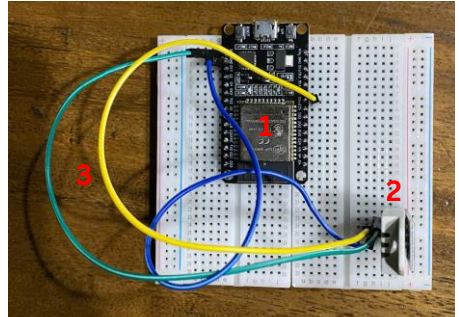


TUGAS UTS:
“SISTEM BERBASIS MIKROPROSESOR”
SEMESTER GENAP 2023/2024

Ryan Krishandi Lukito (22/497249/TK/54488)
Yefta Nathaniel Wibowo (22/492878/TK/53954)

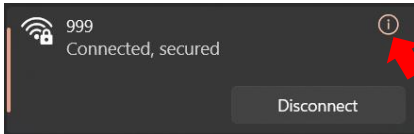
Konfigurasi Rangkaian ESP32 dan DHT-22

1. ESP32 ditempatkan di bagian mana saja pada breadboard.
2. Sensor DHT-22 diletakkan pada posisi vertikal agar tidak terjadi short circuit. Sensor ini digunakan untuk me-retrieve temperature dan humidity secara realtime.
3. Kabel jumper dipasang pada pin 3v3, Ground, dan pin 26 pada ESP32. Kemudian, kabel pin 3v3 disambungkan pada pin positive, kabel Ground disambungkan pada pin negative, dan kabel pin 26 pada pin data dari sensor DHT-22.
4. Sambungkan ESP32 pada laptop dengan menggunakan kabel micro-usb dan siapkan Arduino IDE.



IP Address Laptop dari Access Point

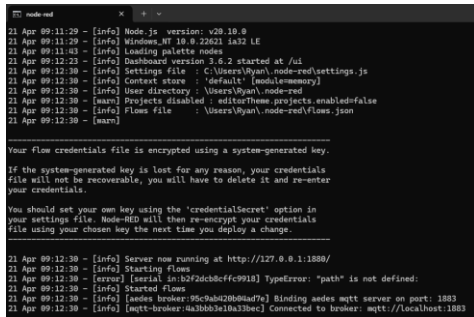
1. Hubungkan laptop anda dengan koneksi wifi yang biasa digunakan
2. Untuk mendapatkan alamat ip, gunakan command prompt dengan memasukkan command "netsh wlan show profile".
3. Kemudian lihat pada bagian IPv4 Address.
4. IP Address juga bisa didapatkan dengan menekan lambang "i" pada wifi.



DNS server assignment:	Automatic (DHCP)
SSID:	Yefta
Protocol:	Wi-Fi 4 (802.11n)
Security type:	WPA3-Personal
Manufacturer:	Intel Corporation
Description:	Intel(R) Wi-Fi 6 AX201 160MHz
Driver version:	22.160.0.4
Network band:	2.4 GHz
Network channel:	6
Link speed (Receive/Transmit):	144/144 (Mbps)
Link-local IPv6 address:	fe80::a7e3:53ad:8c74:1f88%5
IPv4 address:	172.20.10.2
IPv4 DNS servers:	172.20.10.1 (Unencrypted)
Physical address (MAC):	70-A6-CC-78-72-CE
View Wi-Fi security key	

Node Aedes MQTT Broker

Aktifkan node-red pada command prompt dan buka node-red menggunakan chrome atau search engine lainnya.



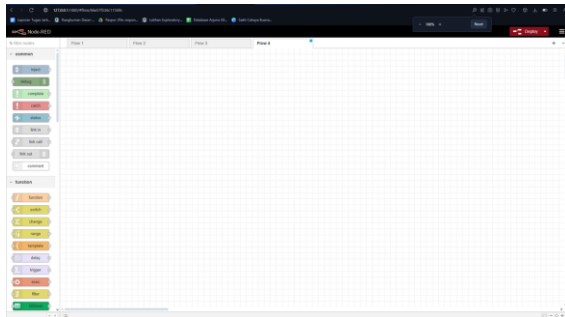
```
node-red
21 Apr 09:11:29 - [info] Node.js version: v20.10.0
21 Apr 09:11:29 - [info] Windows_NT 10.0.22621 ia32 LE
21 Apr 09:11:43 - [info] Loading palette nodes
21 Apr 09:12:23 - [info] Dashboard version 3.6.2 started at /ui
21 Apr 09:12:30 - [info] Settings file : C:\Users\Ryan\.node-red\settings.js
21 Apr 09:12:30 - [info] Context store : 'default' [module:memory]
21 Apr 09:12:30 - [info] User directory : \Users\Ryan\.node-red
21 Apr 09:12:30 - [warn] Projects disabled : editorTheme.projects.enabled=false
21 Apr 09:12:30 - [info] Flows file : \Users\Ryan\.node-red\flows.json
21 Apr 09:12:30 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

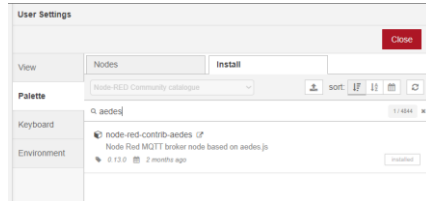
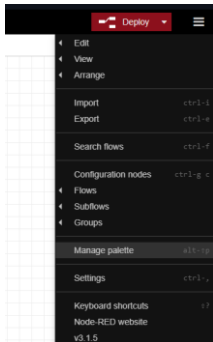
You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----

21 Apr 09:12:30 - [info] Server now running at http://127.0.0.1:1880/
21 Apr 09:12:30 - [info] Starting flows
21 Apr 09:12:30 - [error] [serial.in:b2f2dcb8cffe9918] TypeError: "path" is not defined
21 Apr 09:12:30 - [info] Started flows
21 Apr 09:12:30 - [info] [aedes broker:95c9ab420b04ad7e] Binding aedes mqtt server on port: 1883
21 Apr 09:12:30 - [info] [mqtt-broker:4a3bbb3e10a33bec] Connected to broker: mqtt://localhost:1883
```



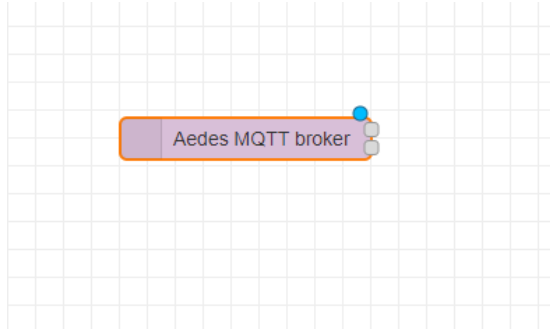
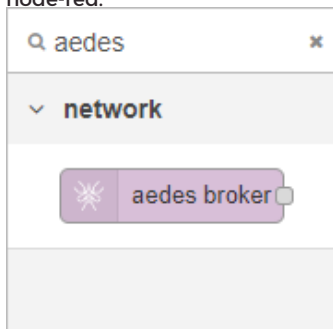
Node Aedes MQTT Broker

Install Node Aedes MQTT Broker pada
manage palletes



Node Aedes MQTT Broker

Setelah proses instalasi selesai, pilih Aedes MQTT Broker pada filter nodes dan drag ke dalam canvas node-red.



Koneksi Antar DHT-22 dengan ESP32

1. Install library DHT pada Library Manager Arduino IDE.
2. Import library DHT dengan command “#include <DHT.h>” (tanpa tanda petik).
3. Tulis line code di bawah ini untuk mengaktifkan dan mendapatkan bacaan temperature serta humidity dari DHT-22

```
11  DHT sensor_dht(DHT_PIN,DHT22);  
  
83  float temp = sensor_dht.readTemperature();  
84  float hum = sensor_dht.readHumidity();  
85  
86  doc["Temperature"] = temp;  
87  doc["Humidity"] = hum;  
88  
89  char jsonString[100];  
90  serializeJson(doc, jsonString);  
91  client.publish("/yefra_sensor/data", jsonString);  
92  
93  Serial.println("Data Publisheed: ");  
94  Serial.println(jsonString);  
95  
96  delay(2000);
```

Hasil Serial
Monitor

sketch_mar22a.ino

```
1  #include <WiFi.h>  
2  #include <PubSubClient.h>  
3  #include <DHT.h>  
4  #include <ArduinoJson.h>
```

Output Serial Monitor x

Message (Enter to send message to 'ESP32 Dev Module' on 'COM3')

```
Data Publisheed:  
{ "Temperature":30.60000038,"Humidity":86}  
Data Publisheed:  
{ "Temperature":30.60000038,"Humidity":85.90000153}  
Data Publisheed:  
{ "Temperature":30.60000038,"Humidity":85.90000153}  
Data Publisheed:  
{ "Temperature":30.70000076,"Humidity":85.90000153}  
Data Publisheed:  
{ "Temperature":30.70000076,"Humidity":86.09999847}
```

Koneksi Antar WiFi dengan ESP32

1. Install library WiFi pada Library Manager Arduino IDE.
2. Import library WiFi dengan command “#include <WiFi.h>” (tanpa tanda petik).
3. Tulis line code seperti pada slide berikutnya ini untuk mengaktifkan fitur WiFi pada ESP32.



sketch_mar22a.ino

```
1  #include <WiFi.h>
2  #include <PubSubClient.h>
3  #include <DHT.h>
4  #include <ArduinoJson.h>
```


Koneksi Antar WiFi dengan ESP32

```
6 const int DHT_PIN = 26;
7 const char* ssid = "Yefta"; // wifi ssid
8 const char* password = "12345678";
9 const char* mqtt_server = "172.20.10.2"; // mosquitto server url
10
11 DHT_sensor_dht(DHT_PIN,DHT22);
12 WiFiClient espClient;
13 PubSubClient client(espClient);
14 unsigned long lastMsg = 0;
15 float temp = 0;
16 float hum = 0;
17
18 StaticJsonDocument<100> doc;
```

```
20 void setup_wifi() {
21   delay(10);
22   Serial.println();
23   Serial.print("Wifi terkoneksi ke : ");
24   Serial.println(ssid);
25
26   WiFi.mode(WIFI_STA);
27   WiFi.begin(ssid, password);
28
29   while (WiFi.status() != WL_CONNECTED) {
30     delay(500);
31     Serial.print(".");
32   }
33
34   randomSeed(micros());
35
36   Serial.println("");
37   Serial.println("WiFi berhasil terkoneksi");
38   Serial.print("Alamat IP : ");
39   Serial.println(WiFi.localIP());
40 }
41
42 void callback(char* topic, byte* payload, unsigned int length) {
43   Serial.print("Message arrived [");
44   Serial.print(topic);
45   Serial.print("] ");
46   for (int i = 0; i < length; i++) {
47     Serial.print((char)payload[i]);
48   }
49 }
50
51 void reconnect() {
52   while (!client.connected()) {
53     Serial.print("Baru melakukan koneksi MQTT ...");
54     String clientId = "ESP32Client-";
55     clientId += String(random(0xffff), HEX);
56     if (!client.connect(clientId.c_str())) {
57       Serial.println("Connected");
58     } else {
59       Serial.print("failed, rc=");
60       Serial.print(client.state());
61       Serial.println(" try again in 5 seconds");
62       delay(5000);
63     }
64   }
65 }
```

Output Serial Monitor ×

Message (Enter to send message to 'ESP32 Dev Module' on 'COM3')

.....

WiFi berhasil terkoneksi
Alamat IP : 172.20.10.8
Baru melakukan koneksi MQTT ...Connected

Koneksi Antar MQTT Server dengan ESP32

1. Install library PubSubClient pada Library Manager Arduino IDE.
2. Import library WiFidengan command “#include <PubSubClient.h>” (tanpa tanda petik).
3. Tulis line code di bawah ini untuk mengaktifkan fitur PubSubClient pada ESP32. Command lengkap tersedia pada potongan code berikut.

```
75 void loop() {
76   if (!client.connected()) {
77     reconnect();
78   }
79   client.loop();
80
81   unsigned long now = millis();
82   if (now - lastMsg > 3000) { //perintah publish data
83     lastMsg = now;
84
85     float temp = sensor_dht.readTemperature();
86     float hum = sensor_dht.readHumidity();
87
88     doc["temperature"] = temp;
89     doc["humidity"] = hum;
90
91     char jsonString[100];
92     serializeJson(doc, jsonString);
93     client.publish("/mqtt_sensor/data", jsonString);
94
95     Serial.println("Data Published: ");
96     Serial.println(jsonString);
97     delay(2000);
98   }
99 }
```

sketch_mar22a.ino

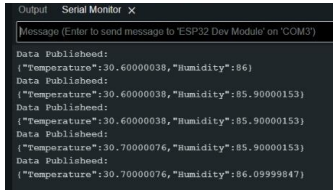
```
1  #include <WiFi.h>
2  #include <PubSubClient.h>
3  #include <DHT.h>
4  #include <ArduinoJson.h>
```

```
13 PubSubClient client(espClient);
```

```
50 void reconnect() {
51   while (!client.connected()) {
52     Serial.print("Baru melakukan koneksi MQTT ...");
53     String clientId = "ESP32client-";
54     clientId += String(random(0xffff), HEX);
55     if (!client.connect(clientId.c_str())) {
56       Serial.println("Connected");
57     } else {
58       Serial.print("failed, rc=");
59       Serial.print(client.state());
60       Serial.println(" try again in 5 seconds");
61       delay(5000);
62     }
63   }
64 }
65 void setup() {
66   //pinMode(2, OUTPUT);
67   Serial.begin(115200);
68   sensor_dht.begin();
69   setup_wifi();
70   client.setServer(mqtt_server, 1883);
71   client.setCallback(callback);
72 }
```

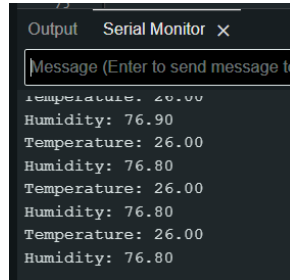
Koneksi Antar MQTT Server dengan ESP32

1. Berikut data dalam bentuk topik dan message secara langsung
2. Berikut data dalam bentuk JSON

A screenshot of a serial monitor window titled "Output Serial Monitor X". It shows a text input field with the placeholder "Message (Enter to send message to 'ESP32 Dev Module' on 'COM3')". Below the input field, there is a list of JSON messages. Each message is preceded by the text "Data Publisheed:". The messages are: {"Temperature":30.60000038,"Humidity":86}, {"Temperature":30.60000038,"Humidity":85.90000153}, {"Temperature":30.60000038,"Humidity":85.90000153}, {"Temperature":30.70000076,"Humidity":85.90000153}, {"Temperature":30.70000076,"Humidity":86.09999847}.

```
Output Serial Monitor X
Message (Enter to send message to 'ESP32 Dev Module' on 'COM3')

Data Publisheed:
{"Temperature":30.60000038,"Humidity":86}
Data Publisheed:
{"Temperature":30.60000038,"Humidity":85.90000153}
Data Publisheed:
{"Temperature":30.60000038,"Humidity":85.90000153}
Data Publisheed:
{"Temperature":30.70000076,"Humidity":85.90000153}
Data Publisheed:
{"Temperature":30.70000076,"Humidity":86.09999847}
```

A screenshot of a serial monitor window titled "Output Serial Monitor X". It shows a text input field with the placeholder "Message (Enter to send message to)". Below the input field, there is a list of plain text messages. Each message is preceded by the text "temperature. 26.00". The messages are: Humidity: 76.90, Temperature: 26.00, Humidity: 76.80, Temperature: 26.00, Humidity: 76.80, Temperature: 26.00, Humidity: 76.80.

```
Output Serial Monitor X
Message (Enter to send message to)

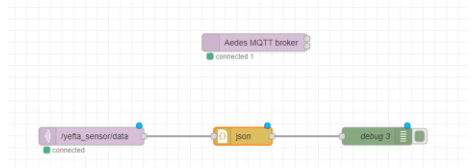
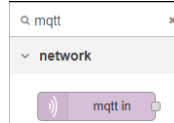
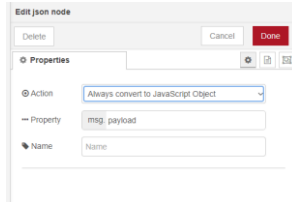
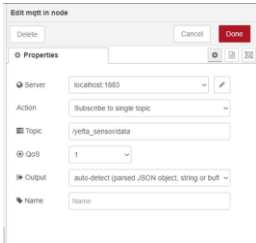
temperature. 26.00
Humidity: 76.90
Temperature: 26.00
Humidity: 76.80
Temperature: 26.00
Humidity: 76.80
Temperature: 26.00
Humidity: 76.80
```

Koneksi Antar MQTT Server dengan ESP32

Kode lengkap untuk menghubungkan ESP32
denga Wi-Fi dan output temperature dan humidity
dalam bentuk JSON dapat dilihat pada link berikut:
<https://bit.ly/3wcrSdh>

Debug Node-Red dengan MQTT

1. Buatlah rangkaian node pada node-red seperti gambar di samping dengan node aedes, json, dan mqtt in.
2. Jangan lupa atur konfigurasi node seperti di bawah.



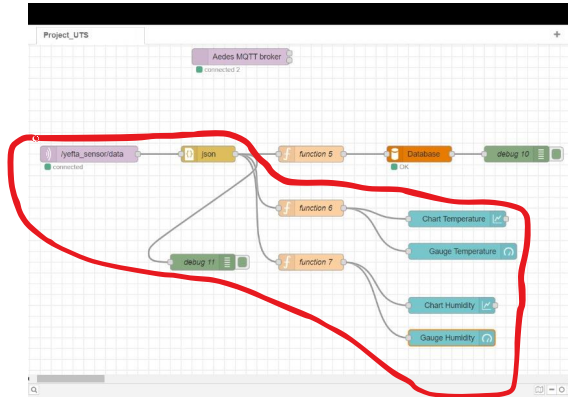
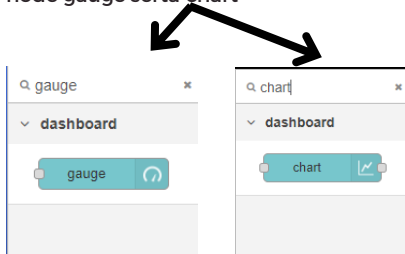
Debug Node-Red dengan MQTT

Klik “Deploy” dan apabila keluaran sudah seperti gambar di bawah, maka proses retrieve data dari ESP32 ke MQTT Node-Red sudah berhasil.



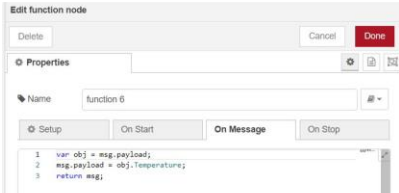
Implementasi Dashboard Gauge dan Chart

Untuk membuat Gauge dan Chart pada Dashboard Node-red, buatlah rangkaian seperti gambar di samping (bagian yang dilingkari warna merah) dengan tambahan node gauge serta chart

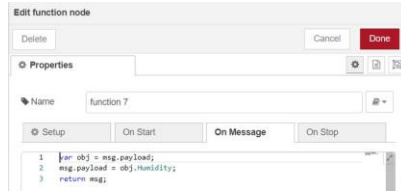


Implementasi Dashboard Gauge dan Chart

Masukkan kode di bawah pada Function 6



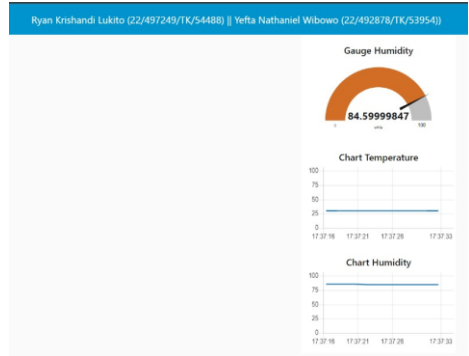
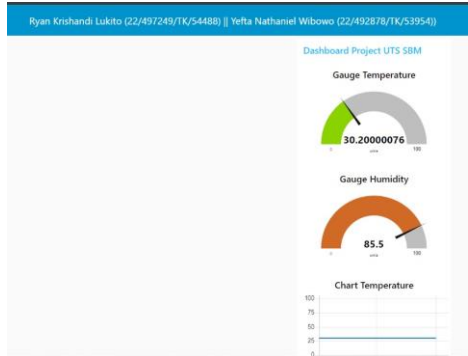
Masukkan kode di bawah pada Function 7



Isi dari node gauge dan chart tidak perlu diubah
(kecuali ingin mengubah posisi gauge dan chart).

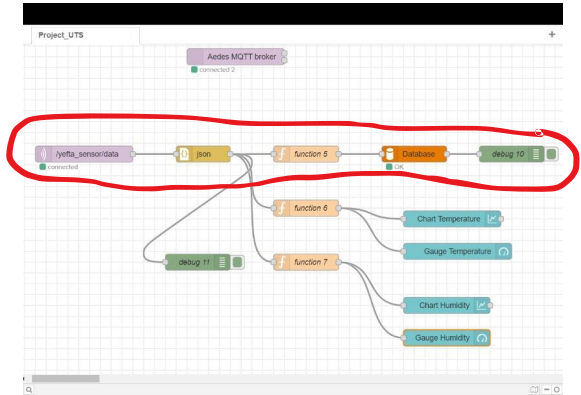
Implementasi Dashboard Gauge dan Chart

Berikut hasil akhir dari implementasi Gauge dan Chart pada Dashboard Node-red



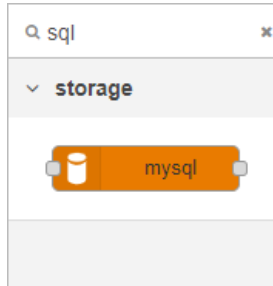
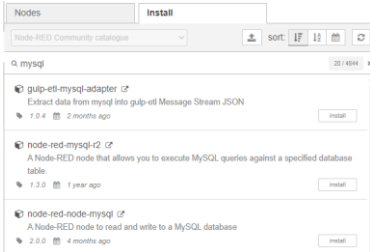
Implementasi MySQL pada Node-red

Untuk menambahkan data ke dalam database MySQL melalui node-red, buatlah rangkaian seperti gambar di samping (bagian yang dilingkari warna merah)



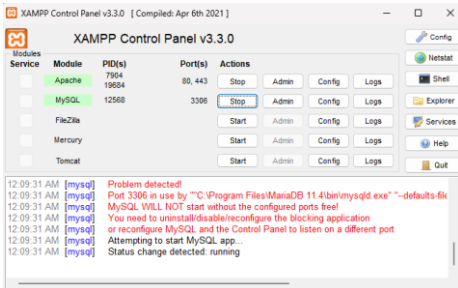
Node MySQL untuk Database

Install node database untuk memasukkan data dari ESP32 ke dalam database. Dalam penugasan ini, database yang digunakan adalah MySQL berbasis xampp.



Implementasi MySQL pada Node-red

Install XAMPP dan jalankan Apache serta MySQL. Setelahnya, tulis "localhost/phpmyadmin" pada search engine yang biasa anda gunakan.



Implementasi MySQL pada Node-red

Di dalam “localhost/phpmyadmin”, buat sebuah database bernama “sbm_db” dan tabel bernama “uts_table”.

Create database

sbm_db utf8mb4_general_ci Create

Create new table

Table name Number of columns

uts_table 4 Create

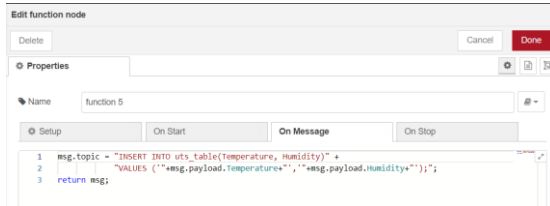
Implementasi MySQL pada Node-red

Isikan “uts_table” dengan beberapa atribut seperti id, Temperature, Humidity, dan Time.

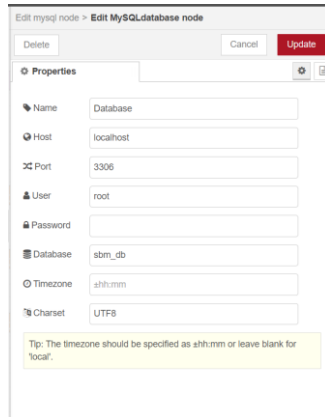
Name	Type	Length/Values	Default	Collation	Attributes	Null Index	Comments
<input type="text" value="id"/>	INT		None			<input type="checkbox"/> PRIMARY	<input checked="" type="checkbox"/>
<small>Pick from Central Columns</small>							
<input type="text" value="Temperature"/>	INT		None			<input type="checkbox"/> --	<input type="checkbox"/>
<small>Pick from Central Columns</small>							
<input type="text" value="Humidity"/>	INT		None			<input type="checkbox"/> --	<input type="checkbox"/>
<small>Pick from Central Columns</small>							
<input type="text" value="Time"/>	DATE		CURRENT_TIME			<input type="checkbox"/> --	<input type="checkbox"/>
<small>Pick from Central Columns</small>							

Implementasi MySQL pada Node-red

Masukkan kode di bawah pada node
Function 5

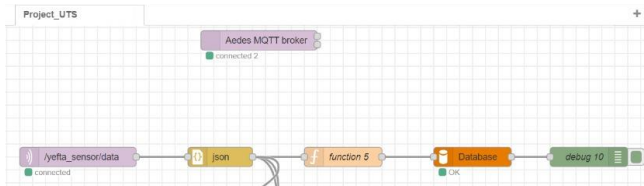


Sesuaikan konfigurasi pada node database



Implementasi MySQL pada Node-red

Berikut rangkaian node untuk
menghubungkan MySQL dengan ESP32
pada Node-red



Implementasi MySQL pada Node-red

Klik “Deploy” apabila seluruh konfigurasi nodes sudah sesuai. Kemudian, lihat pesan pada debug. Apabila keluarannya seperti pada gambar disamping, maka data dari DHT-22 sudah berhasil dimasukkan ke dalam MySQL

```
4/1/2024, 5:31:52 PM node: debug 10  
INSERT INTO uts_table(Temperature,  
Humidity)VALUES ('30.5','86.30000305'); : msg.payload  
: ResultSetHeader  
"[object Object]"
```

