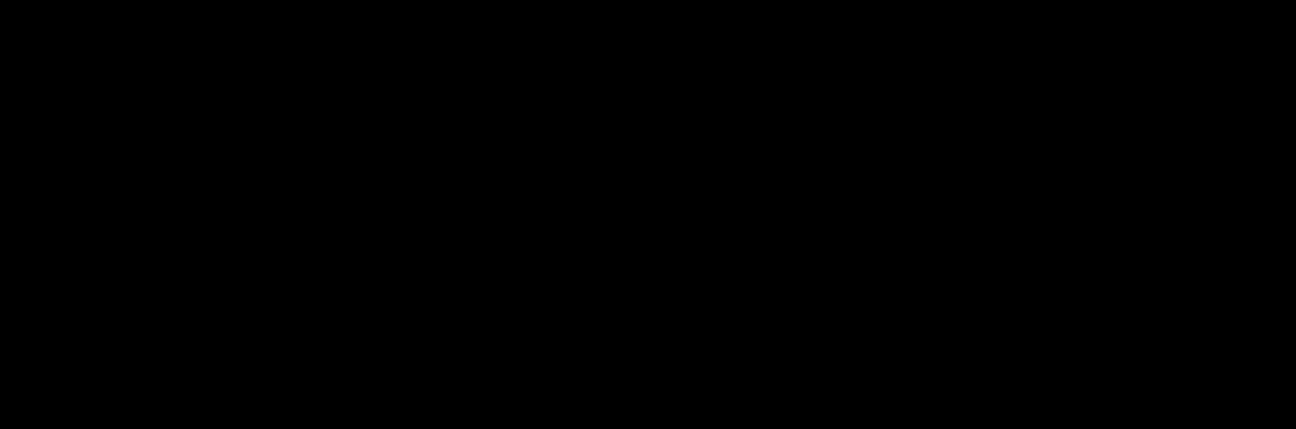
* Mosca Broker dapat diperoleh di laman <https://github.com/moscajs/mosca>
* Mosquitto Broker dapat diperoleh di laman <https://mosquitto.org/download>

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>

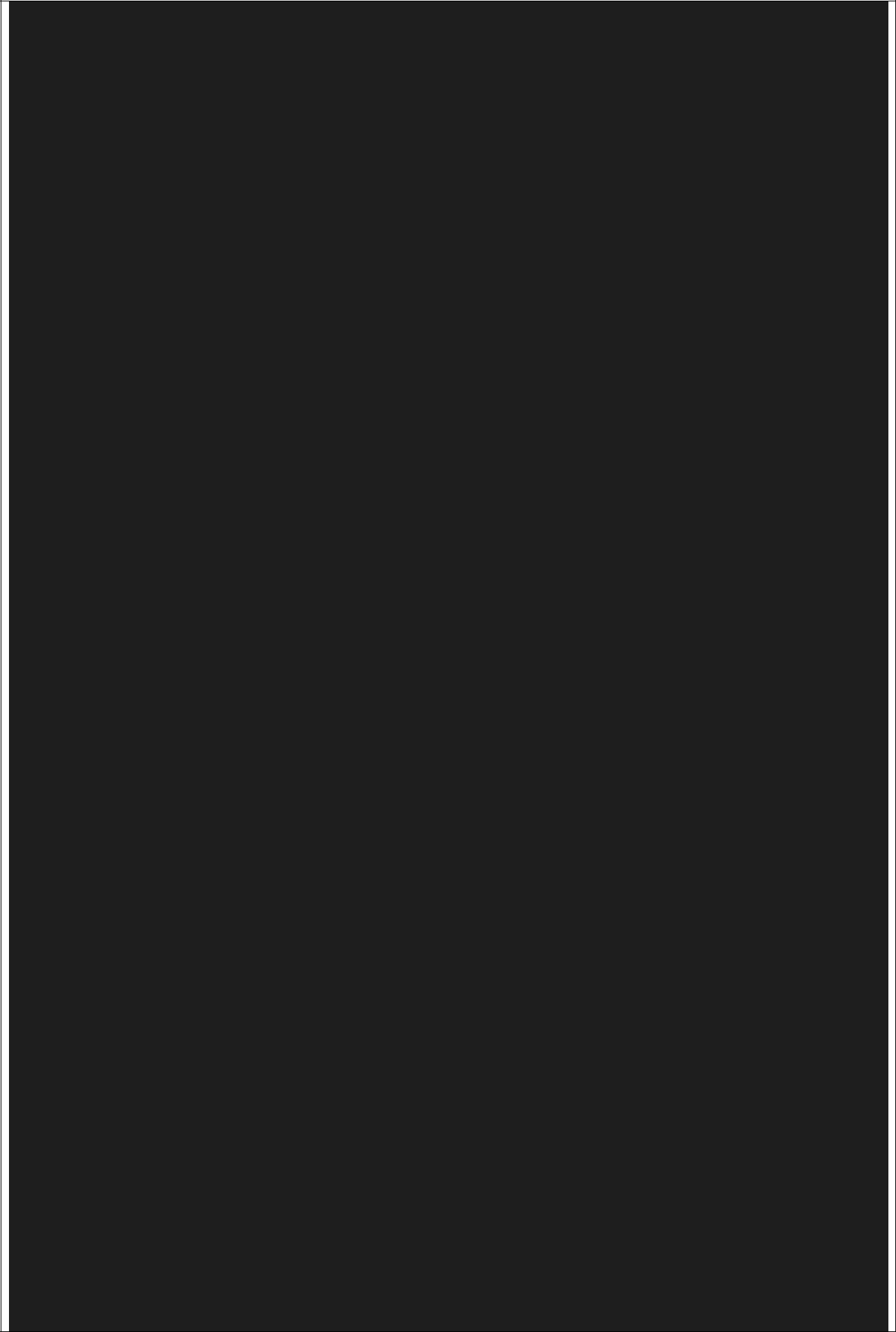


WiFiClient espClient;



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);

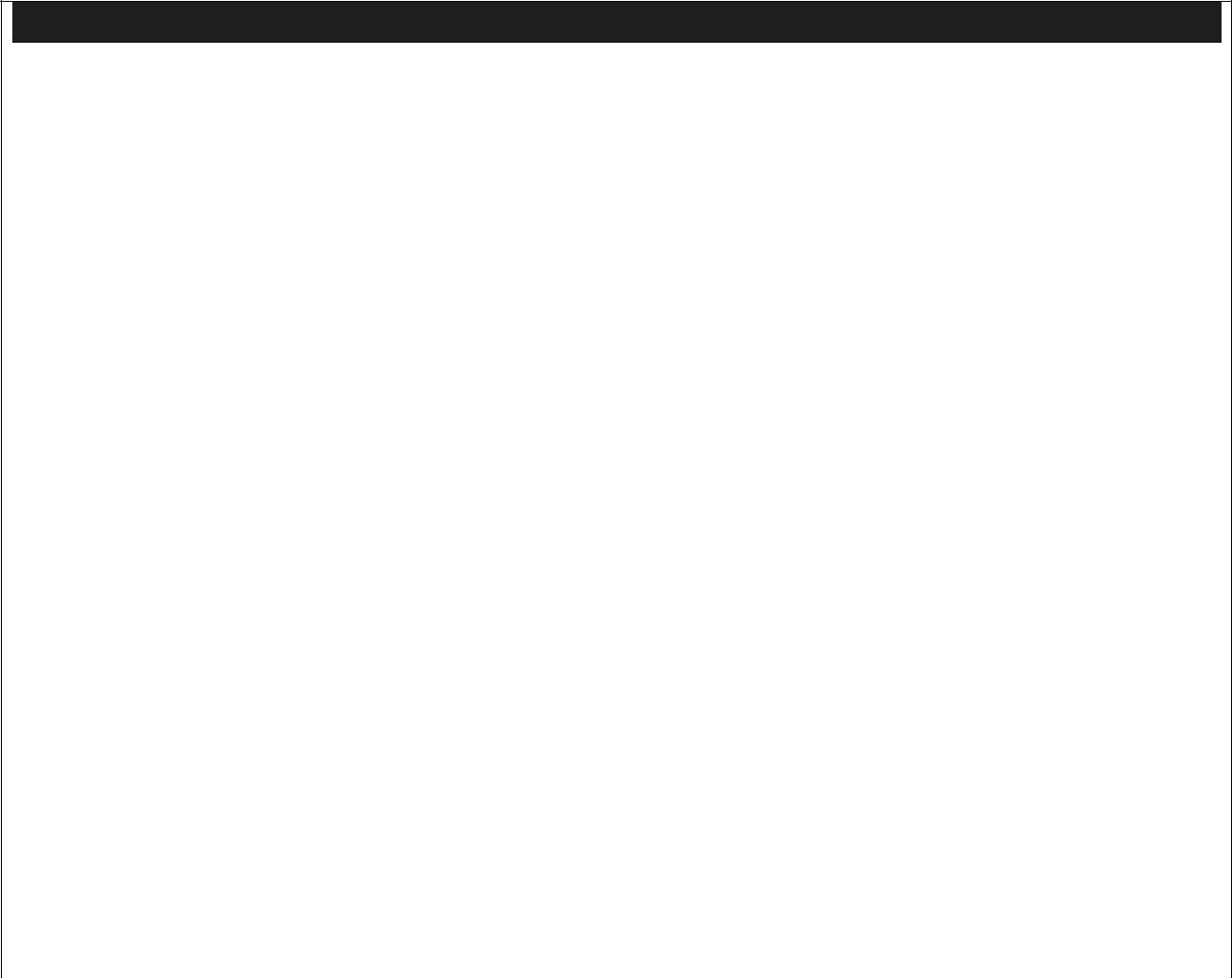
}

}

}

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.



*Gambar 1. Berhasil terhubung ke MQTT Broker*

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



*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);

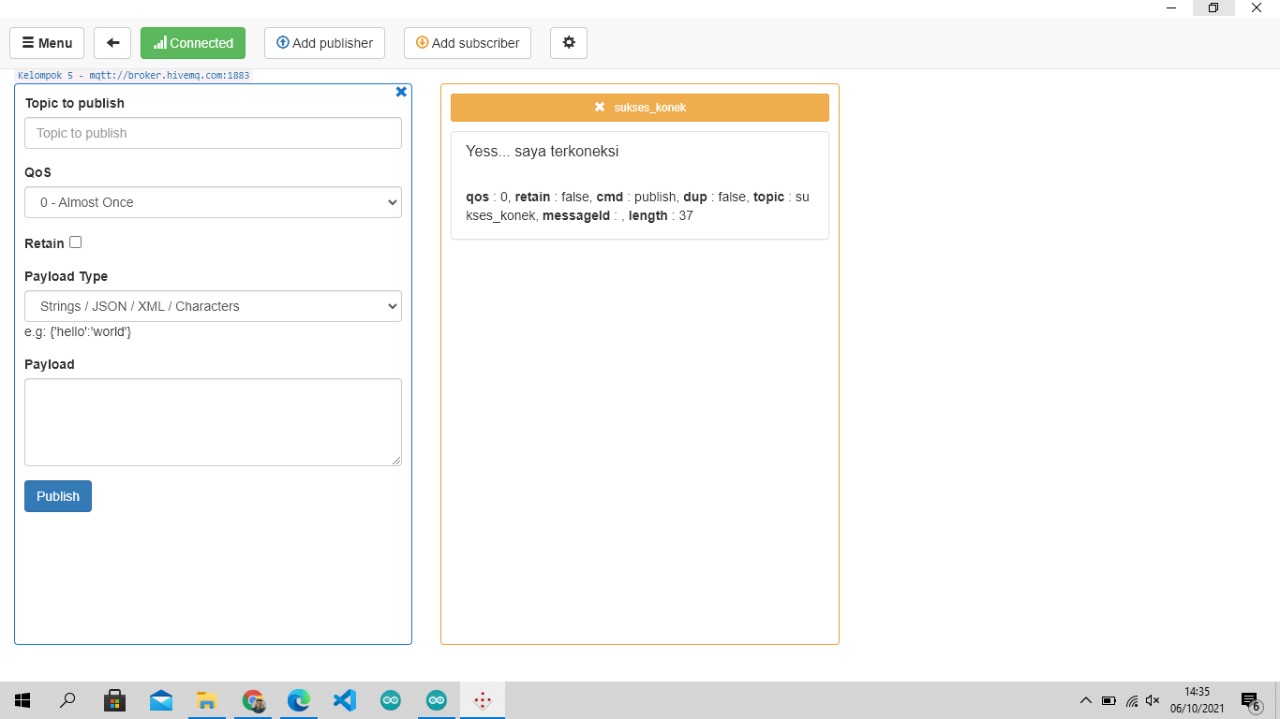
Serial.print("Publish message: ");

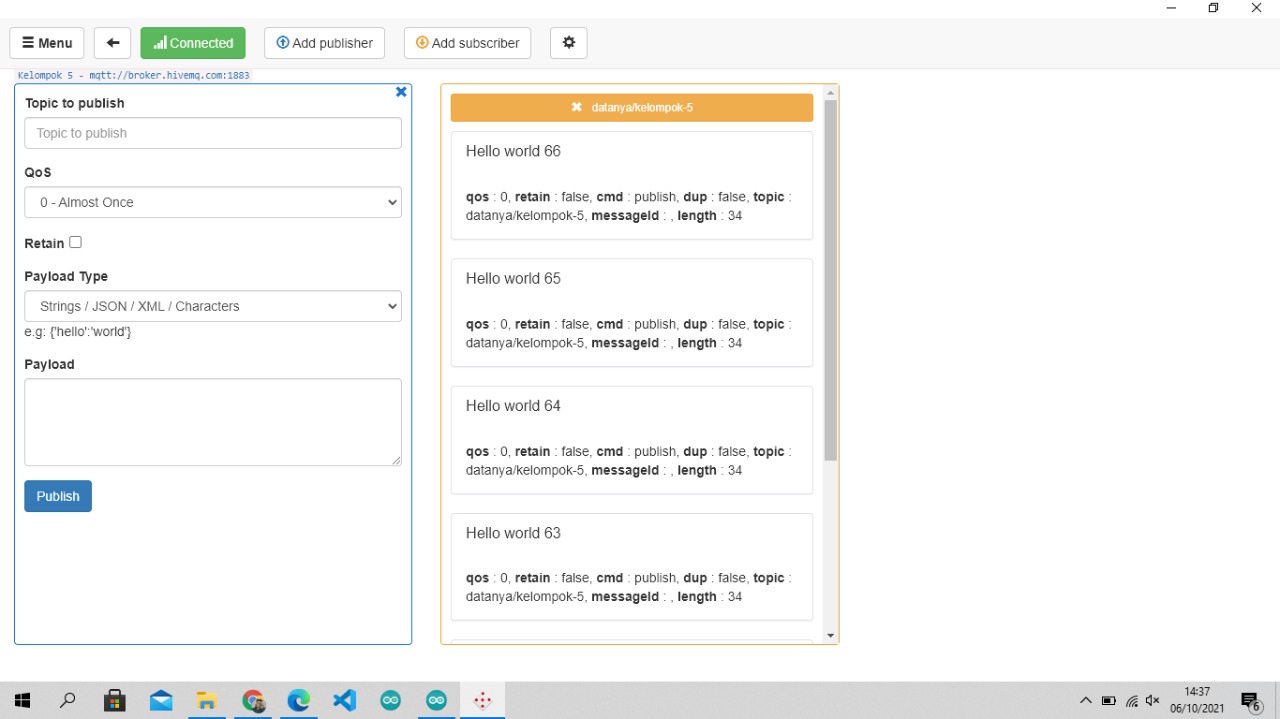
Serial.println(data);

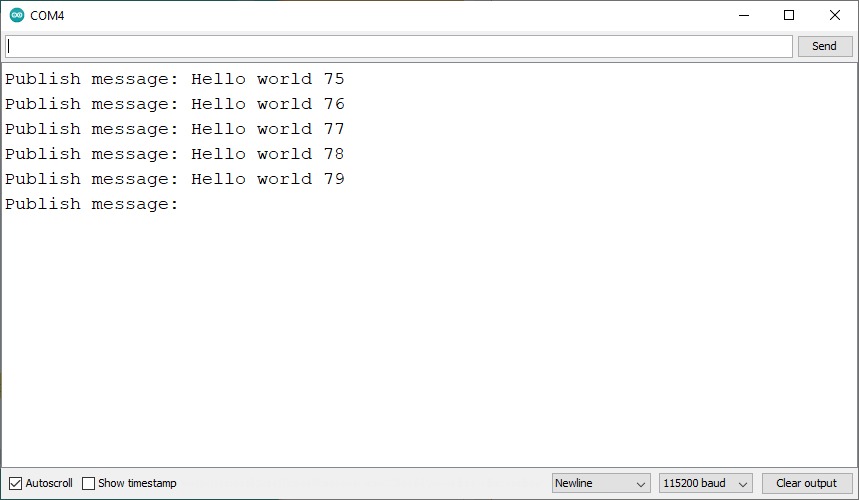
client.publish("datanya/kelompok-5", data);

}

}





**

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.

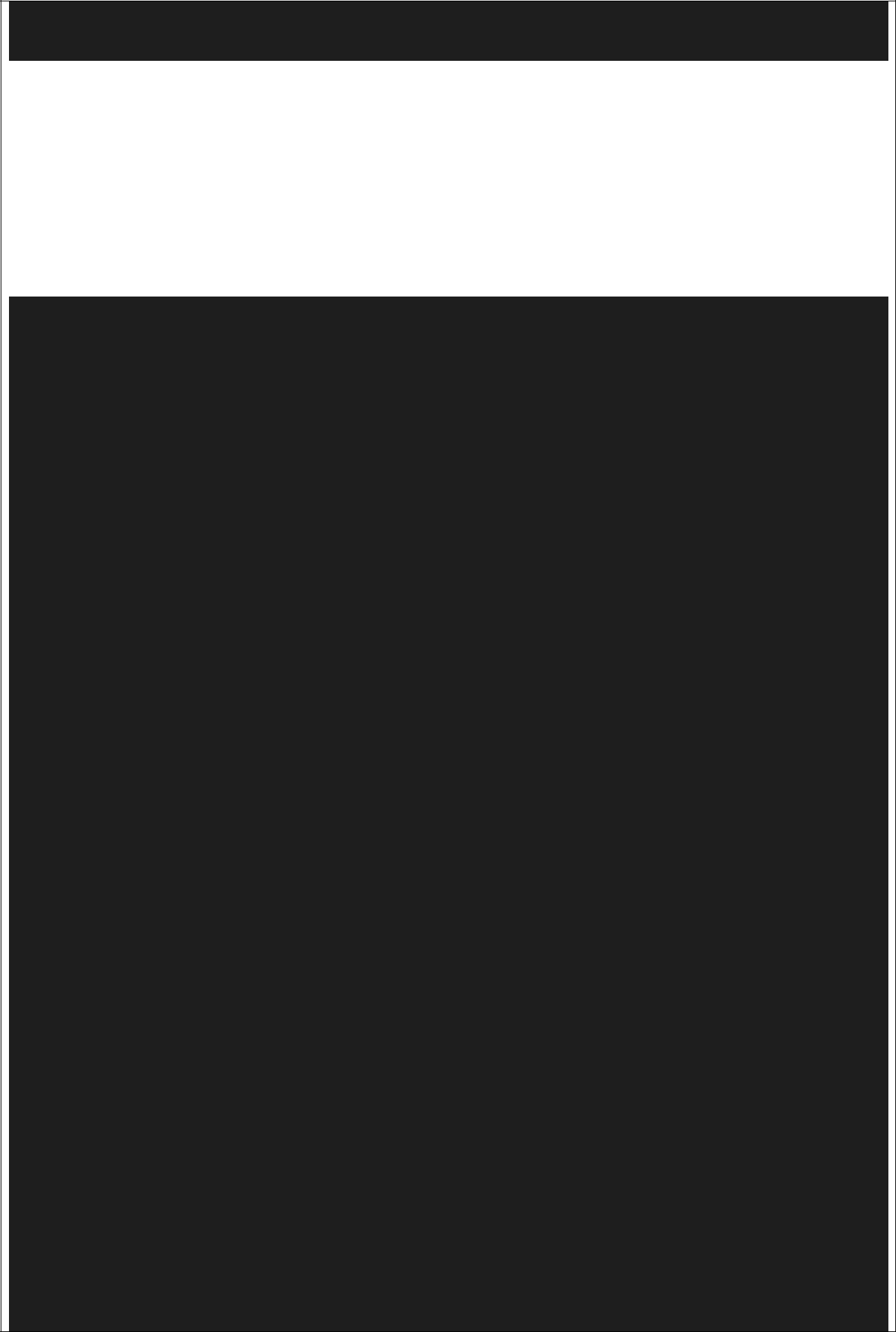
**Program Subscribe**

****

#include <ESP8266WiFi.h>

#include <PubSubClient.h>





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() {

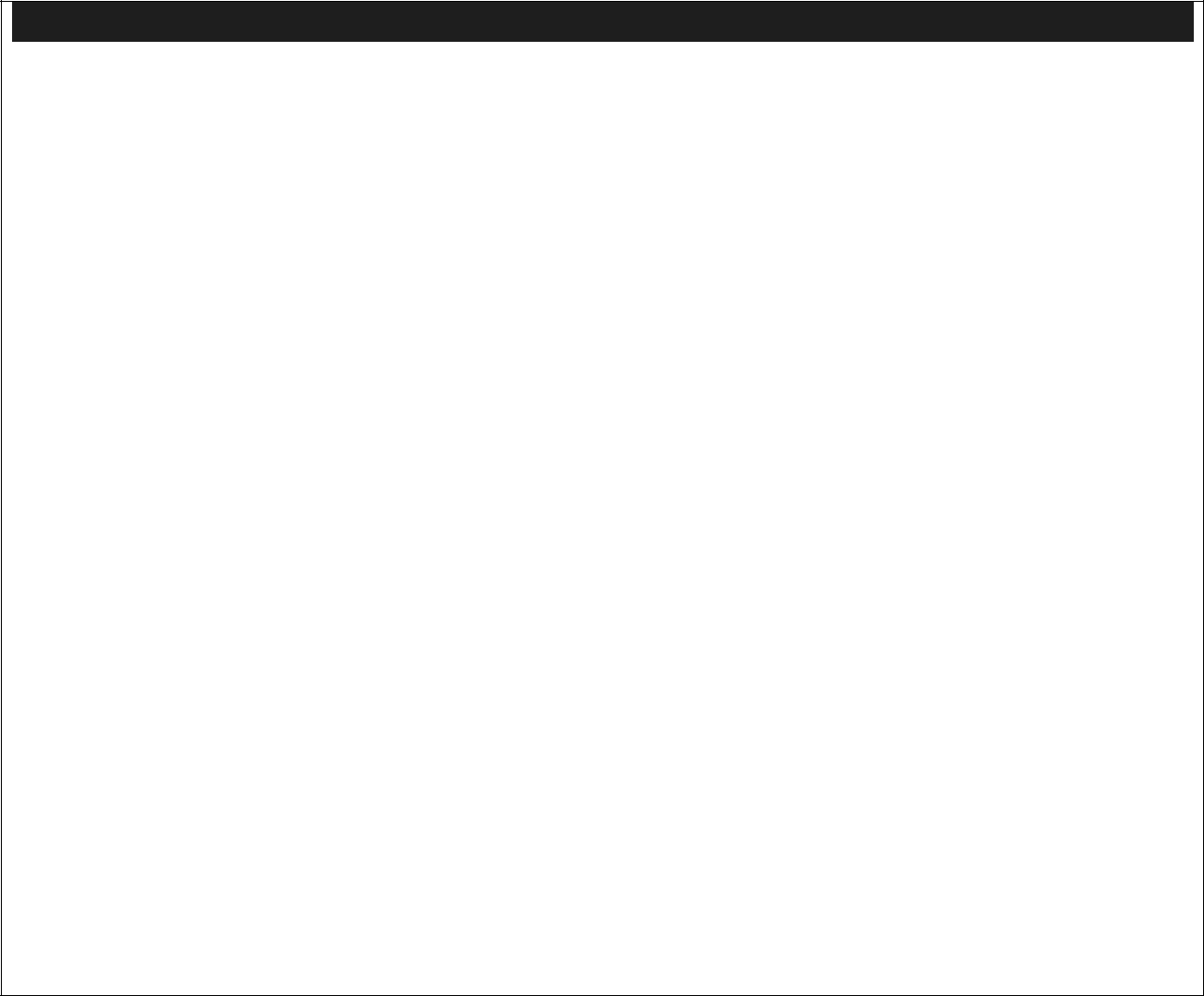
while (!client.connected()) {

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();



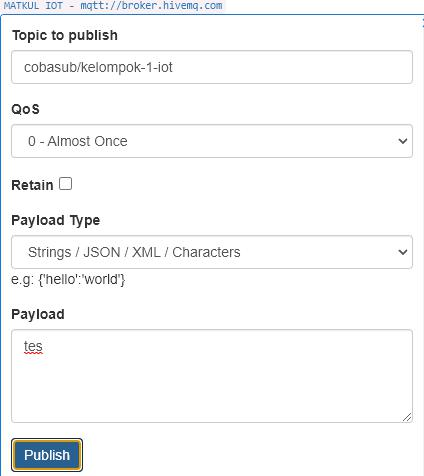
}



client.loop();

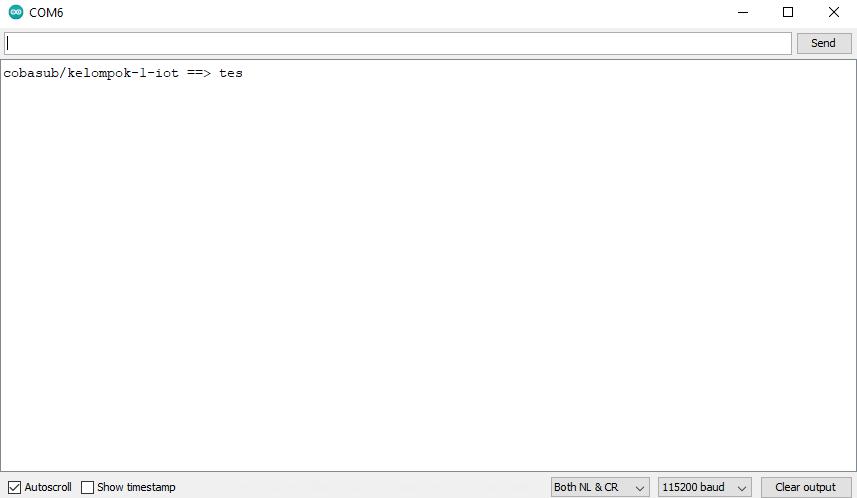
}

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



*Gambar 3. Publish data melalui MQTT Box*

Data yang diterima oleh NodeMCU dan ditempilkan pada serial monitor.



*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() {

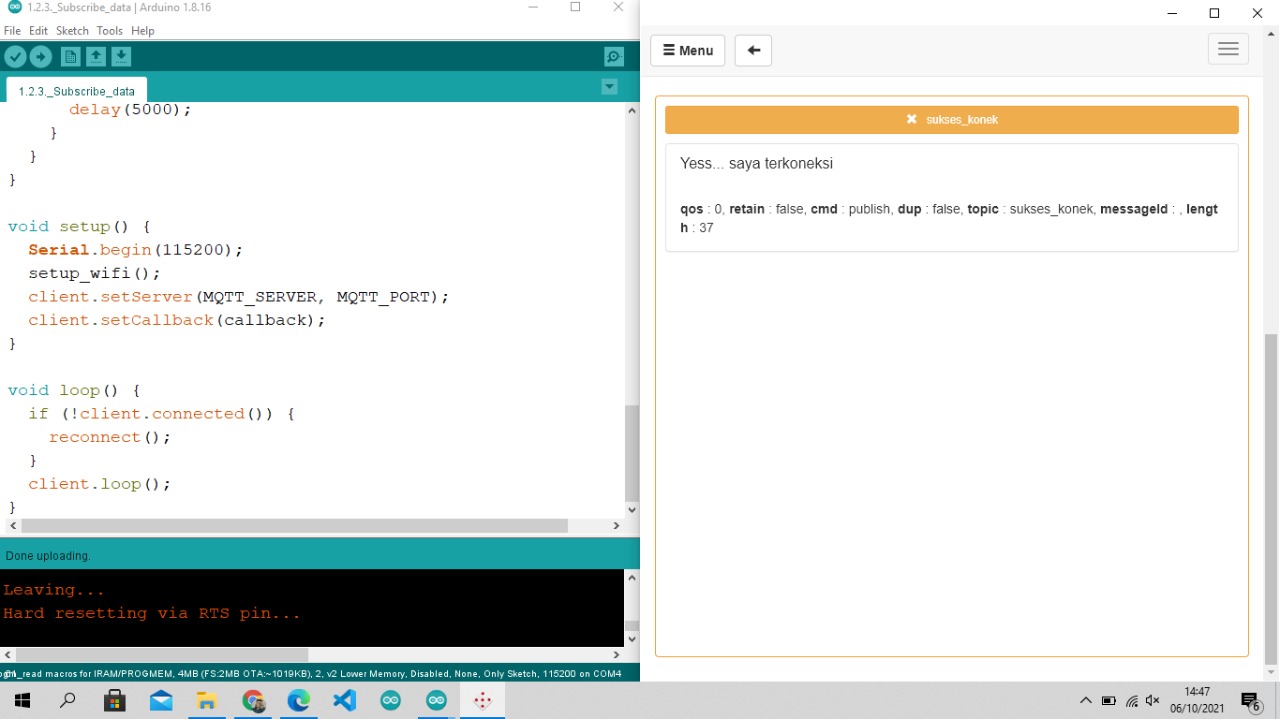
if (!client.connected()) {

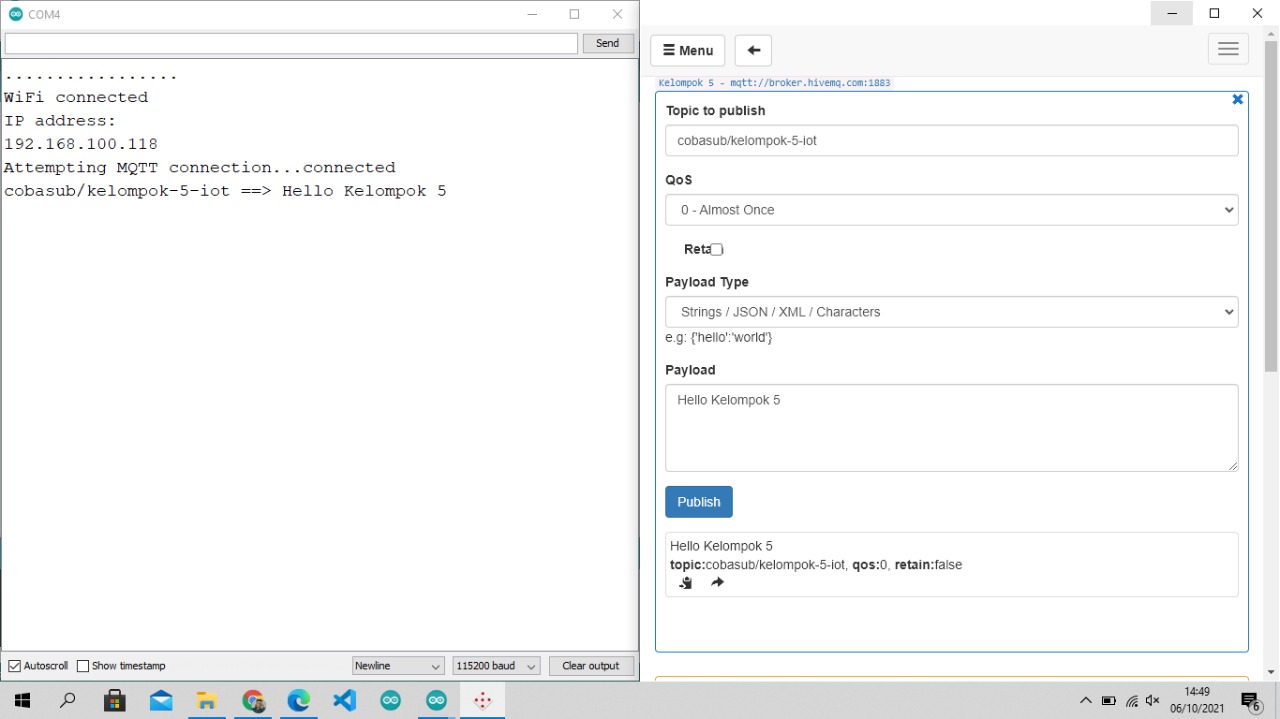
reconnect();

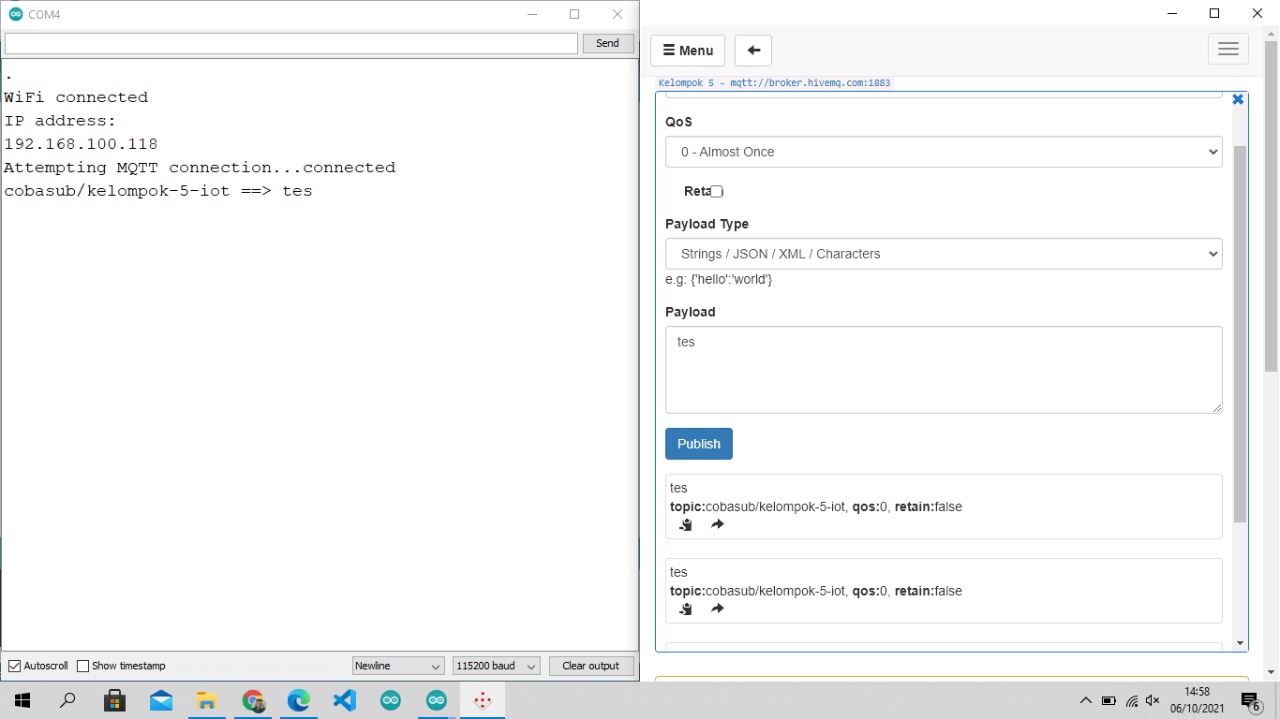
}

client.loop();

}



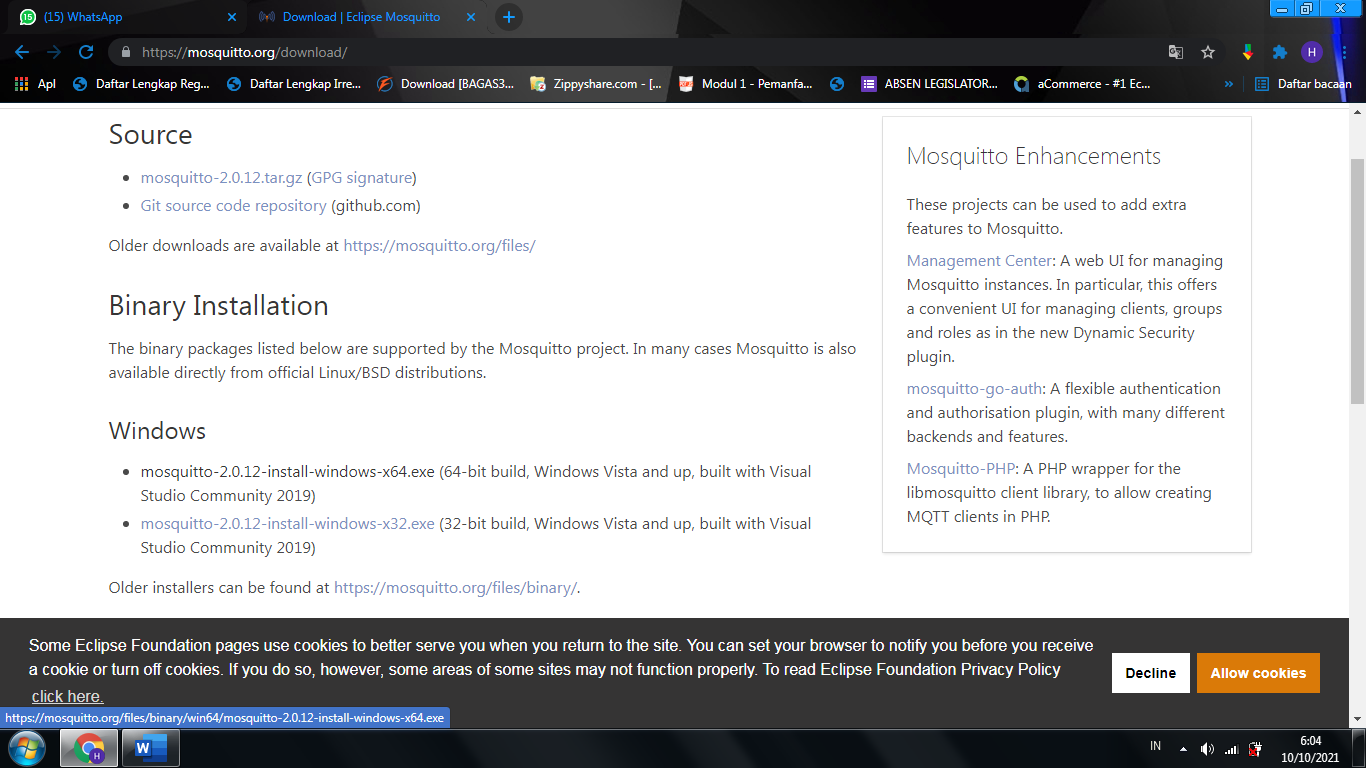




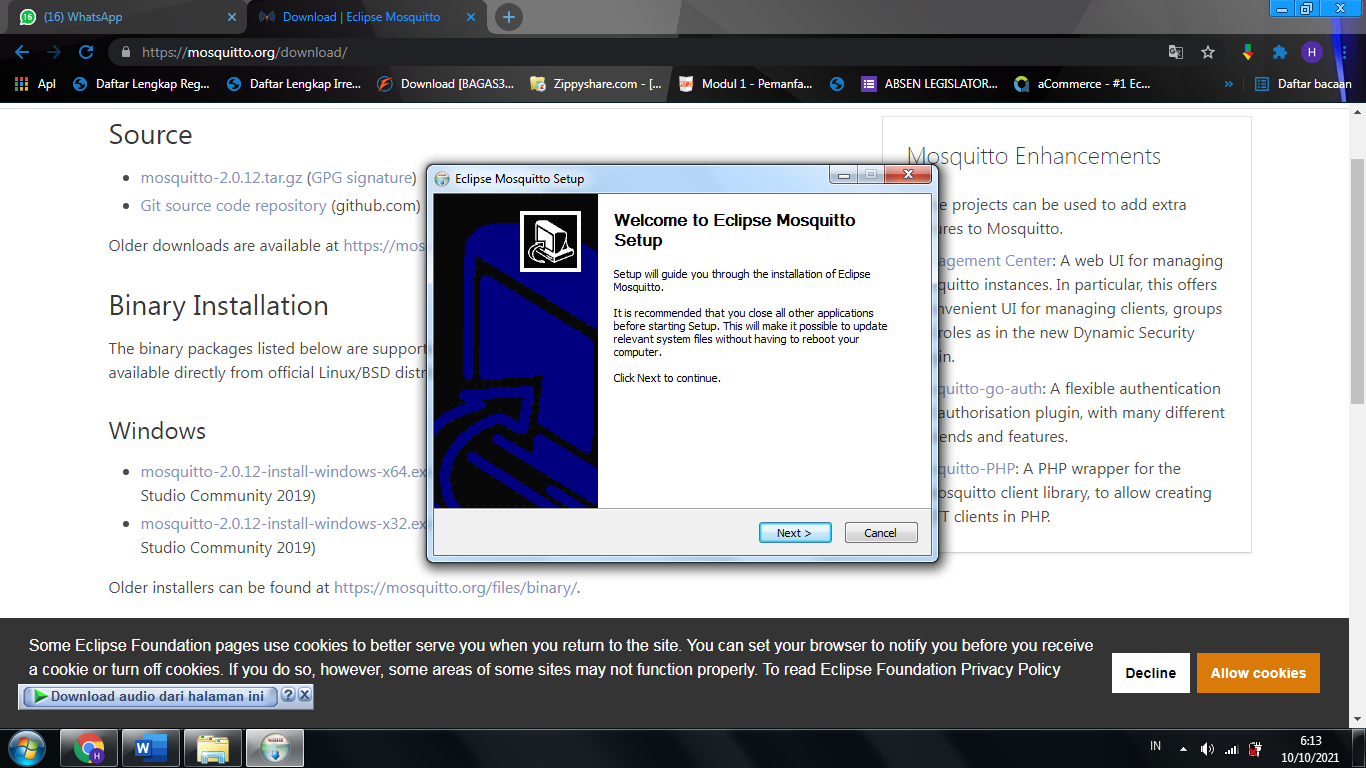
Kesimpulan : Nodemcu akan menerrima data atau subscribe contojnya seperti diatas, melalui mqtt broker publik dengan topik “cobasub/kelompok-5-iot”

**1.3. Tugas**

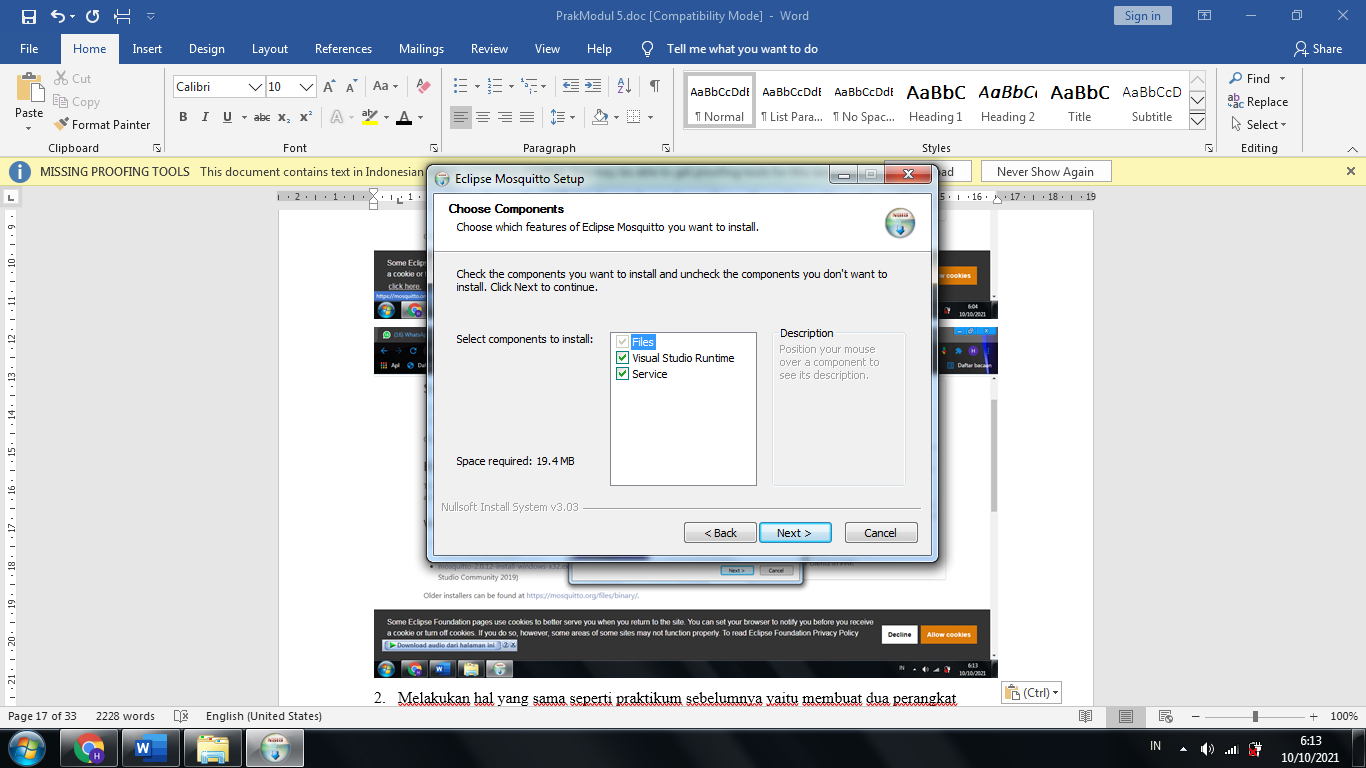
1. Screenshoot setiap langkah-langkah installasi/persiapan Broker MQTT pilih salah satu (Mosca, Mosquitto, EMQX) secara lokal (di komputer/PC/Laptop masing-masing).
   1. Disini kami menggunakan aplikasi mosquitto, jadi yang pertama download mosquitto terlebih dahulu di https://mosquitto.org/download/



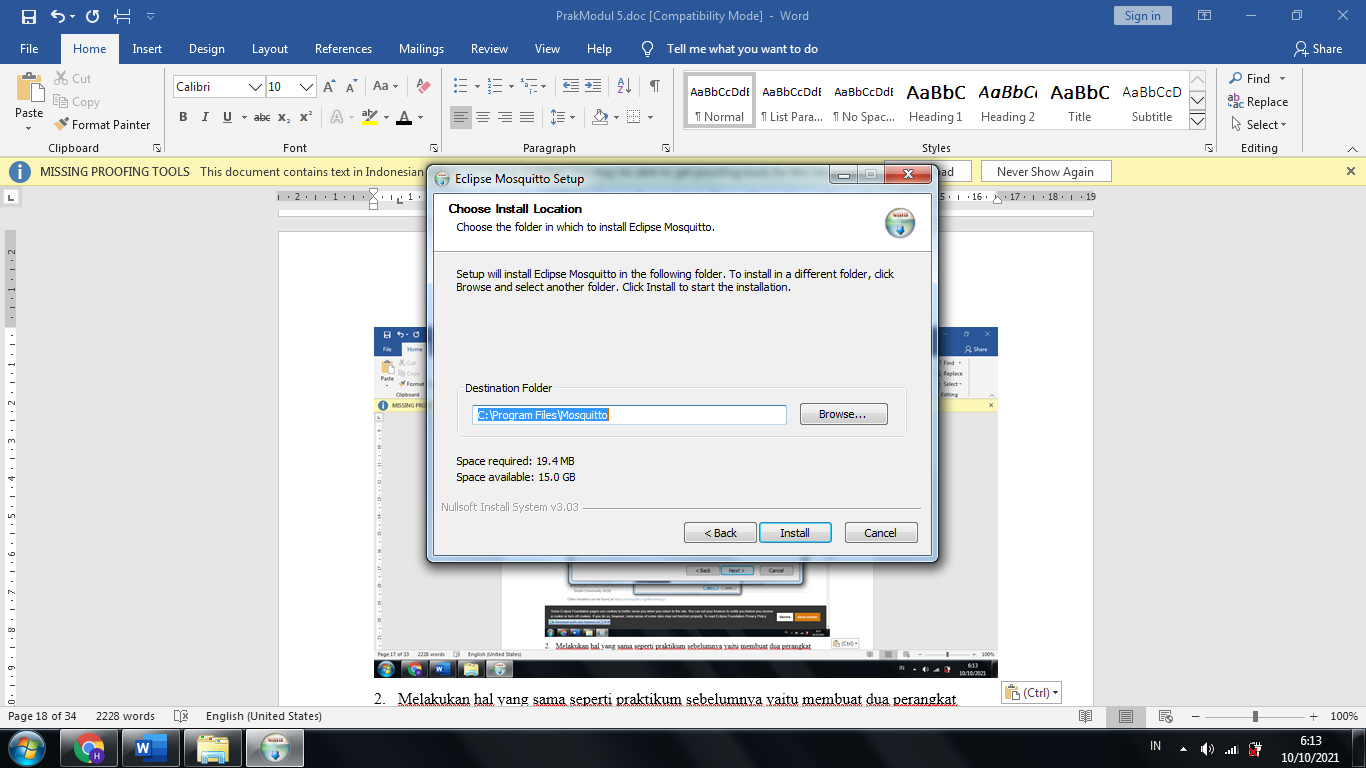
* 1. Setelah didownload, lalu kita double klik untuk menjalankan installernya.
  2. Setelah itu pilih next



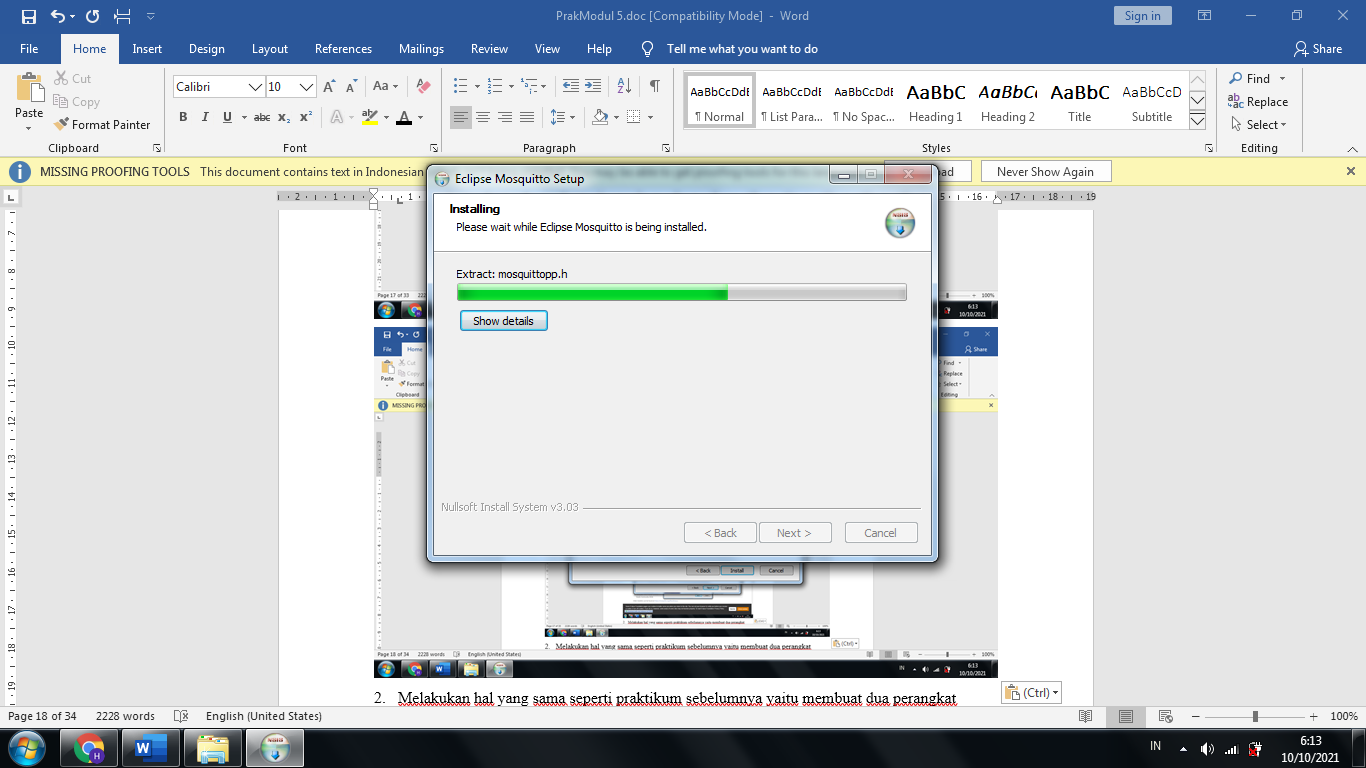
* 1. Centang semua komponen, Setelah itu pilih next



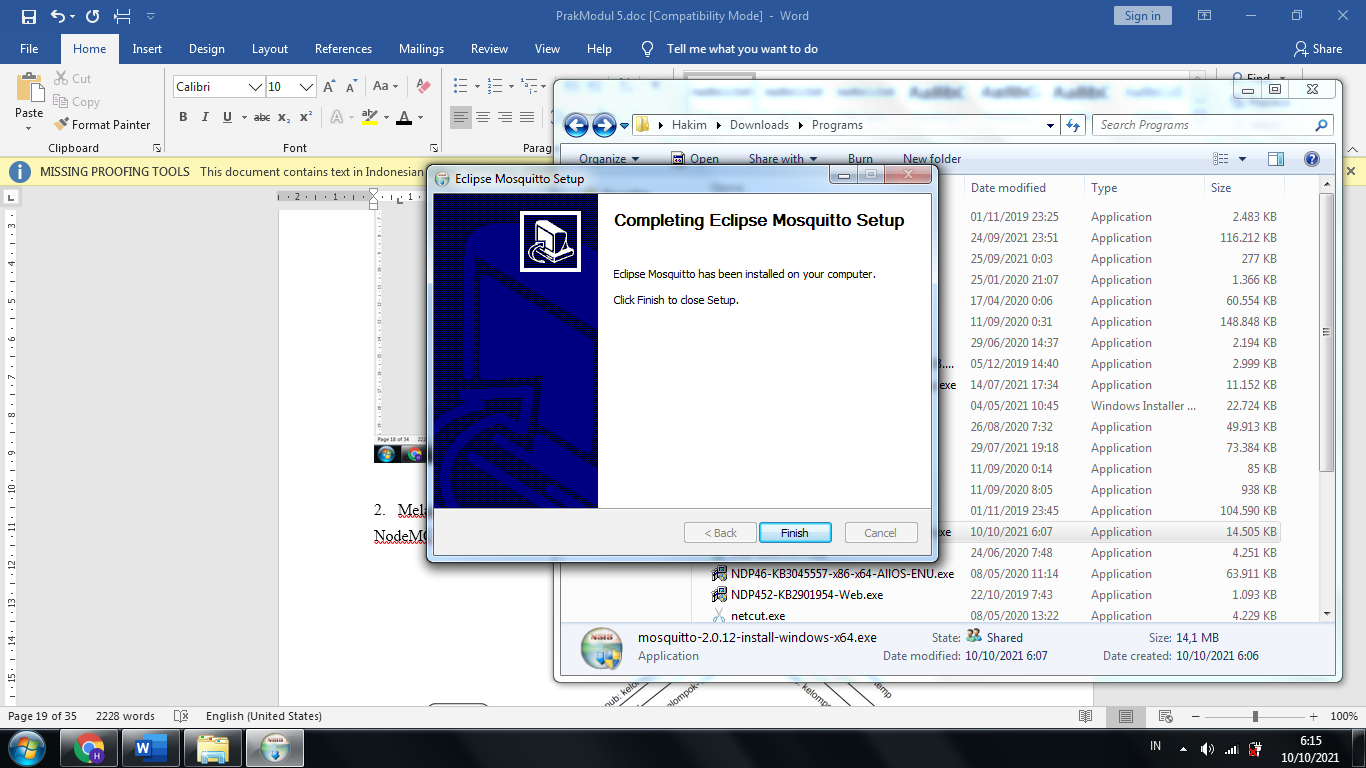
* 1. Setelah itu pilih next



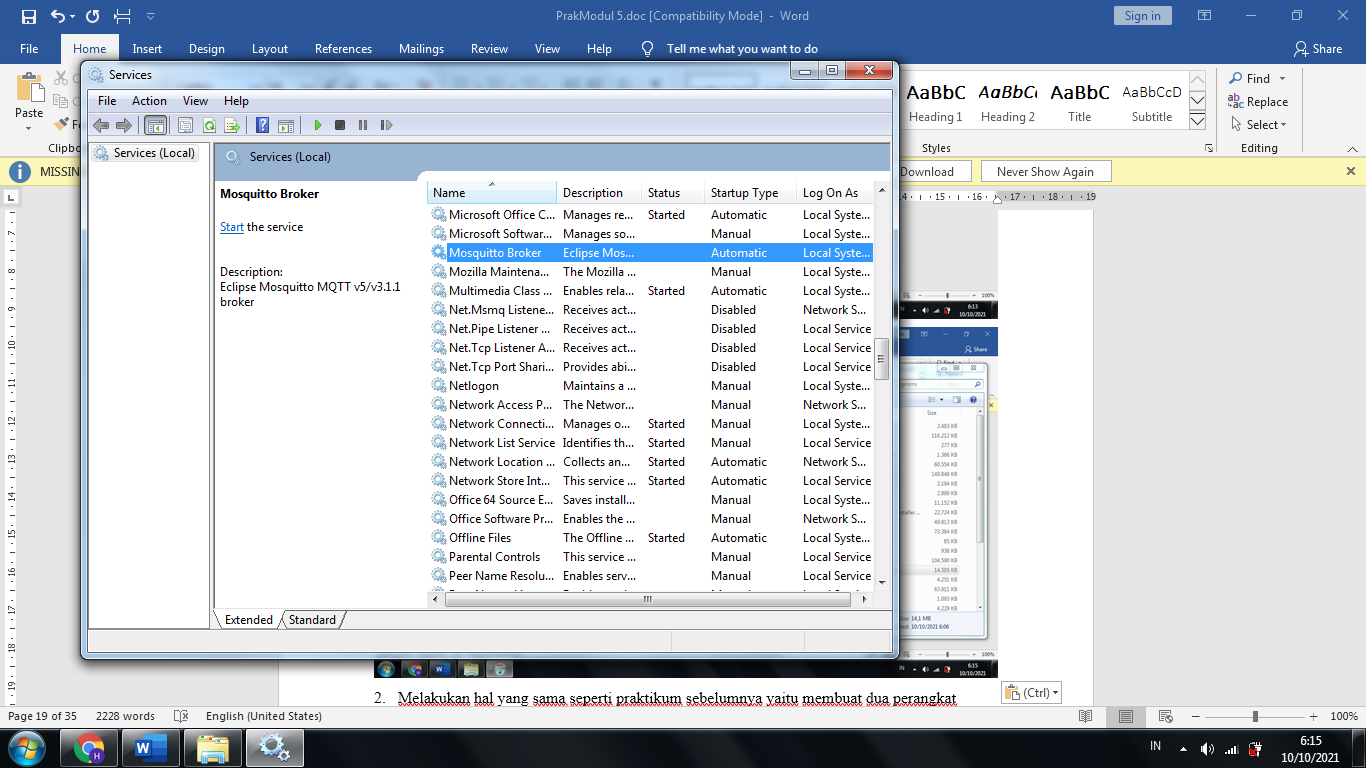
* 1. Berikut proses installasi nya



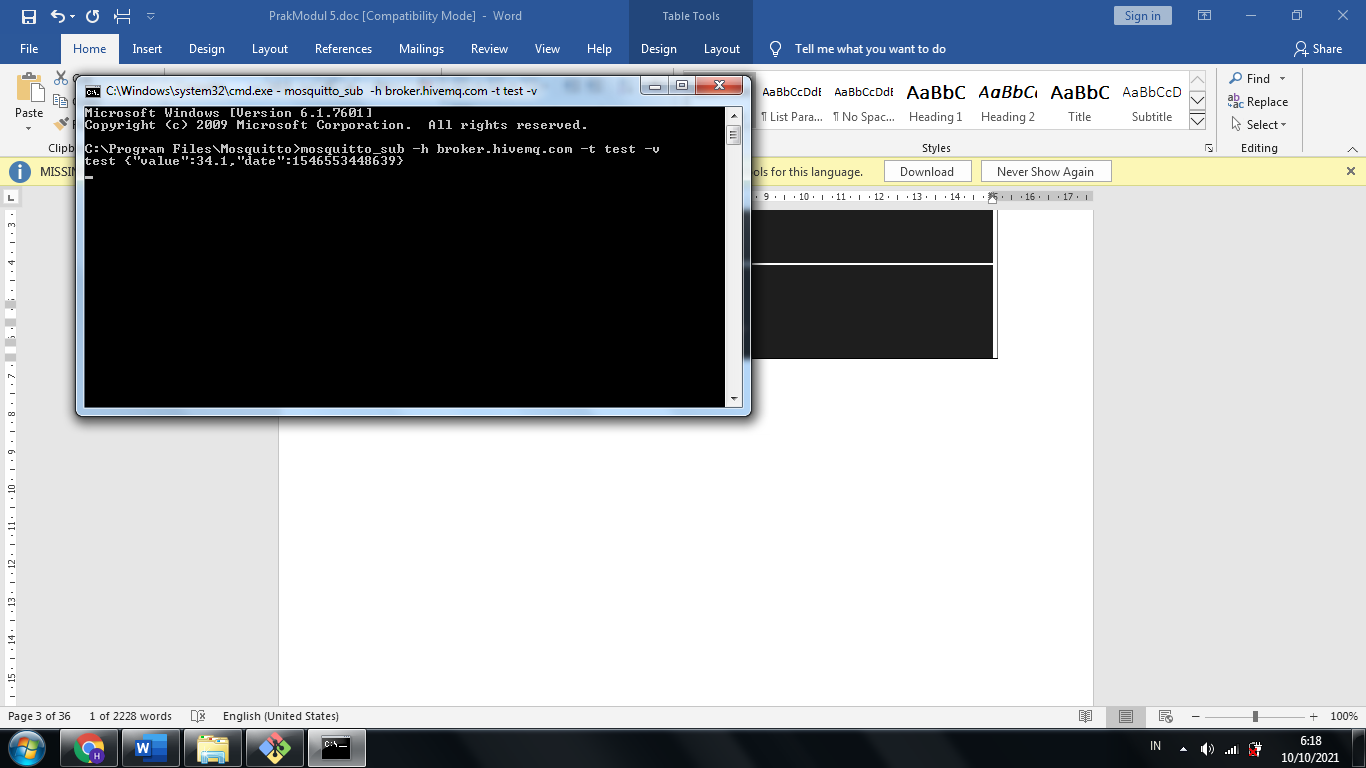
* 1. Setelah itu pilih finish



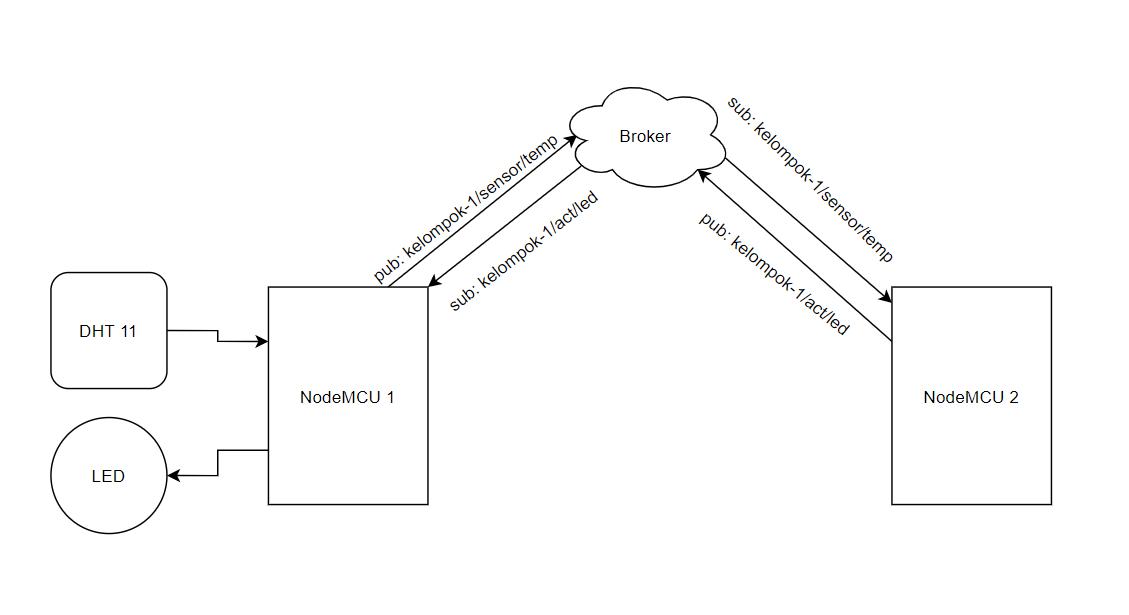
* 1. Setelah itu start mosquitto broker di services.msc



* 1. Hasil Installasinya



1. 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);

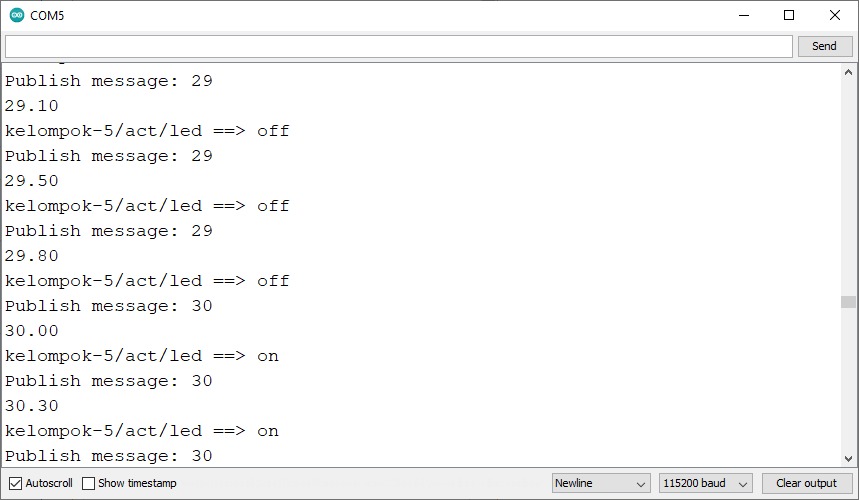
Serial.println(temperature);

client.publish("kelompok-5/sensor/temp", data);

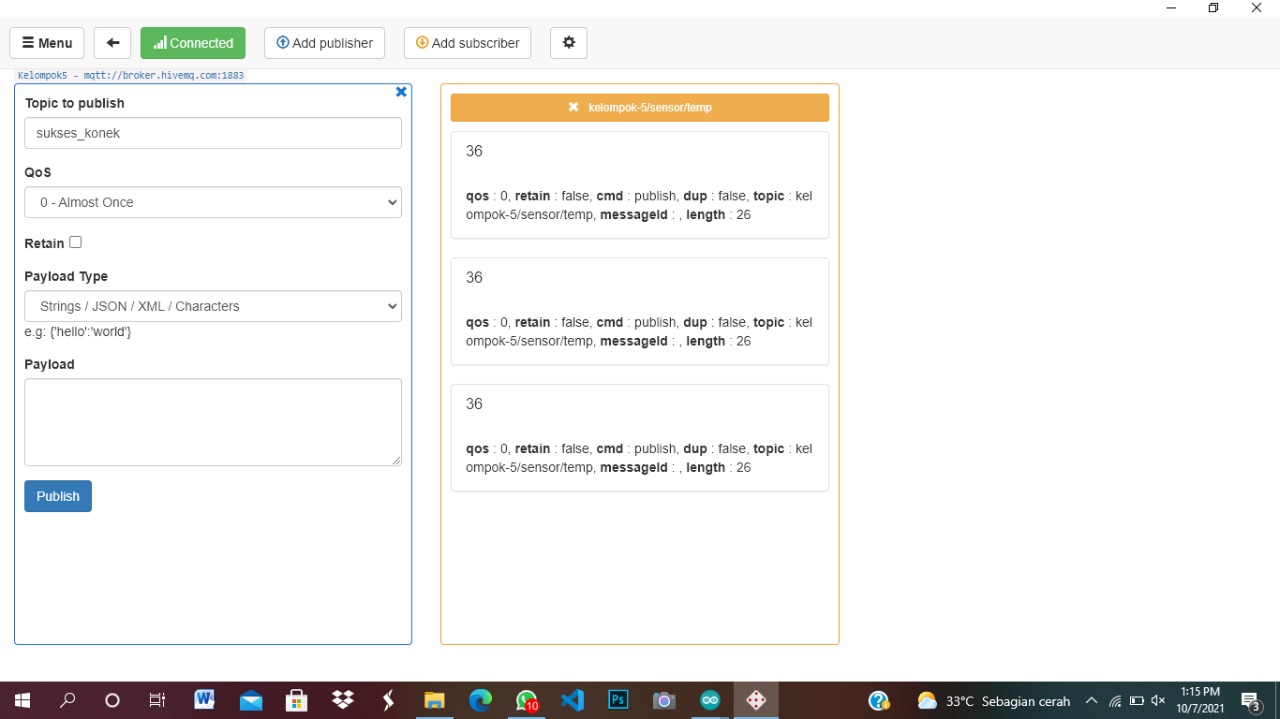
}

}

#Sisi Server



#Sisi server



#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);

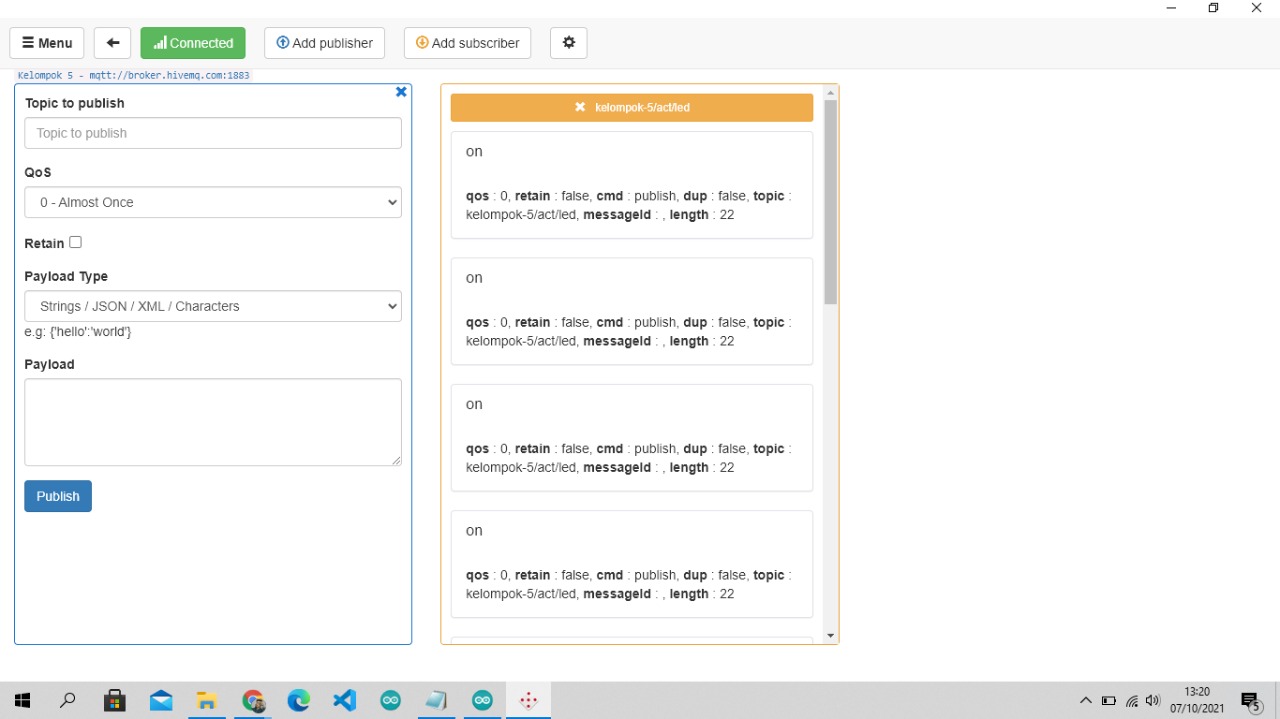
Serial.println(temperature);

client.publish("kelompok-5/sensor/temp", data);

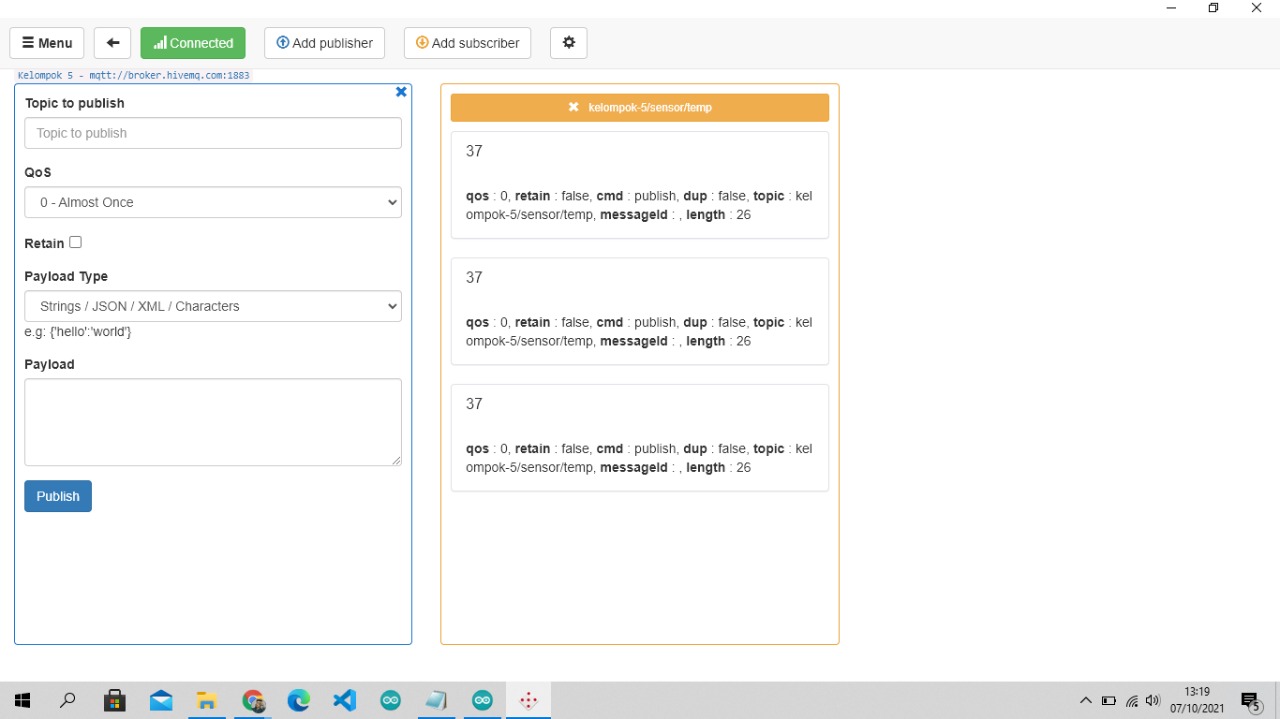
}

}

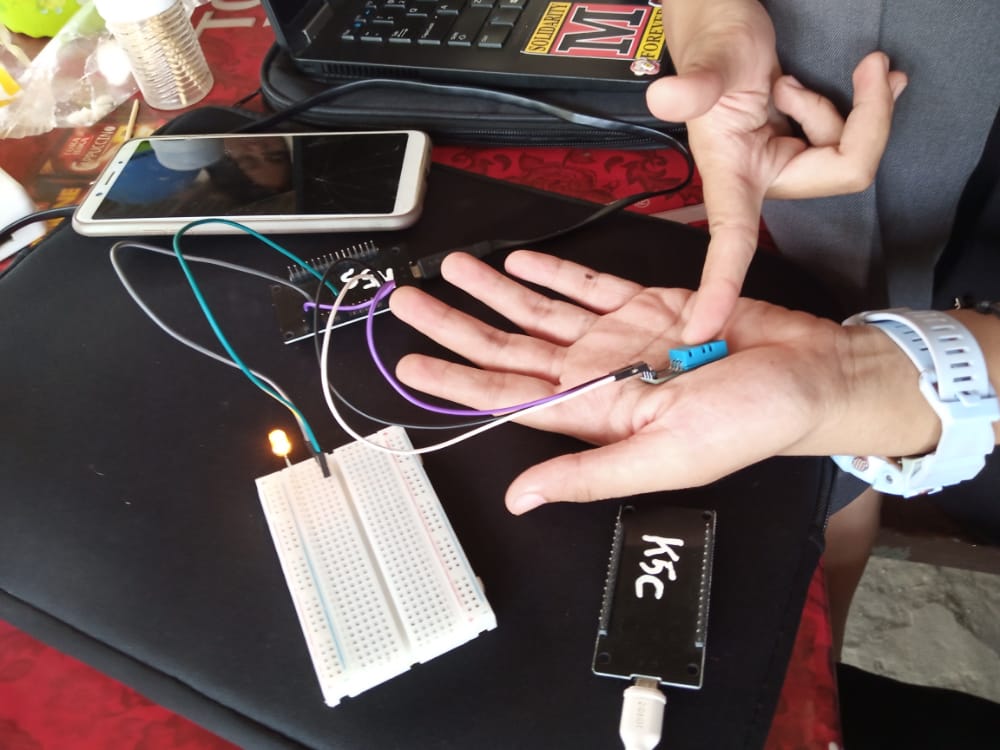
#Sisi Client



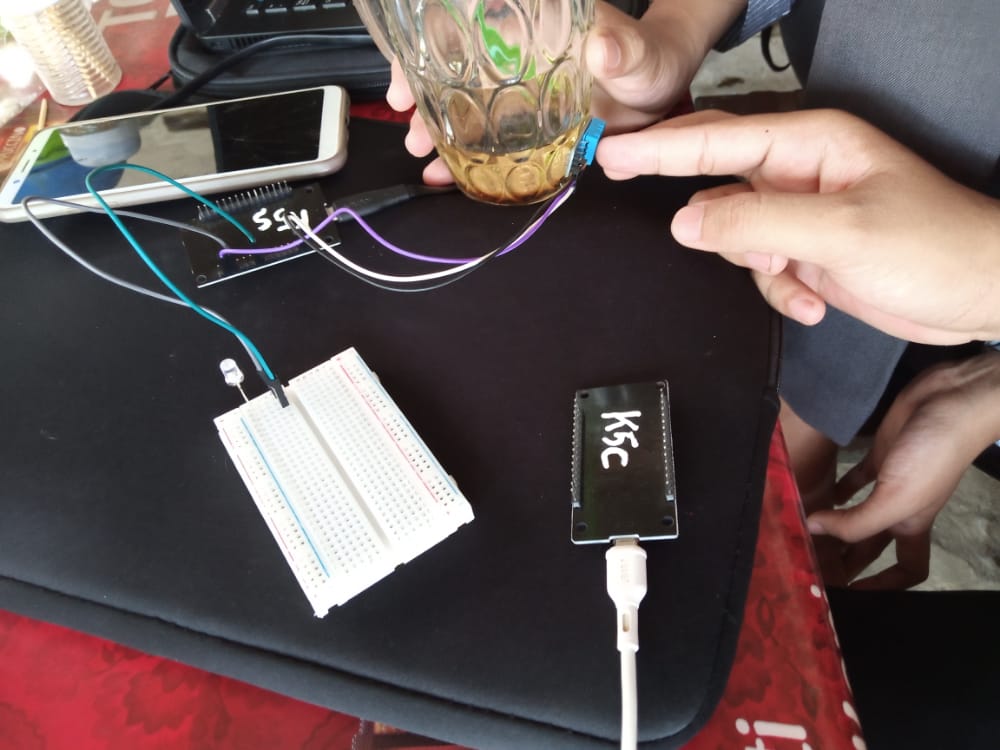
#Sisi Client



#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

1. 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()) {

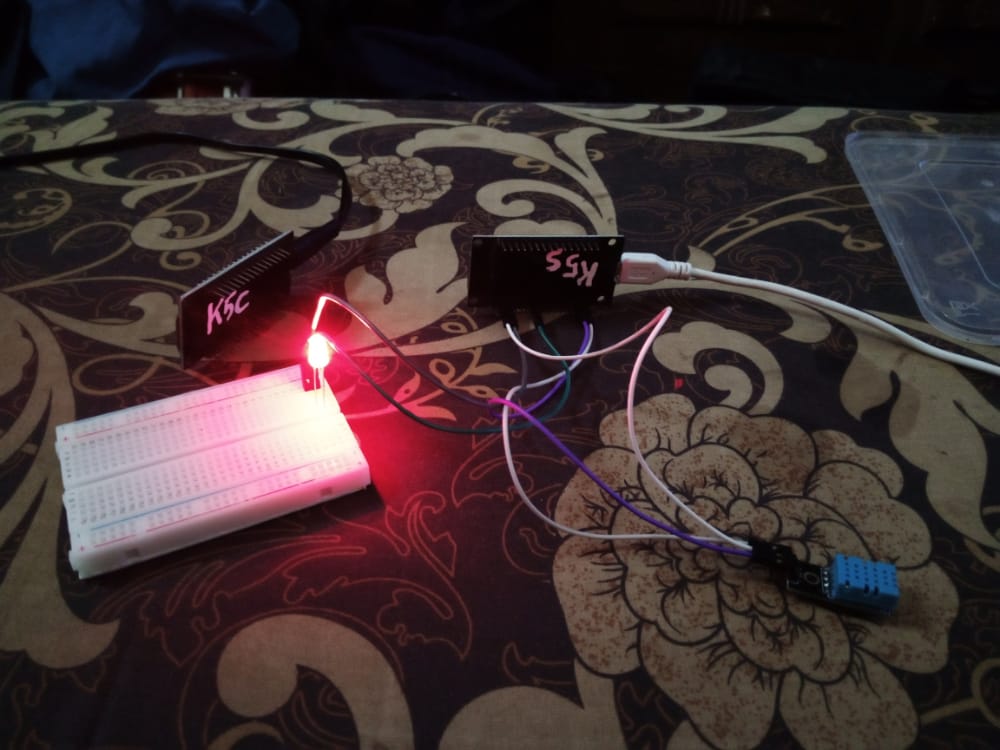
reconnect();

}

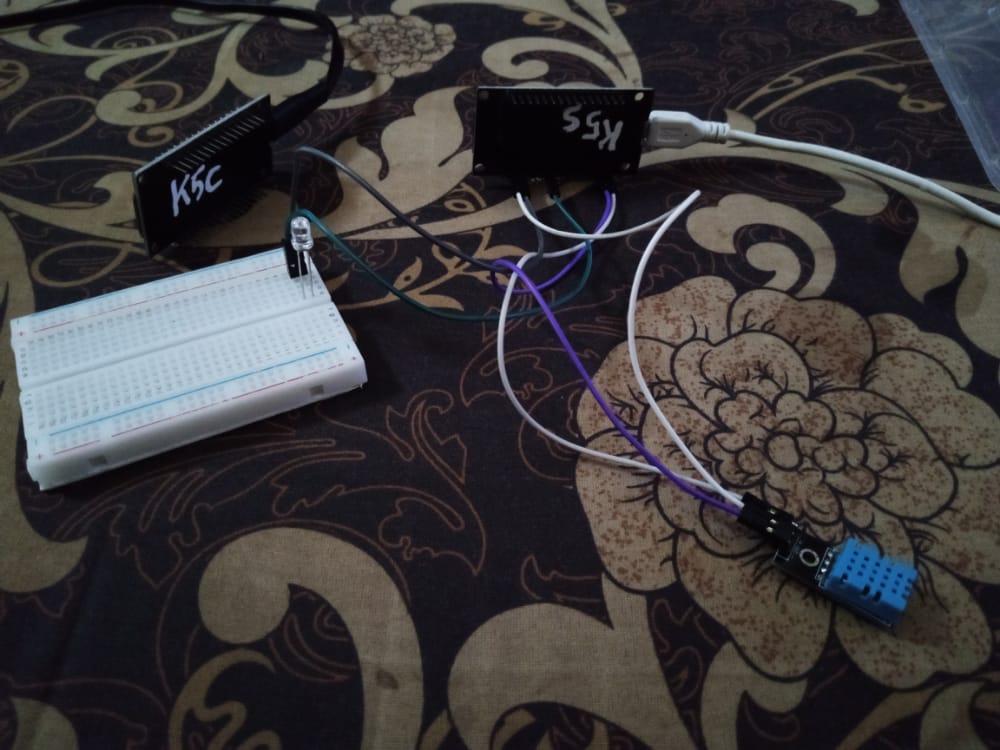
client.loop();

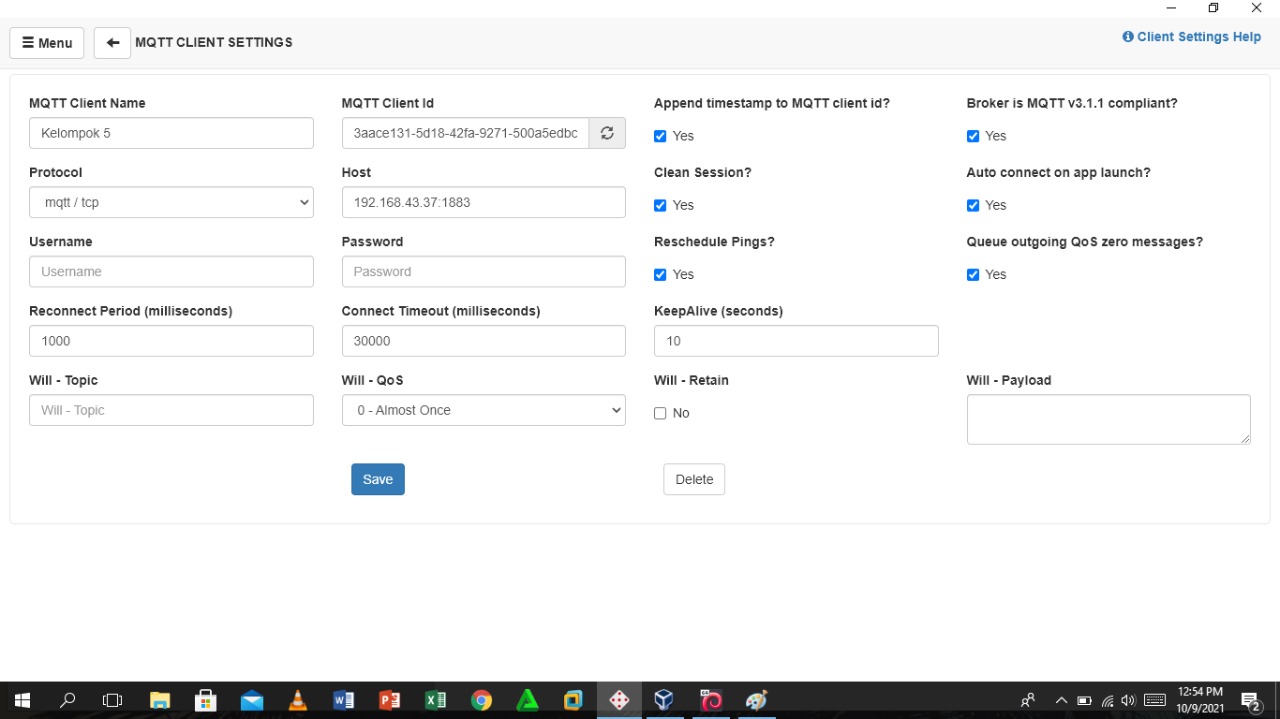
}

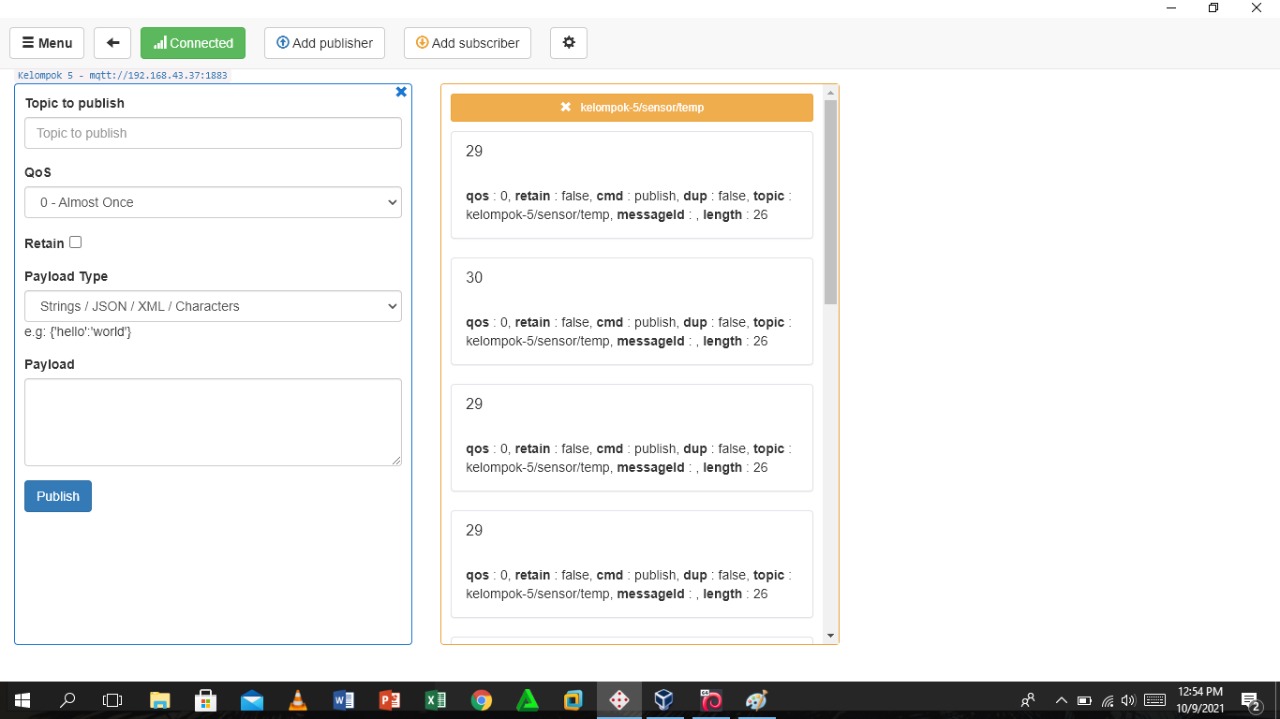
Suhu > 29

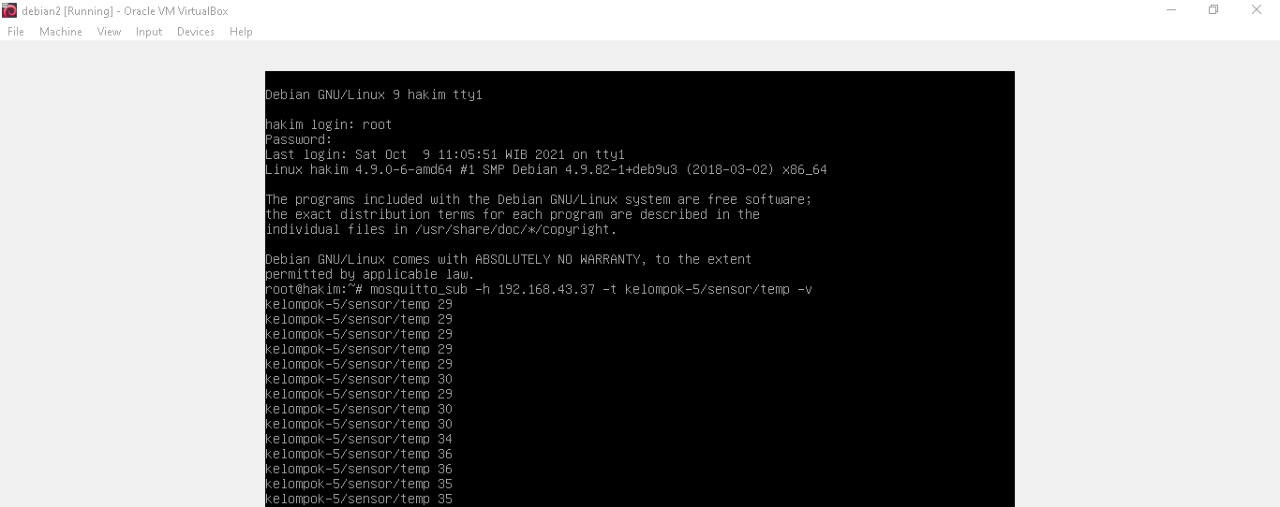


Suhu <29



Settingan MQTTBox local

Hasil



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