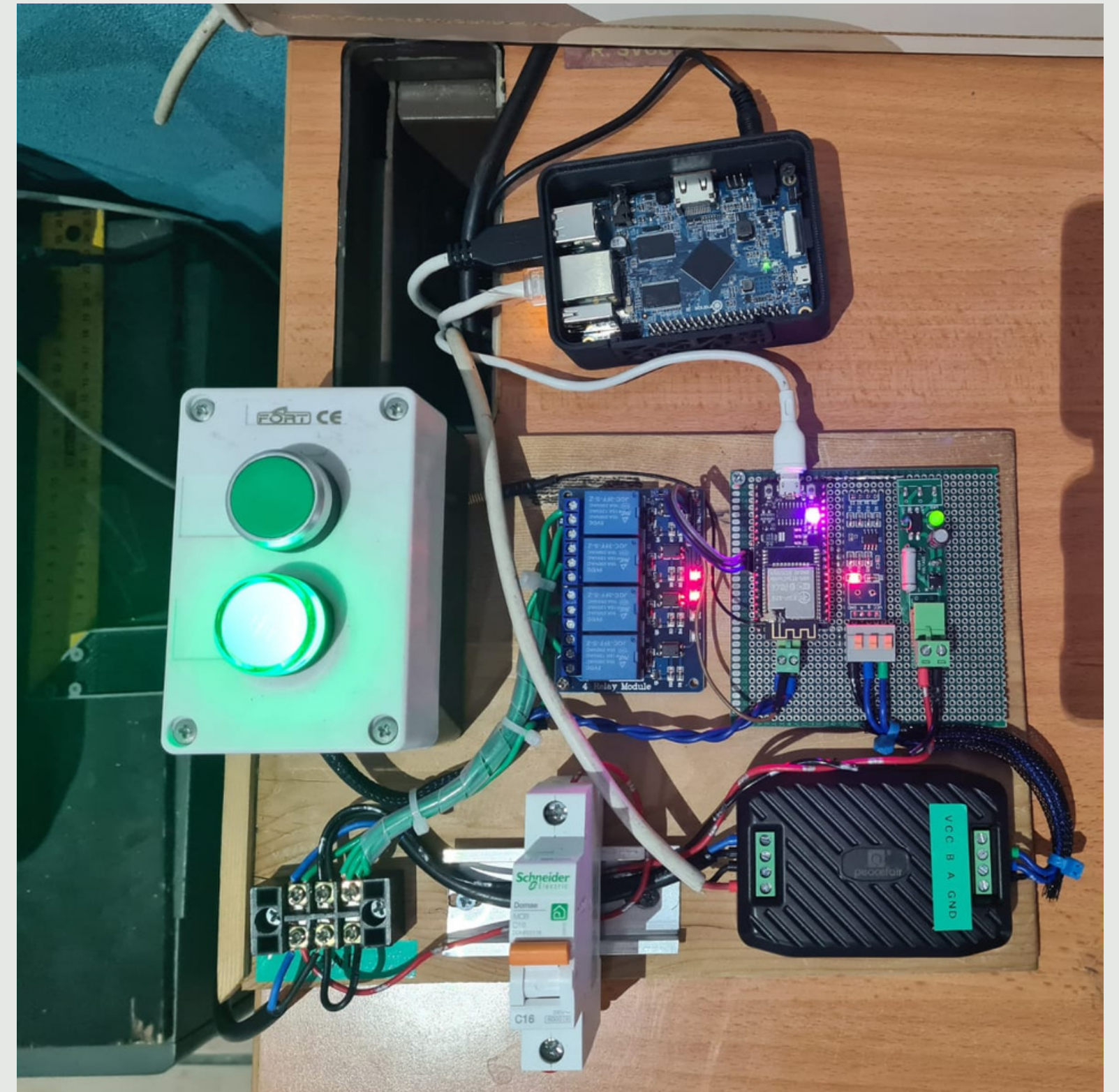


SYSTEM MONITORING ELECTRICITY

Ath Thaareq Mahesa



Daftar Isi

1



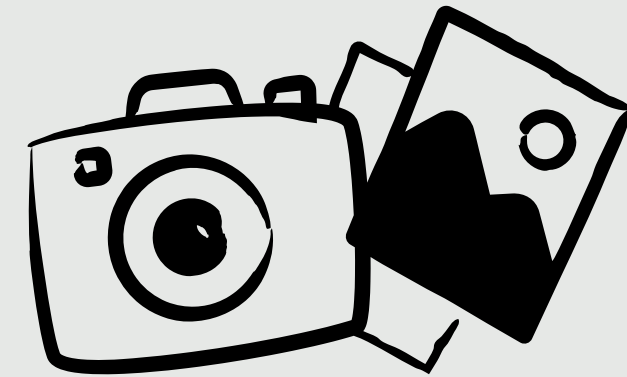
Architecture
Wiring & System

2

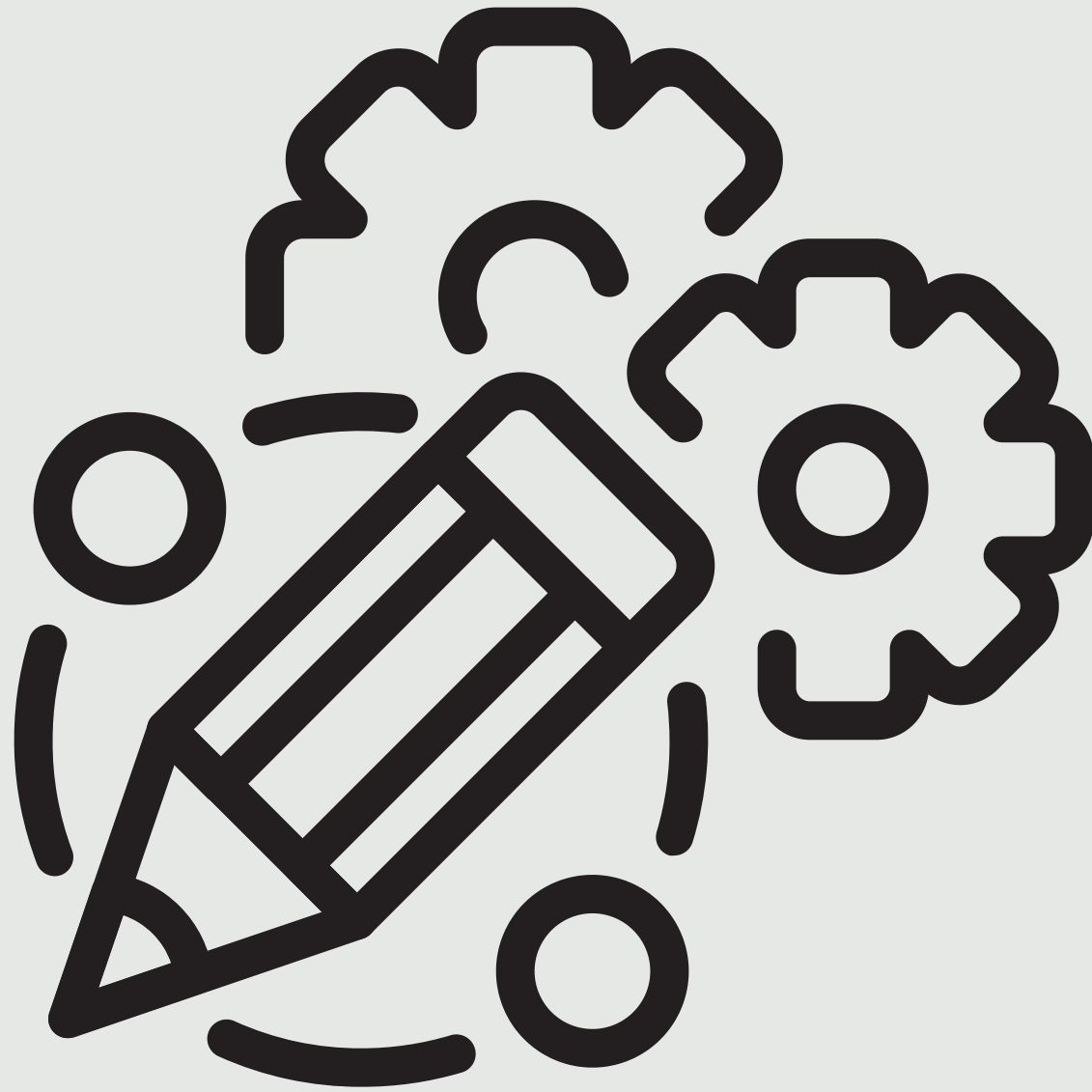


Fitur Monitoring

3

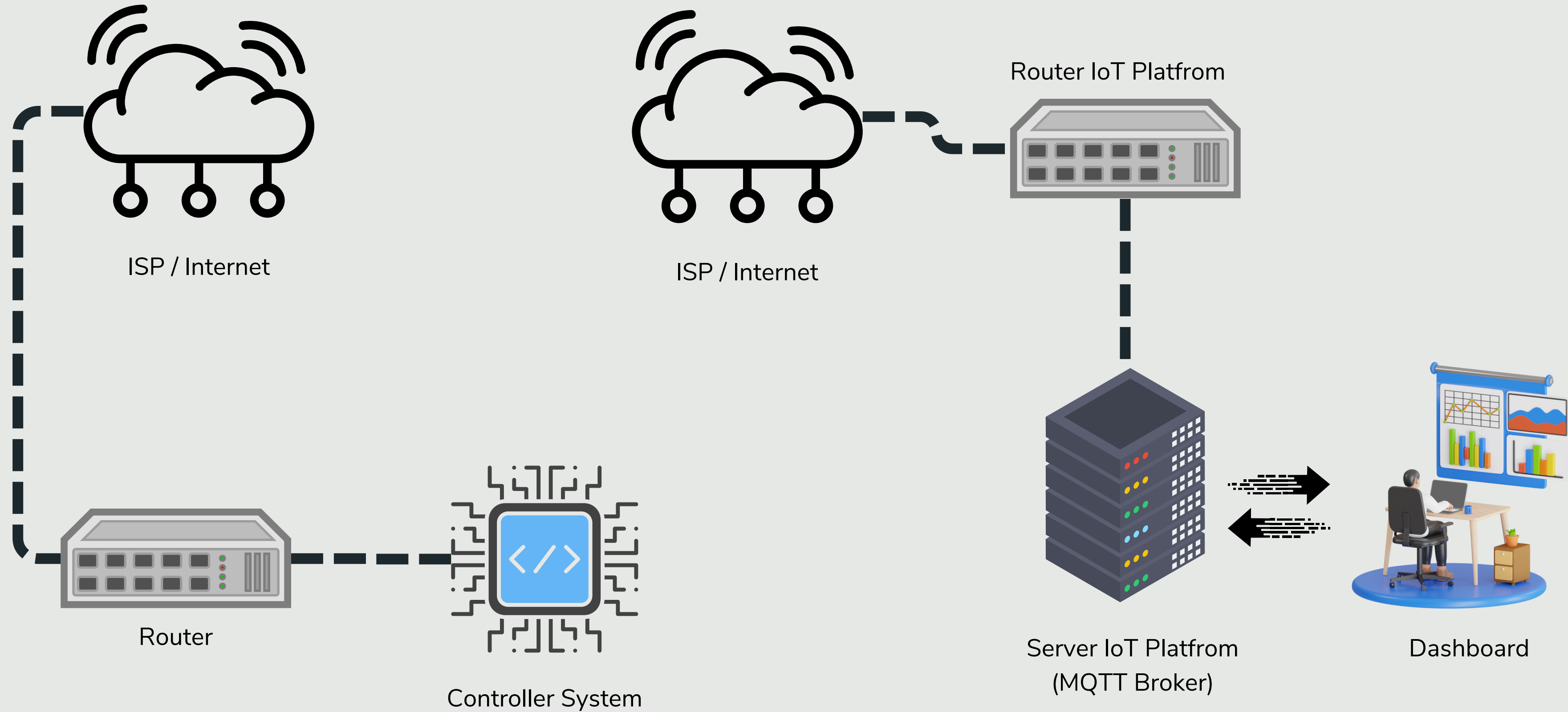


Dokumentasi

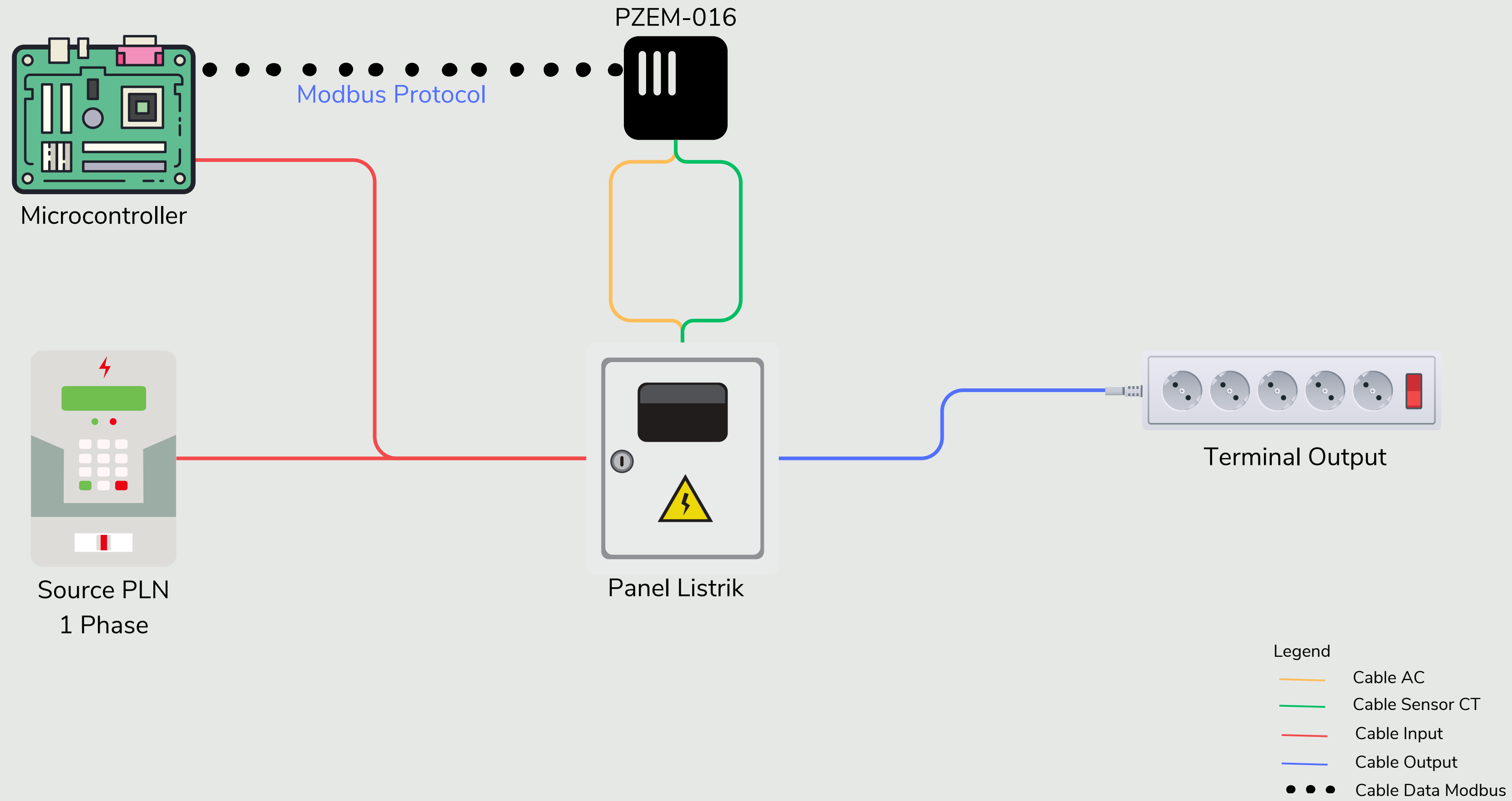


Architecture
Wiring & System

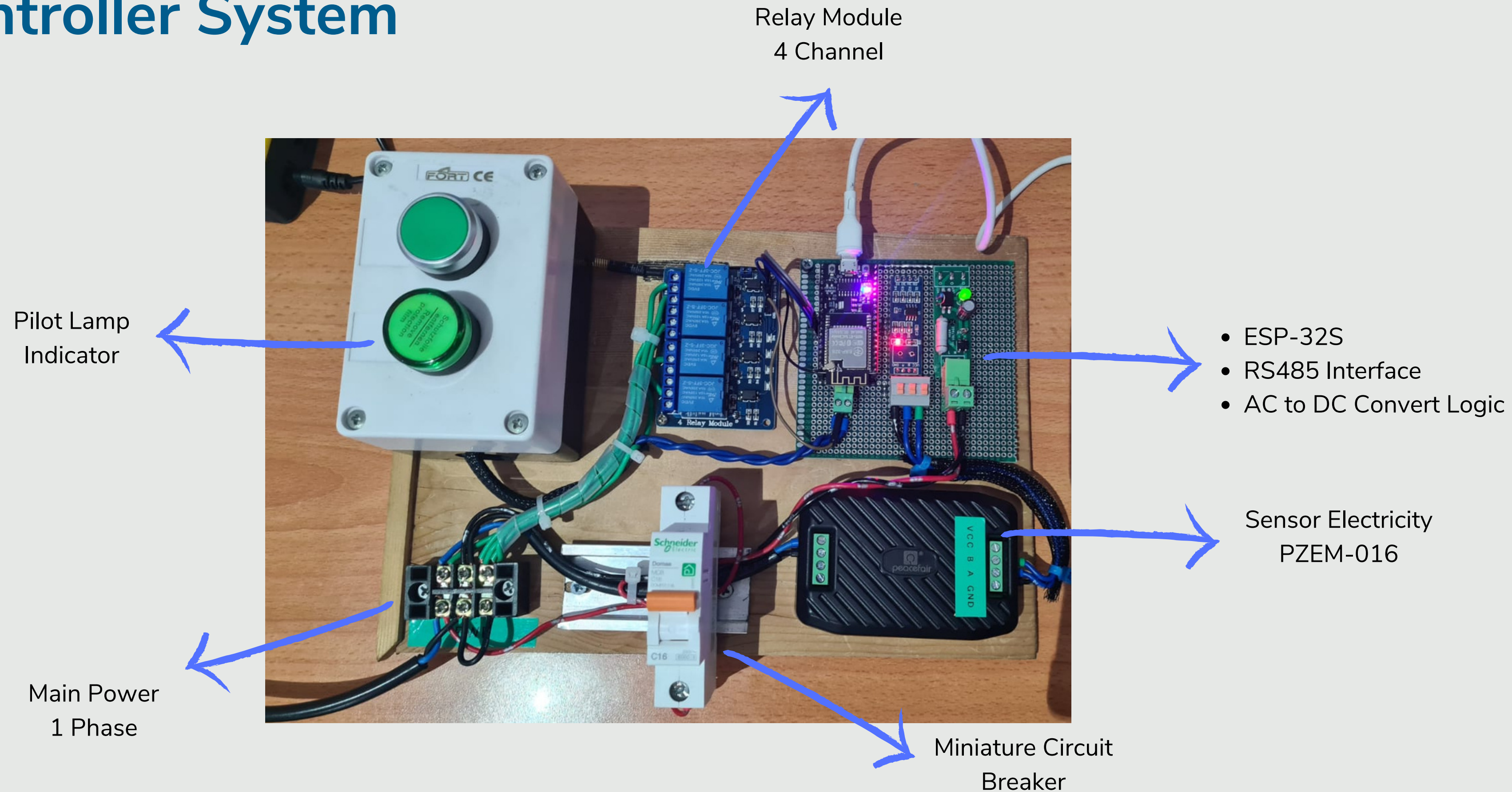
Architecture System



Architecture Wiring



Controller System



Function Of Each Component

1

Microcontroller ESP32S

Microcontroller ESP32 difungsikan sebagai pemroses input dari sensor ataupun output dari aktuator, sebagai menerima perintah dari server dan mengirimkan data yang didapat dari sensor ke server

2

RS485 Module

RS485 module difungsikan sebagai Transmisi Data Serial, untuk melakukan akuisisi pembacaan data sensor dari PZEM-016, dimana sensor menggunakan protocol modbus, dimana dapat menerima data serial jarak jauh (hingga 1200 meter)

3

AC To DC Converter Logic

AC To DC Converter Logic difungsikan membaca last state dari MCB, apakah terjadi trip atau aktif, sensor ini berupa logic TTL (HIGH / LOW). Dimana dari hasil pengujian nanti parameternya adalah :

- LOW / 0 = Sensor mendeteksi adanya tegangan 220v AC
- HIGH / 1 = Tidak adanya tegangan 220v AC yang mengalir

4

Relay Module 4 Channel

Relay Module 4 Channel difungsikan sebagai driver untuk mengontrol suatu aktuator, seperti motor, solenoid, ataupun lampu, dimana cara kerja nya memutuskan salah satu polaritas kabel yang terhubung ke kaki relay



Fitur Monitoring

Fitur Monitoring

1

Monitoring Kelistrikan

Menampilkan status listrik PLN (active or deactivate), tegangan (V), arus (A), daya (W), frekuensi (Hz), pemakaian energi (KwH) dan estimasi biaya listrik (Rp)

2

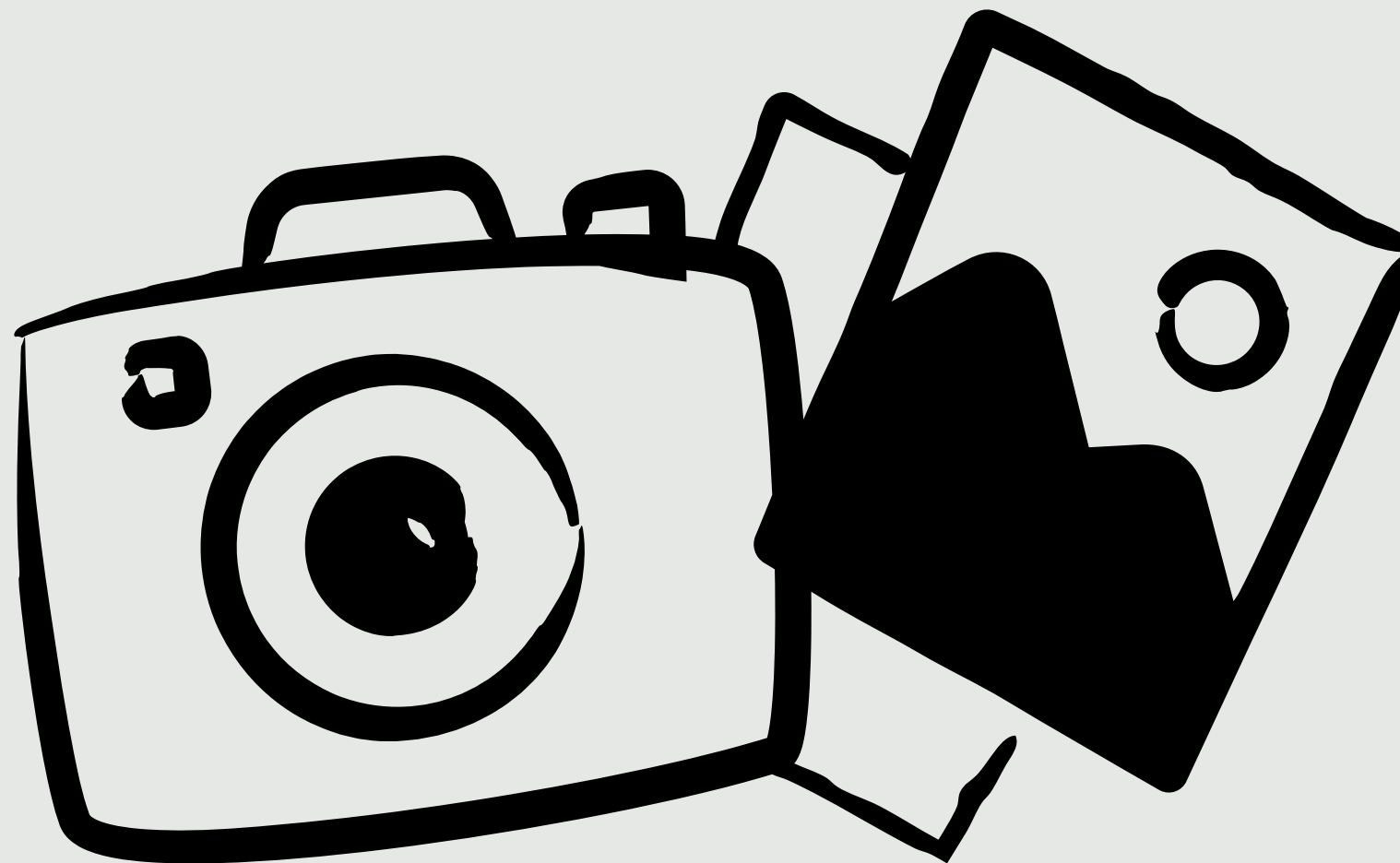
Monitoring MCB Status

Menampilkan status keadaan posisi MCB apakah aktif atau trip

3

Control Lampu

Mengontrol lampu baik ON / OFF, dan melihat status lampu tersebut menyala atau mati



Dokumentasi

1

Hasil Pengujian PUB/SUB MQTT Method

The screenshot displays the MQTTBox application window. The top bar includes a menu, a back arrow, a 'Connected' status indicator, and buttons for 'Add publisher' and 'Add subscriber'. The main interface is divided into three panels. The left panel, titled 'Publish', contains fields for 'Topic to publish' (set to 'relay/edspert/control/cmdnd'), 'QoS' (set to '0 - Almost Once'), 'Retain' (unchecked), 'Payload Type' (set to 'Strings / JSON / XML / Characters'), and a 'Payload' field (set to '0'). A 'Publish' button is located below these fields. The right panel, titled 'Subscribe', shows two active subscriptions. The first subscription is for the topic 'pow/edspert/data/stat', displaying a JSON payload with various sensor readings and a 'qos : 0, retain : false, cmd : publish, dup : false, topic : pow/edspert/data/stat, mes sageld : , length : 143, Raw payload : 1233480798769823458565146495757575544348679768434585051514649575757554434657780345848465255554434708269813458534844348070345848465553443469786982718934584846495044347767663458347978344434766577803458347970703444348283837334583445534834125' message. The second subscription is for the topic 'pow/edspert/status/cmdnd', displaying a 'qos : 0, retain : true, cmd : publish, dup : false, topic : pow/edspert/status/cmdnd, mes sageld : , length : 31, Raw payload : 79110108105110101' message.

Publish

Subscribe

Pengujian menggunakan tool MQTTBOX, untuk mengetahui apakah broker MQTT sudah aktif, lalu di ikuti dengan pengujian publish data untuk menyalakan relay dan juga subscribe data untuk memonitoring electricity

2

Hasil Pengujian PUB/SUB MQTT Method



✕ pow/edspert/data/stat

`{"POWER":83.199997,"VOLT":233.199997,"AMP":0.477,"FREQ":50,"PF":0.75,"ENERGY":0.12,"MCB":"ON","LAMP":"OFF","RSSI":"-50"}`

qos : 0, **retain** : false, **cmd** : publish, **dup** : false, **topic** : pow/edspert/data/stat, **msgid** : , **length** : 143, **Raw payload** : 1233480798769823458565146495757575544348679768434585051514649575757554434657780345848465255554434708269813458534844348070345848465553443469786982718934584846495044347767663458347978344434766577803458347970703444348283837334583445534834125

`{"POWER":78.199997,"VOLT":233.300003,"AMP":0.437,"FREQ":50,"PF":0.77,"ENERGY":0.12,"MCB":"ON","LAMP":"OFF","RSSI":"-51"}`

qos : 0, **retain** : false, **cmd** : publish, **dup** : false, **topic** : pow/edspert/data/stat, **msgid** : , **length** : 143, **Raw payload** : 12334807987698234585556464957575755443486797684345850515146514848484851443465778034584846525155443470826981345853484434807034584846555443469786982718934584846495044347767663458347978344434766577803458347970703444348283837334583445534934125

`{"POWER":79.900002,"VOLT":233.300003,"AMP":0.449,"FREQ":50,"PF":0.76,"ENERGY":0.12,"MCB":"ON","LAMP":"OFF","RSSI":"-52"}`

Publish menggunakan format JSON, bertujuan melakukan efisien dalam melakukan pengiriman data, tidak memerlukan topic yang banyak sesuai dengan variable pada gambar di samping serta efisiensi bandwidth ukuran pesan yang dikirimkan melalui MQTT

3

Pembuatan Source Code Controller Menggunakan Platformio

```
monitoring_electricity > src > main.cpp > MCB_read()
158 void powerRead()
159 {
195     JSONEncoder["RSSI"] = rssi;
196
197     char JSONmessageBuffer_electric[300];
198     JSONEncoder.printTo(JSONmessageBuffer_electric, sizeof(JSONmessageBuffer_electric));
199     client.publish(MQTT_ELECTRICITY_STATE, JSONmessageBuffer_electric);
200
201     // Serial.println("Error sending message");
202
203     Serial.println("Sending message to MQTT topic..");
204     Serial.println(JSONmessageBuffer_electric);
205 }
206
207 void callback(char *topic, byte *message, unsigned int length)
208 {
209     Serial.print("Message arrived in topic: ");
210     Serial.println(topic);
211     Serial.print("Message:");
212     String messageTemp;
213
214     for (int i = 0; i < length; i++)
215     {
216         Serial.print((char)message[i]);
217         Serial.println();
218         if ((char)message[i] != '')
219             messageTemp += (char)message[i];
220     }
221
222     if (String(topic) == MQTT_RELAY_CMD)
223     {
224         if (messageTemp == "1")
225         {
226             digitalWrite(lampu, HIGH);
227         }
228     }
229 }
```

4

Source Code

[https://github.com/eunbiline98/system-monitoring-electricity-
pzem-016-](https://github.com/eunbiline98/system-monitoring-electricity-pzem-016-)