

Membuat program socket server

Tujuan

- Mampu memahami protokol komunikasi socket client TCP/IP
- Mampu membuat program komunikasi antara MCU sebagai server socket dan komputer sebagai client socket dengan C#
- Mampu membuat program untuk mendengarkan client (listening) dan menerima data sensor dari MCU ke komputer secara real-time dengan protokol komunikasi socket TCP/IP

Capaian

- Menjelaskan cara kerja socket server yang bertugas sebagai “listening” dari semua socket client yang terkoneksi
- Menjelaskan konsep socket Asynchronous dengan C#
- Membuat program Socket Server dengan GUI C#, Java, Python, dan lain-lain untuk menerima data sensor MCU, kemudian menampilkannya secara real-time di sisi socket server

Teori Singkat

Untuk program socket server sudah dibuat pada pertemuan [sebelumnya](#), tetapi pada kode socket tersebut masing sederhana yaitu hanya menerima data dari socket client yang dikirimkan.

Untuk kode socket server akan dibuat agar fungsi-fungsi didalamnya lebih banyak lagi, agar tidak hanya bisa menerima data dari client.

Praktikum

Masih menggunakan kode socket server yang sebelumnya, ubahlah kode pada fungsi `run` menjadi seperti di bawah ini.

```
def run(self):
    while True:
        try:
            MESSAGE = input("Input response:")
            conn.send(MESSAGE.encode("utf8")) # echo
        except Exception as e:
            print(e)
            break
        sleep(0.25)
```

Dengan mengubah program tersebut, socket server yang akan kita buat mampu menerima input dari keyboard sehingga dapat dimanfaatkan untuk memasukan perintah pada controller atau ESP8266 yang kita miliki.

```
#include <Arduino.h>
#include <ESP8266WiFi.h>

const char *ssid = "####";
const char *password = "####";
const uint16_t port = 2004;
const char *host = "####";
```

```

WiFiClient client;

void connect_wifi()
{
  Serial.printf("Connecting to %s ", ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println(" connected");
  delay(250);
}

void connect_server()
{
  while (!client.connect(host, port))
  {
    Serial.printf("\n[Connecting to %s ... ", host);
    delay(1000);
    return;
  }
  Serial.println("connected");
  delay(1000);
}

void setup()
{
  Serial.begin(115200);
  Serial.println("Contoh penggunaan socket client");
  connect_wifi();
  connect_server();
}

void loop()
{
  if (client.connected())
  {
    Serial.print("[Response:");
    String line = client.readStringUntil('\n');
    Serial.println(line);
    if (line.equalsIgnoreCase("led-on"))
    {
      pinMode(LED_BUILTIN, HIGH);
      delay(3000);
      pinMode(LED_BUILTIN, LOW);
    }
  }
  delay(250);
}

```

Kode tersebut mirip dengan kode pada pertemuan sebelumnya, yang perlu dimodifikasi adalah bagian di bawah ini

```

if (line.equalsIgnoreCase("led-on"))
{
  pinMode(BUILTIN_LED, HIGH);
  delay(3000);
  pinMode(BUILTIN_LED, LOW);
}

```

Fungsi kode di atas digunakan untuk membaca setiap data dari socket server, ketika data tersebut `led-on` berarti akan menghidupkan LED bawaan esp8266.

Tugas

Terdapat sebuah dusun di desa tertentu yang sudah menerapkan IoT, contoh penerapan tersebut di gang-gang ketika sudah beranjak malam lampu yang terdapat pada gang tersebut akan menyala. Pada dusun tersebut juga terdapat kebun rumah kaca, dimana suhu dan kelembaban sangat diperhatikan untuk menjaga produktivitas sayur-sayur di dalam kebun. Semua sensor yang terdapat pada dusun tersebut juga dapat dimonitoring dan semua lampu yang terdapat pada gang-gang dapat dinyalakan melalui server.

Dari kasus di atas, buat program untuk kebutuhan tersebut baik dari sisi controller (ESP8266) atau dari sisi server.