

LAPORAN PRAKTIKUM



Kelompok : 5

Nama Anggota : Alif Rizki Ananta Harahap
Andita Farah Salsabila
Hakim Asrori
Mali Nur Al Isthifa
Muhammad Ridzal Maulana
Rantika

Kelas : D3TI.2C

Mata Kuliah : [Internet Of Things](#)

Praktikum ke / Judul : 7/ Penyimpanan Data dan Komunikasi Mqtt

Tanggal Praktikum : 1 November 2021

Dosen Pengampu : Ahmad Rifai, S.Tr.Kom., M.Tr.Kom

MODUL 7 – PENYIMPANAN DATA DAN KOMUNIKASI MQTT

Tujuan

Mahasiswa mampu untuk menyimpan data monitoring dari node IoT ke database MySQL

Mahasiswa mampu untuk menampilkan data yang tersimpan dalam database

Alat dan Bahan

NodeMCU ESP8266

Software Thonny

Visual Code

Library Paho MQTT Python

Library MySQL Connection Python

Arduio IDE

Praktikum

Thonny IDE

Software thonny digunakan untuk menuliskan program dalam bahasa Python. Software ini sangat sederhana dan ringan namun fitur-fiturnya sudah cukup powerfull untuk kegiatan developing program python. Kelebihannya lagi, software ini dapat digunakan untuk memprogram NodeMCU dengan menggunakan script python yang dikenal dengan sebutan MicroPython. Software ini dapat diperoleh dari website officialnya di link berikut ini (<https://thonny.org/>).

Paho MQTT Library

Python merupakan bahasa pemrograman yang memungkinkan untuk menjadi service backend untuk melakukan penyimpanan data monitoring melalui protocol komunikasi MQTT dan menyimpan data monitoring ke database MySQL. Salah satu library yang dapat digunakan untuk berkomunikasi dengan protocol MQTT dalam bahasa Python adalah Python Paho MQTT.

Instalasi library Paho MQTT dapat dilakukan dengan mengetikkan perintah berikut ini di command prompt/terminal: ***pip install paho-mqtt*** apabila telah berhasil terinstall, maka dapat dilakukan tes program dengan program berikut ini.

Sample Paho MQTT library

```
import paho.mqtt.client as mqtt

def on_connect(client, userdata, flags, rc): print("Connected with result code "+str(rc))client.subscribe("$SYS/#")

def on_message(client, userdata, msg): print(msg.topic+" "+str(msg.payload))

client = mqtt.Client() client.on_connect = on_connectclient.on_message = on_message
client.connect("mqtt.eclipseprojects.io", 1883, 60)client.loop_forever()
```

```
C:\Users\hp>pip install paho-mqtt
Collecting paho-mqtt
  Downloading paho-mqtt-1.6.1.tar.gz (99 kB)
    |#####| 99 kB 1.5 MB/s
Building wheels for collected packages: paho-mqtt
  Building wheel for paho-mqtt (setup.py) ... done
  Created wheel for paho-mqtt: filename=paho_mqtt-1.6.1-py3-none-any.whl size=65423 sha256=6ad1633b4ed2a9a8435519f698d2e
835f1d16c09505a33986be40b1d96b6b46b
  Stored in directory: c:\users\hp\appdata\local\pip\cache\wheels\d0\bf\ac\2b3f43f8c6fcd0f4ba5395397458c521eb0b52d33b574
5a40
Successfully built paho-mqtt
Installing collected packages: paho-mqtt
Successfully installed paho-mqtt-1.6.1
WARNING: You are using pip version 20.1.1; however, version 21.3.1 is available.
You should consider upgrading via the 'C:\Users\hp\AppData\Local\Programs\Thonny\python.exe -m pip install --upgrade pip
command.
C:\Users\hp>
```

<pre>1 import paho.mqtt.client as mqtt 2 3 def on_connect(client, userdata, flags, rc): 4 print("Connected with result code "+str(rc)) 5 client.subscribe("sensor/D001") 6 7 def on_message(client, userdata, msg): 8 print(msg.topic+" "+str(msg.payload)) 9 10 client = mqtt.Client() 11 client.on_connect = on_connect 12 client.on_message = on_message 13 client.connect("broker.hivemq.com", 1883, 60) 14 client.loop_forever()</pre>	<pre>sensor/D001 b'60.00#28.30#82.94#301.45' sensor/D001 b'60.00#28.30#82.94#301.45' sensor/D001 b'60.00#28.30#82.94#301.45' sensor/D001 b'60.00#28.30#82.94#301.45' sensor/D001 b'60.00#28.30#82.94#301.45' sensor/D001 b'60.00#28.30#82.94#301.45' sensor/D001 b'60.00#28.30#82.94#301.45' sensor/D001 b'60.00#28.30#82.94#301.45' sensor/D001 b'60.00#28.30#82.94#301.45'</pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

MySQL Library

Agar dapat menyimpan data monitoring yang telah diterima pada fungsi callback subscriber pada program Testing Paho MQTT Library, maka perlu juga library untuk bisa terhubung dengan service database MySQL. Salah satu library yang dapat digunakan untuk terhubung dengan database MQTT adalah Python MySQL Connector. Library tersebut dapat di install dengan perintah seperti berikut ini: ***pip install mysql-connector-python***. Berikut ini adalah contoh program menggunakan library MySQL Connector di Python.

Program Insert ke Database

```
import mysql.connector

mydb = mysql.connector.connect( host="localhost", user="yourusername",
password="yourpassword", database="mydatabase"
)

mycursor = mydb.cursor()

sql = "INSERT INTO customers (name, address) VALUES (%s, %s)"
val = ("John", "Highway 21")
mycursor.execute(sql, val)

mydb.commit()

print(mycursor.rowcount, "record inserted.")

mydb.cursor()
```

Program Select Data

```
import mysql.connector

mydb = mysql.connector.connect( host="localhost", user="yourusername",
password="yourpassword", database="mydatabase"
)

mycursor = mydb.cursor() mycursor.execute("SELECT * FROM customers") myresult =
mycursor.fetchall()
for x in myresult:print(x)
```

```
C:\Users\hp>pip install mysql-connector-python
Collecting mysql-connector-python
  Downloading mysql_connector_python-8.0.27-1commercial-py2.py3-none-any.whl (302 kB)
    |#####| 302 kB 2.2 MB/s
Collecting protobuf>=3.0.0
  Downloading protobuf-3.19.1-cp37-cp37m-win32.whl (776 kB)
    |#####| 776 kB 6.8 MB/s
Installing collected packages: protobuf, mysql-connector-python
Successfully installed mysql-connector-python-8.0.27 protobuf-3.19.1
WARNING: You are using pip version 20.1.1; however, version 21.3.1 is available.
You should consider upgrading via the 'C:\Users\hp\AppData\Local\Programs\Thonny\python.exe -m pip install --upgrad
' command.
```

Program Insert ke Database

```
1 import mysql.connector
2
3
4 mydb = mysql.connector.connect(
5     host="localhost",
6     user="root",
7     password="",
8     database="kelompok5"
9 )
10
11
12 mycursor = mydb.cursor()
13 sql = "INSERT INTO admin (nama, alamat) VALUES (%s, %s)"
14 val = ("Andita", "Indramayu")
15 mycursor.execute(sql, val)
16
17 mydb.commit()
18
19 print(mycursor.rowcount, "record inserted.")
20
```

✓ Showing rows 0 - 0 (1 total, Query took 0.0019 seconds.)

SELECT * FROM `admin`

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: 25 ▼ Filter rows:

+ Options

	id	nama	alamat
<input type="checkbox"/>		Andita	Indramayu

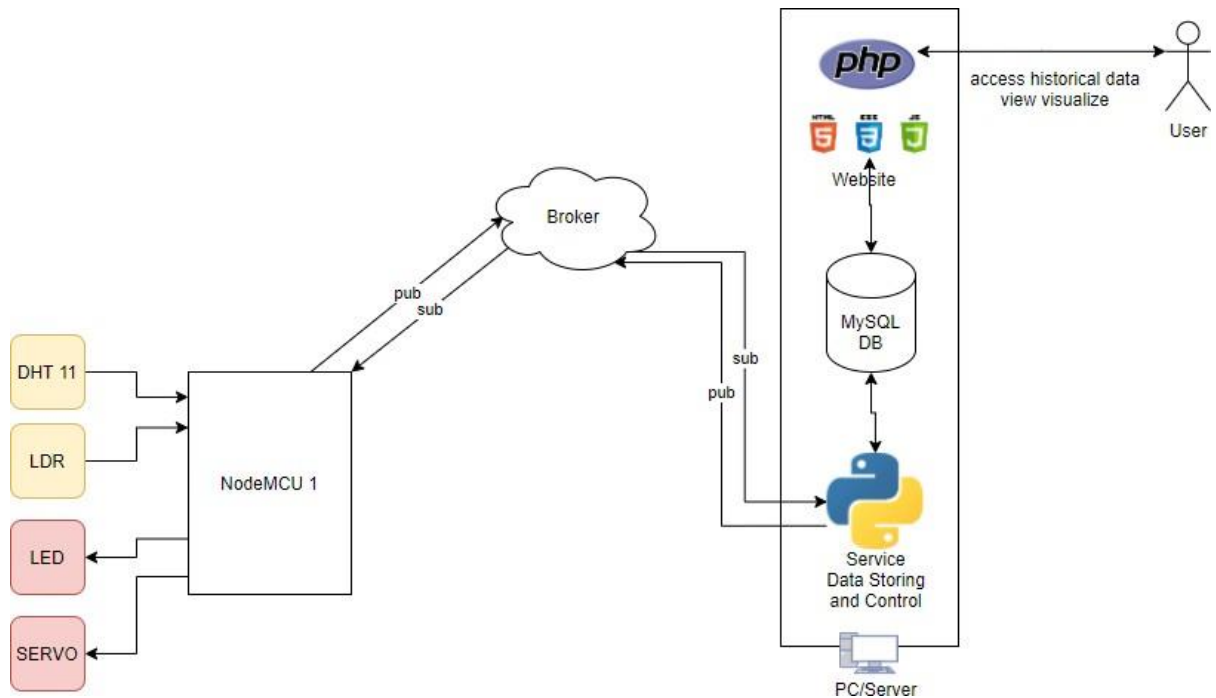
Program Select Data

```
1 import mysql.connector
2
3 mydb = mysql.connector.connect(
4     host="localhost",
5     user="root",
6     password="",
7     database="kelompok5"
8 )
9
10 mycursor = mydb.cursor()
11 mycursor.execute("SELECT * FROM admin")
12 myresult = mycursor.fetchall()
13 for x in myresult:
14     print(x)
15
```

```
>>> %Run 'Program Select Data.py'
(1, 'Andita', 'Indramayu')
>>>
```

Tugas

Berikut ini adalah desain sistem IoT yang bertujuan untuk menyimpan data monitoring suhu, kelembaban dan intensitas cahaya dalam suatu rumah (Smart Home). Data monitoring tersebut akan disimpan dengan menggunakan service Python menggunakan protokol MQTT agar dapat berkomunikasi dengan Node sensor. Service tersebut berfungsi sebagai gateway untuk penyimpanan data dan juga digunakan sebagai sistem pendukung keputusan (mengontrol LED dan Servo). Apabila telah tersimpan ke database, user dapat melihat data historical-nya dengan mengakses website yang dibuat dengan menggunakan script-script dalam pembuatan website (PHP, HTML, JavaScript).



Buatkan program service Data Storing dan Control menggunakan bahasa Python. Dan buatkan Website untuk menampilkan data monitoring yang sudah tersimpan di database menggunakan script PHP.

Fotokan/screenshoot hasil dari program yang telah dibuat ! Contoh: Respon data yang diterimadan dipublish dalam service Data Storing dan Control dll.

JAWAB

1.

Kode program tony

```
import paho.mqtt.client as mqtt
import mysql.connector
import json
```

```
def on_connect(client, userdata, flags, rc):
    print("Connected with result code "+str(rc))
    client.subscribe("kelompok-5/sensor/temp")
```

```
def on_message(client, userdata, msg):
    txt = msg.payload
    txt2 = txt.decode('UTF-8')
    split = txt2.split("#")
```

```
servo = ""
led = ""
```

```
if(float(split[0]) > 29):
    servo = 90
```

```

else:
    servo = 0

if(int(split[1]) > 200):
    led = "ON"
else:
    led = "OFF"

mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="coba_db"
)

mycursor = mydb.cursor()

sql = "INSERT INTO coba (temperature, ldr, status_servo, status_led) VALUES (%s, %s, %s, %s)"
val = (split[0], split[1], servo, led)
mycursor.execute(sql, val)
mydb.commit()

data = json.dumps({'servo': servo, 'led': led})
print(data)
client.publish("kelompok-5/act/led", data)

client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message
client.connect("broker.hivemq.com", 1883, 60)
client.loop_forever()

```

Kode program Arduino

```

#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include "DHT.h"
#include <Servo.h>
#include <ArduinoJson.h>

// Instansiasi library-library
Servo myservo;
WiFiClient espClient;
PubSubClient client(espClient);

// Pendeklarasian dari WIFI SSID, PASS hingga String buffData
#define WIFI_SSID "ELDAS_WIFI"
#define WIFI_PASS ""
#define MQTT_SERVER "broker.hivemq.com"
#define MQTT_PORT 1883
#define DHTPIN D2
#define DHTTYPE DHT11
#define PIN_LDR A0

```



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

```
DHT dht(DHTPIN, DHTTYPE);
```

```
String buffData;
```

```
// Fungsi yang berguna untuk mengkoneksikan ke jaringan yang dimaksud pada deklarasi wifi
```

```
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());
}
```

```
// fungsi yang menerima payload/data yang dikirimkan oleh broker
```

```
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);
```

```
// instansiasi staticjsondocument
StaticJsonDocument<200> doc;
deserializeJson(doc,payload);
```

```
int rotate = doc["servo"];
```

```
myservo.write(rotate);
```

```
if(doc["led"] == "HIGH")
  digitalWrite(D3, HIGH);
else
  digitalWrite(D3, LOW);
}
```

```
// fungsi yang berguna untuk cek konektivitas dari brokernya
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);
  }
}
}
```

```
// fungsi yang berguna untuk men setup seperti setup_wifi dan lainnya
void setup() {
```

```
  Serial.begin(57600);
  setup_wifi();
  pinMode(D3, OUTPUT);
  myservo.attach(D1);
  client.setServer(MQTT_SERVER, MQTT_PORT);
  client.setCallback(callback);
  dht.begin();
}
```

```
// fungsi yang berguna untuk mengirimkan data berulang-ulang kali
void loop() {
```

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

```
// fungsi yang mengirim data sensor
```

```
void kirimPer2Detik(){
  _now = millis();
  if(millis() - _waiting > 2000){
    _waiting = _now;

    sprintf(data, "%g#%i",
      dht.readTemperature(),
      analogRead(PIN_LDR)
    );
  }
}
```

```

Serial.print("Publish message: ");
Serial.println(data);
client.publish("kelompok-5/sensor/temp", data);
}
}

```

- Serial monitor

```

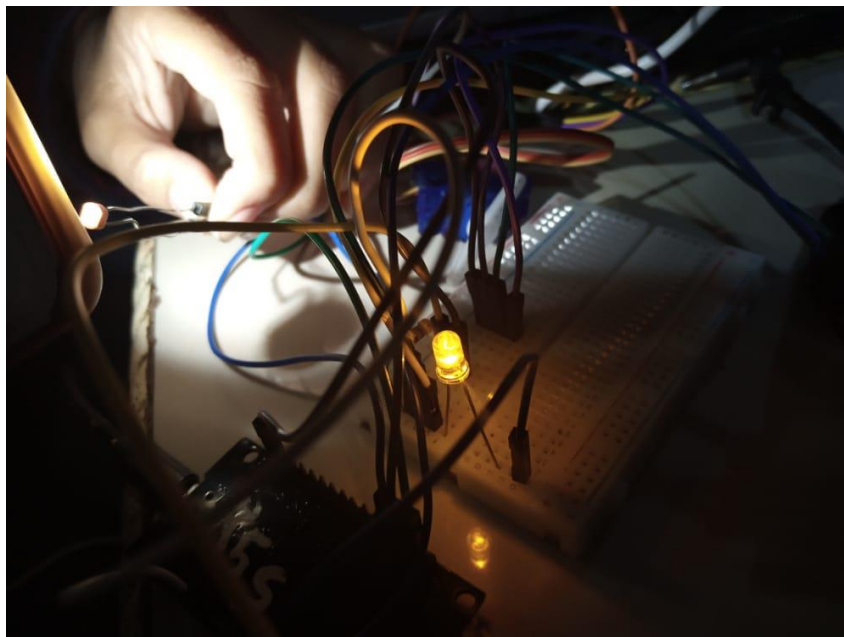
Publish message: 29.8#21
kelompok-5/act/led ==> {"servo": 90, "led": "ON"}
Publish message: 29.8#21
kelompok-5/act/led ==> {"servo": 90, "led": "OFF"}
Publish message: 29.8#14
kelompok-5/act/led ==> {"servo": 90, "led": "OFF"}
Publish message: 29.8#10
kelompok-5/act/led ==> {"servo": 90, "led": "OFF"}
Publish message: 29.8#10
kelompok-5/act/led ==> {"servo": 90, "led": "OFF"}
Publish message: 29.8#10
kelompok-5/act/led ==> {"servo": 90, "led": "OFF"}
Publish message: 29.8#10
kelompok-5/act/led ==> {"servo": 90, "led": "OFF"}
Publish message: 29.8#13
kelompok-5/act/led ==> {"servo": 90, "led": "OFF"}
Publish message: 29.8#12
kelompok-5/act/led ==> {"servo": 90, "led": "OFF"}

```

Gambar 1.1 Serial Monitor

Keterangan Gambar 1.1 -> Serial monitor ini menampilkan data yang dihasilkan oleh kode program kemudian di up ke nodemcu

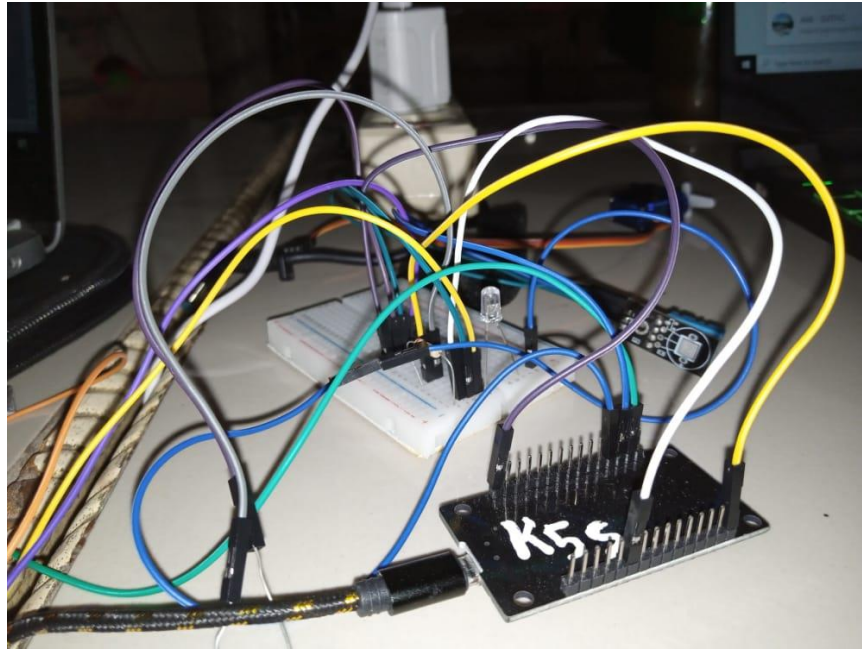
- Led on



Gambar 1.2 LedOn

Keterangan Gambar 1.2-> LED On Ketika sensor LDR mendapatkan nilai lebih dari 200

- Led Off



Gambar 1.3 LED Off

Keterangan Gambar 1.3 ->LED OFF Ketika sensor LDR mendapatkan nilai kurang dari 200

- Serial Monitor

```
Publish message: 29.5#13
kelompok-5/act/led ==> {"servo": 0, "led": "OFF"}
Publish message: 29.5#12
kelompok-5/act/led ==> {"servo": 0, "led": "OFF"}
kelompok-5/act/led ==> {"servo": 0, "led": "OFF"}
Publish message: 29.5#12
kelompok-5/act/led ==> {"servo": 0, "led": "OFF"}
Publish message: 29.5#12
kelompok-5/act/led ==> {"servo": 0, "led": "OFF"}
Publish message: 29.5#12
kelompok-5/act/led ==> {"servo": 0, "led": "OFF"}
Publish message: 29.5#12
Publish message: 29.5#12
kelompok-5/act/led ==> {"servo": 90, "led": "OFF"}
Publish message: 29.6#12
kelompok-5/act/led ==> {"servo": 90, "led": "OFF"}
Publish message: 29.5#14
kelompok-5/act/led ==> {"servo": 90, "led": "OFF"}
Publish message: 29.5#18
```

Gambar 1.4 Serial Monitor

Keterangan Gambar 1.4 -> Serial monitor ini menampilkan data yang diterima oleh sensor dan dikirimkan oleh mqtt dengan topik kelompok-5/act/led

- Servo



Gambar 1.5 servo 0

Keterangan Gambar 1.5-> Servo 0 derajat Ketika sensor DHT mendapatkan nilai kurang dari 29



Gambar 1.6 Servo 90

Keterangan Gambar 1.6 -> Servo 90 derajat Ketika sensor DHT mendapatkan nilai lebih dari 29

- Tampilan WEB

Data Suhu	Data LDR	Status Servo	Status LED
29.4	269	90	ON
29.4	368	90	ON
29.4	268	90	ON
29.3	405	90	ON
29.4	115	90	OFF
29.4	90	90	OFF
29.4	98	90	OFF
29.4	368	90	ON
29.4	268	90	ON
29.3	405	90	ON
29.4	115	90	OFF
29.4	90	90	OFF
29.4	90	90	OFF

Gambar 1.7 Tampilan WEB

Keterangan : Jadi gambar ini menampilkan data yang diupload ke database mysql

2.

- Python

```

cobap.py
1 def on_connect(client, userdata, flags, rc):
2     print("Connected with result code "+str(rc))
3     client.subscribe("kelompok-5/sensor/temp")
4
5 def on_message(client, userdata, msg):
6     txt = msg.payload
7     txt2 = txt.decode('UTF-8')
8     split = txt2.split("#")
9
10    print("Hasil yang diterima : ")
11    print(txt)
12
13    servo = ""
14    led = ""
15
16    if(float(split[0]) > 29):
17        servo = "ON"
18        led = "ON"
19    else:
20        servo = "OFF"
21        led = "OFF"
22
23    print("Servo: " + servo + ", Led: " + led)
24
25 client.on_connect = on_connect
26 client.on_message = on_message
27 client.connect("127.0.0.1", 1883, 60)
28 client.loop_forever()

```

Shell

```

Connected with result code 0
Hasil yang diterima :
b'29.0#10'
("servo": 90, "led": "OFF")
Hasil yang diterima :
b'29.7#26'
("servo": 90, "led": "OFF")
Hasil yang diterima :
b'29.0#24'
("servo": 90, "led": "OFF")
Hasil yang diterima :
b'29.0#15'
("servo": 90, "led": "OFF")
Hasil yang diterima :
b'29.0#19'
("servo": 90, "led": "OFF")
Hasil yang diterima :
b'29.0#17'
("servo": 90, "led": "OFF")

```

Assistant

TypeError: can only concatenate str (not "bytes") to str
cobap.py, line 14
Python was asked to do an operation with an object which doesn't support it.
Did you expect another type?
Maybe you forgot some details about this operation?

Warnings
May help you find the cause of the error.
cobap.py
Line 5: Redefining name 'client' from outer scope (line 47)
Line 5: Unused argument 'userdata'
Line 5: Unused argument 'flags'
Line 9: Redefining name 'client' from outer scope (line 47)
Line 9: Unused argument 'userdata'
Line 20: Incompatible types in assignment (expression has type "int", variable has type "str")
Line 22: Incompatible types in assignment (expression has type "int", variable has type "str")
Was it helpful or confusing?
General advice on fixing errors

Gambar 2.1 Python

Keterangan : di gambar tersebut merupakan respon dari protocol mqtt yang dikirim melalui nodemcu