

LAPORAN PRAKTIKUM MINGGU KE-14

IoT Gateway dan Message Broker INTERNET OF THINGS



Disusun oleh:

Omar Abdul-Raof Taha Ghaleb Al-Maktary

1941720237

TI-3H

D4 TEKNIK INFORMATIKA

TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2022

LAPORAN

A. PRAKTIKUM

First step is to configure the instance and download necessary packages such as php and composer:

```
ubuntu@iot-instance:~$ #!/bin/bash
ubuntu@iot-instance:~$ sudo apt update
Hit:1 http://ap-sydney-1-ad-1.clouds.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ap-sydney-1-ad-1.clouds.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ap-sydney-1-ad-1.clouds.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
15 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@iot-instance:~$ sudo apt install apache2 php libapache2-mod-php -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
```

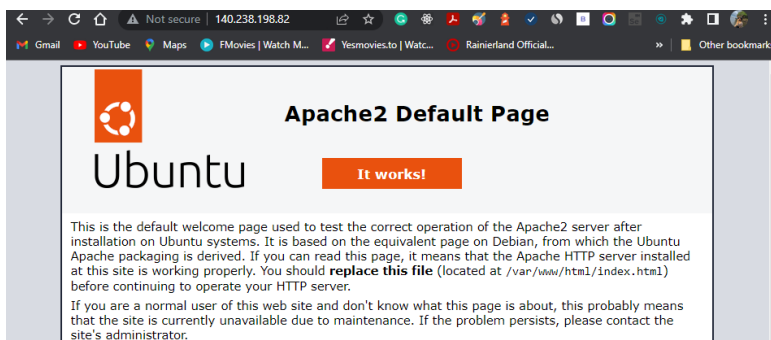
Running Apache on port 80:

```
ubuntu@iot-instance:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2022-06-26 16:26:27 UTC; 7min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 891 (apache2)
    Tasks: 6 (limit: 1082)
  Memory: 24.9M
     CPU: 394ms
   CGroup: /system.slice/apache2.service
           └─891 /usr/sbin/apache2 -k start
             └─893 /usr/sbin/apache2 -k start
               └─894 /usr/sbin/apache2 -k start
                 └─895 /usr/sbin/apache2 -k start
                   └─896 /usr/sbin/apache2 -k start
                     └─897 /usr/sbin/apache2 -k start

Jun 26 16:26:25 iot-instance systemd[1]: Starting The Apache HTTP Server...
Jun 26 16:26:27 iot-instance apachectl[871]: AH00558: apache2: Could not reliably determine the
Jun 26 16:26:27 iot-instance systemd[1]: Started The Apache HTTP Server.

ubuntu@iot-instance:~$ sudo iptables -I INPUT 6 -m state --state NEW -p tcp --dport 80 -j ACCEPT

ubuntu@iot-instance:~$ sudo netfilter-persistent save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables save
ubuntu@iot-instance:~$
```



Install codeigniter using composer:

```
ubuntu@iot-instance:~$ composer create-project codeigniter4/appstarter iot-jti --no-dev
Creating a "codeigniter4/appstarter" project at "./iot-jti"
Info from https://repo.packagist.org: #StandWithUkraine
Installing codeigniter4/appstarter (v4.2.1)
- Downloading codeigniter4/appstarter (v4.2.1)
- Installing codeigniter4/appstarter (v4.2.1): Extracting archive
Created project in /home/ubuntu/iot-jti
Loading composer repositories with package information
Updating dependencies
Lock file operations: 41 installs, 0 updates, 0 removals
- Locking codeigniter4/framework (v4.2.1)
```

Move the new project to www directory and change its permissions and ownership:

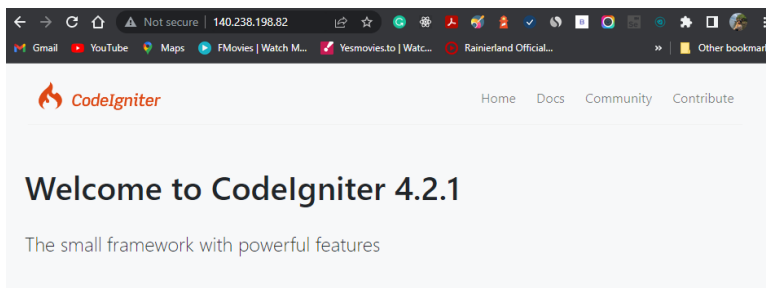
```
ubuntu@iot-instance:~$ ls
iot-jti lib node_modules package.json settings.js
ubuntu@iot-instance:~$ sudo chown -Rv www-data iot-jti/writable
sudo mv iot-jti/ /var/www/html/
changed ownership of 'iot-jti/writable/debugbar/.gitkeep' from ubuntu to www-data
changed ownership of 'iot-jti/writable/debugbar' from ubuntu to www-data
changed ownership of 'iot-jti/writable/session/index.html' from ubuntu to www-data
changed ownership of 'iot-jti/writable/session' from ubuntu to www-data
changed ownership of 'iot-jti/writable/uploads/index.html' from ubuntu to www-data
changed ownership of 'iot-jti/writable/uploads' from ubuntu to www-data
changed ownership of 'iot-jti/writable/.htaccess' from ubuntu to www-data
changed ownership of 'iot-jti/writable/logs/index.html' from ubuntu to www-data
changed ownership of 'iot-jti/writable/logs' from ubuntu to www-data
changed ownership of 'iot-jti/writable/cache/index.html' from ubuntu to www-data
changed ownership of 'iot-jti/writable/cache' from ubuntu to www-data
changed ownership of 'iot-jti/writable' from ubuntu to www-data
```

Change hosting directory:

```
GNU nano 6.2 /etc/apache2/sites-available/iot-jti.conf *
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/iot-jti/public
```

The website is working:



Change environment type:

```
#-----
# ENVIRONMENT
#-----

# CI_ENVIRONMENT = development
```

Download mosquito from a repository and install it:

```
ubuntu@iot-instance:~$ sudo apt-add-repository ppa:mosquitto-dev/mosquitto-ppa -y
sudo apt-get update
sudo apt-get install mosquitto mosquitto-clients -y
[]

ubuntu@iot-instance:~$ sudo iptables -I INPUT 6 -m state --state NEW -p tcp --dport 1883 -j ACCEPT
PT
sudo iptables -I INPUT 6 -m state --state NEW -p tcp --dport 8090 -j ACCEPT
sudo netfilter-persistent save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables save
```

Try to send data with mosquito

```
ubuntu@iot-instance:~$ mosquitto_pub -t "iot/Omar" -m "iot Omar Al-Maktary"
ubuntu@iot-instance:~$ []
```

```
ubuntu@iot-instance:~$ mosquitto_sub -t "iot/Omar"
iot Omar Al-Maktary
[]
```

Add a password:

```
ubuntu@iot-instance:~$ sudo mosquitto_passwd -c /etc/mosquitto/passwd jti
Password:
Reenter password:
```

Add config file:

```

GNU nano 6.2 /etc/mosquitto/conf.d/jti.conf *
listener 8090
protocol websockets

listener 1883
#protocol websockets

per_listener_settings true
allow_anonymous false
password_file /etc/mosquitto/passwd

```

Check results:

```

ubuntu@iot-instance:~$ mosquitto_pub -h localhost -t "test" -m "hello world"
Connection error: Connection Refused: not authorised.
Error: The connection was refused.

```

```

ubuntu@iot-instance:~$ mosquitto_pub -h localhost -t "iot/Omar" -m "hello world" -u jti -P 12345

```

```

ubuntu@iot-instance:~$ mosquitto_sub -t "iot/Omar" -u jti -P 123456789
hello world

```

Question

- What is the function of the command line websockets protocol in the mosquitto config file?
By using websockets we can add other ports to be used as a client to connect to the message broker,
- Please change to false on per_listener_settings true, restart mosquitto. What will happen or what will the effect be?

```

GNU nano 6.2 /etc/mosquitto/conf.d/jti.conf *
listener 8090
protocol websockets

listener 1883
#protocol websockets

per_listener_settings false
allow_anonymous false
password_file /etc/mosquitto/passwd

```

```

ubuntu@iot-instance:~$ mosquitto_pub -t "iot/Omar" -m "Whats up" -u jti -P 123456789
ubuntu@iot-instance:~$ mosquitto_pub -t "iot/Omar" -m "Whats up" -u jti -P 123456789
ubuntu@iot-instance:~$

```

```

ubuntu@iot-instance:~$ sudo nano /etc/mosquitto/conf.d/jti.conf
ubuntu@iot-instance:~$ sudo systemctl restart mosquitto
ubuntu@iot-instance:~$ mosquitto_sub -t "iot/Omar" -u jti -P 123456789
Whats up
Whats up

```

Connect to a smart device web application:

CODE to ESP8266:

```

#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <SimpleDHT.h>

const char *ssid = "WI-FI";
const char *password = "Password";

```

```

const char *mqtt_server = "http://140.238.198.82/";

WiFiClient espClient;
PubSubClient client(espClient);

SimpleDHT11 dht11(D7);

Long now = millis();
Long lastMeasure = 0;
String macAddr = "";

void setup_wifi()
{
    delay(10);
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.print("WiFi connected - ESP IP address: ");
    Serial.println(WiFi.localIP());
    macAddr = WiFi.macAddress();
    Serial.println(macAddr);
}

void reconnect()
{
    while (!client.connected())
    {
        Serial.print("Attempting MQTT connection...");
        if (client.connect(macAddr.c_str()))
        {
            Serial.println("connected");
        }
        else
        {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5 seconds");
            delay(5000);
        }
    }
}

```

```

void setup()
{
  Serial.begin(9600);
  Serial.println("Mqtt Node-RED");
  setup_wifi();
  client.setServer(mqtt_server, 1883);
}

void loop()
{
  if (!client.connected())
  {
    reconnect();
  }
  if (!client.loop())
  {
    client.connect(macAddr.c_str());
  }
  now = millis();
  if (now - lastMeasure > 5000)
  {
    lastMeasure = now;
    int err = SimpleDHTerrSuccess;

    byte temperature = 0;
    byte humidity = 0;
    if ((err = dht11.read(&temperature, &humidity, NULL)) !=
SimpleDHTerrSuccess)
    {
      Serial.print("Reading DHT11 failed, err=");
      Serial.println(err);
      delay(1000);
      return;
    }
    static char temperatureTemp[7];
    dtostrf(temperature, 4, 2, temperatureTemp);
    Serial.println(temperatureTemp);

    client.publish("room/temperature ", temperatureTemp);
  }
}

```

Edit Index file:

```

GNU nano 6.2 /var/www/html/index.html *
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>JTI IoT</title>
  <script type="text/javascript" src="http://ajax.googleapis.com/ajax/1
  <script src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/m
  <script type="text/javascript">
    var MQTTbroker = 'http://140.238.198.82/'; //servernya disesuaikan
    var MQTTport = 1883; //sesuaikan port websockets message broker,
    var MQTTsubTopic = 'room/temperature'; //topiknya perlu disesuaikan
    var chart; // global variuable for chart
    var dataTopics = new Array();

```

Edit esp code to accept user and password

```

void reconnect()
{
  while (!client.connected())
  {
    Serial.print("Attempting MQTT connection...");
    if (client.connect(macAddr.c_str(), MQTT_USER, MQTT_PASS))
    {
      Serial.println("connected");
    }
    else
    {

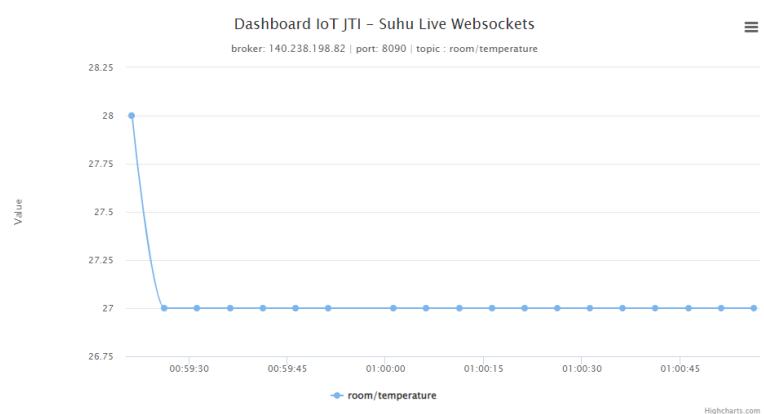
```

```

MQTT Node-RED
Connecting to P.TRIS
.....
WiFi connected - ESP IP address: 192.168.100.70
44:17:93:20:03:CC
Attempting MQTT connection...connected
28.00
27.00
27.00
27.00
27.00

```

Results:



```

mqtt connected
room/temperature 28.00
is a proper number, will send to chart.
(2) [1656266361283, 28]
room/temperature 27.00
is a proper number, will send to chart.
(2) [165626636263, 27]
room/temperature 27.00

```