

## Materi 5

### Komunikasi menggunakan WiFi Shield

#### A. Waktu :

Alokasi Waktu : 2 x 100 menit

- Praktikum Terbimbing: 1 x 100 menit
- Praktikum Mandiri: 1 x 100 menit

#### B. Tujuan Pembelajaran:

1. Mampu melakukan konfigurasi WiFi shield
2. Mampu melakukan komunikasi (mengirim/menerima data) menggunakan WiFi shield

#### C. Alat dan Bahan:

1. Arduino Uno
2. ESP-01 (Tipe ESP8266 yang paling kecil)
3. Kabel Jumper
4. Kabel RJ-45
5. Arduino IDE

#### D. Materi

Arduino WiFi Shield memungkinkan Arduino untuk terhubung ke internet secara nirkabel menggunakan spesifikasi 802.11 (WiFi). Ini didasarkan pada HDG204 Wireless

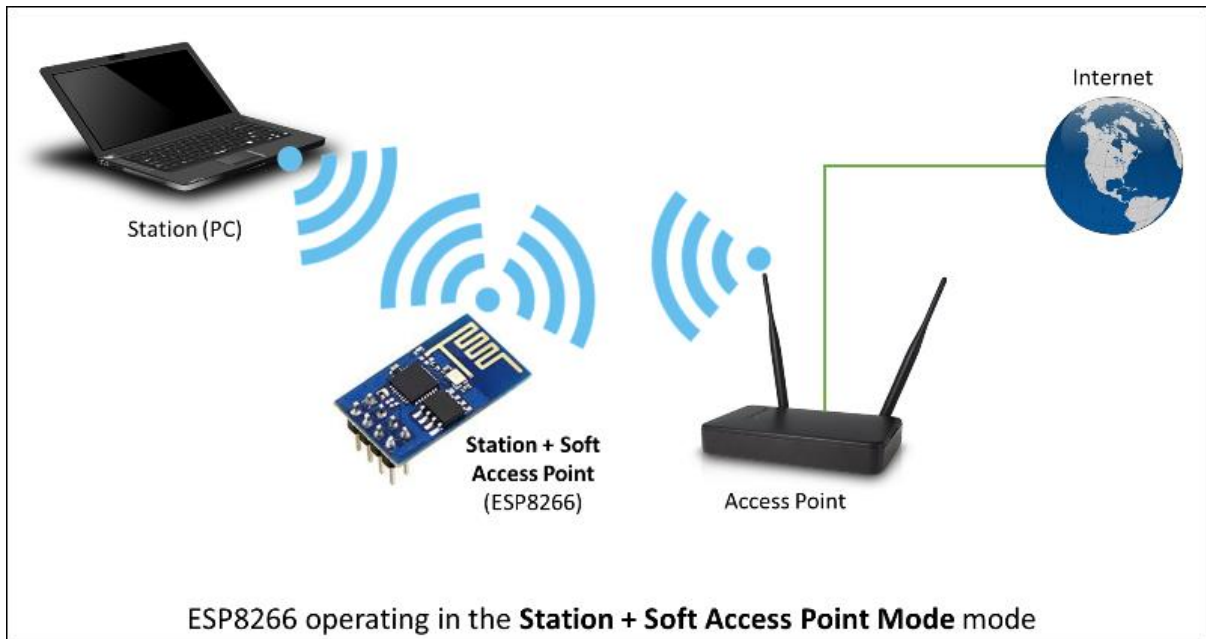
LAN 802.11b / g. AT32UC3 menyediakan jaringan (IP) stack yang dapat digunakan baik TCP maupun UDP. Gunakan library WiFi (ESP866Wifi.h) untuk menulis kode sumber agar terhubung ke internet menggunakan shield. Shield WiFi terhubung ke papan Arduino menggunakan header kawat-bungkus panjang yang membentang melalui shield, hal ini menjaga tata letak pin tetap utuh dan memungkinkan shield lain ditumpuk di atasnya.

Library Wi-Fi untuk ESP8266 telah dikembangkan berdasarkan ESP8266 SDK, menggunakan konvensi penamaan dan filosofi fungsionalitas keseluruhan dari library Arduino WiFi.

### **Bekerja dengan ESP8266**

Perangkat yang terhubung ke jaringan Wi-Fi disebut stasiun (STA). Koneksi ke WiFi disediakan oleh Access Point (AP), yang berfungsi sebagai hub untuk satu atau lebih stasiun. Ujung AP yang lain terhubung ke jaringan kabel. Jalur akses biasanya terintegrasi dengan router untuk menyediakan akses dari jaringan WiFi ke internet. Setiap jalur akses dikenali oleh SSID (Service Set Identifier), yang pada dasarnya adalah nama jaringan yang Anda pilih saat menghubungkan perangkat (stasiun) ke WiFi.

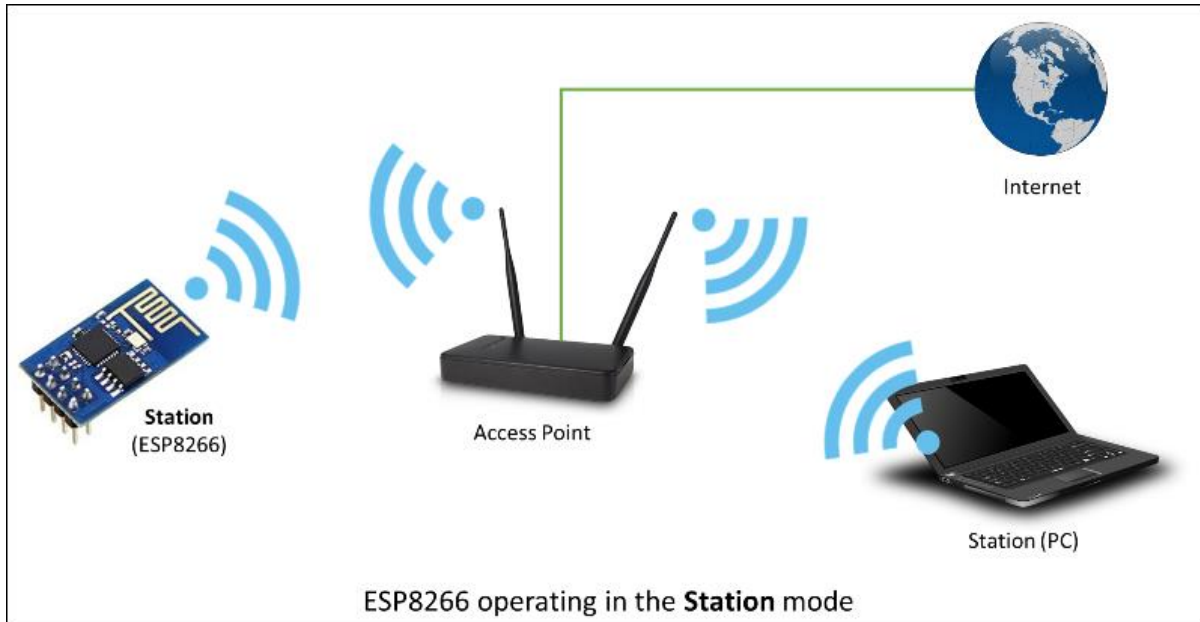
Modul ESP8266 dapat beroperasi sebagai stasiun, sehingga kita dapat menghubungkannya ke jaringan WiFi. Ini juga dapat beroperasi sebagai Access Point lunak (soft-AP), untuk membangun jaringan WiFi sendiri. ESP8266 juga dapat beroperasi baik di stasiun dan mode soft-AP.



## Stasiun

Mode Station (STA) digunakan untuk mendapatkan modul ESP yang terhubung ke jaringan Wi-Fi yang ditetapkan oleh AP. [Library class station](#) memiliki beberapa fitur untuk memudahkan manajemen koneksi Wi-Fi. Jika koneksi terputus, ESP8266 akan secara otomatis terhubung kembali ke AP terakhir yang digunakan, apabila tersedia kembali. Hal yang sama terjadi pada reboot modul. Ini dimungkinkan karena ESP menyimpan kredensial ke AP yang terakhir digunakan dalam memori flash (tidak mudah menguap). Menggunakan data yang disimpan ESP juga akan terhubung kembali jika sketsa telah diubah tetapi kode tidak mengubah mode Wi-Fi atau kredensial.

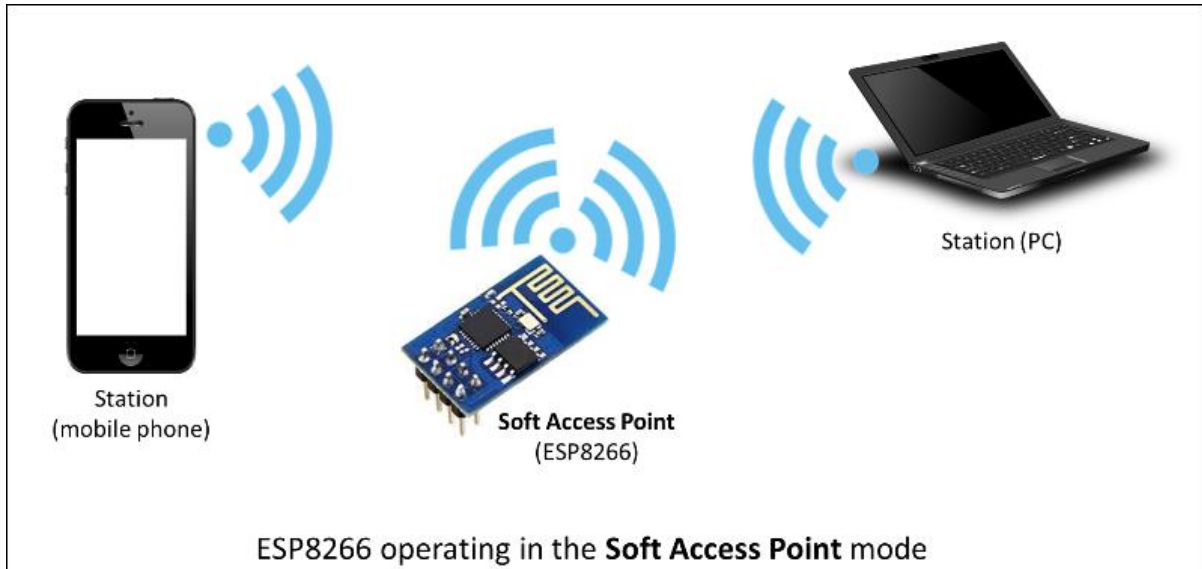
[Klik Library Class ESP8266 sebagai Station](#)



## Soft Access Point

Access Point adalah perangkat yang menyediakan akses ke jaringan WiFi ke perangkat lain (stasiun) dan menghubungkannya lebih jauh ke jaringan kabel. ESP8266 dapat menyediakan fungsi serupa kecuali ia tidak memiliki antarmuka ke jaringan kabel. Mode operasi semacam ini disebut titik akses lunak (soft-AP). Jumlah maksimum stasiun yang secara bersamaan dapat dihubungkan ke soft-AP dapat diatur dari 1 hingga 8, tetapi default ke 4. Mode soft-AP sering digunakan dan langkah lanjutan sebelum menghubungkan ESP ke WiFi dalam mode stasiun. Aplikasi lain yang berguna dari mode soft-AP adalah untuk mengatur jaringan mesh. ESP dapat beroperasi dalam mode soft-AP dan Station sehingga dapat bertindak sebagai node dari jaringan mesh.

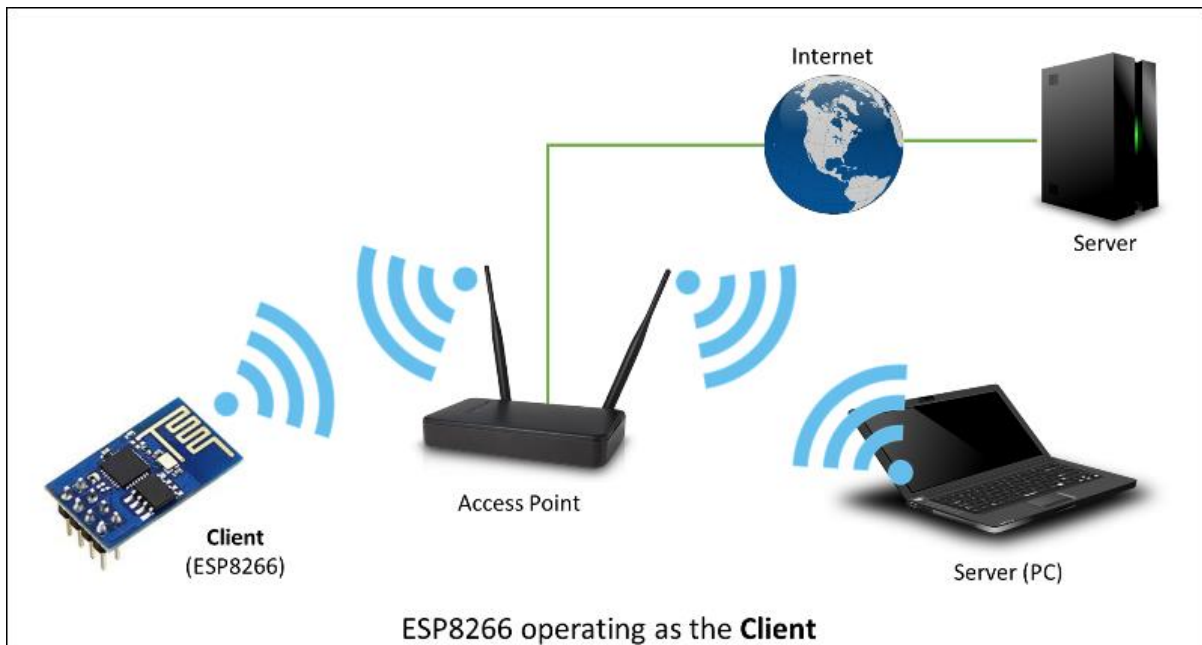
[Library Class ESP8266 sebagai Soft-Access Point:](#)



## Client

Kelas Client menciptakan klien yang dapat mengakses layanan yang disediakan oleh server untuk mengirim, menerima, dan memproses data.

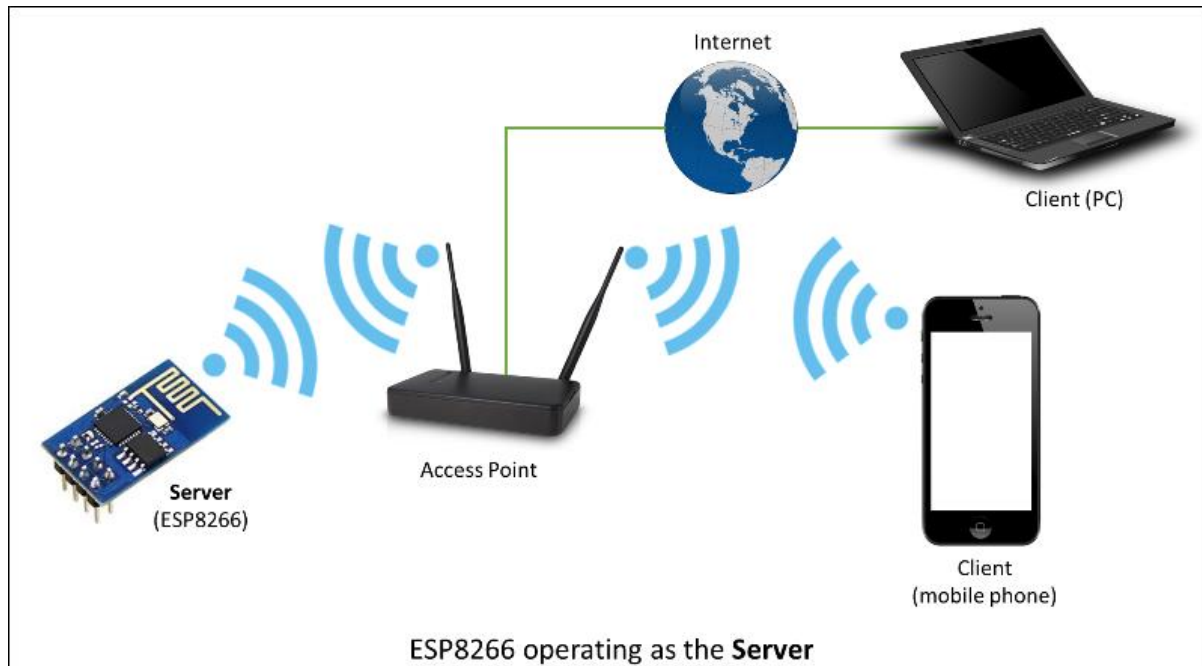
[Library Class ESP8266 sebagai Client](#)



## Server

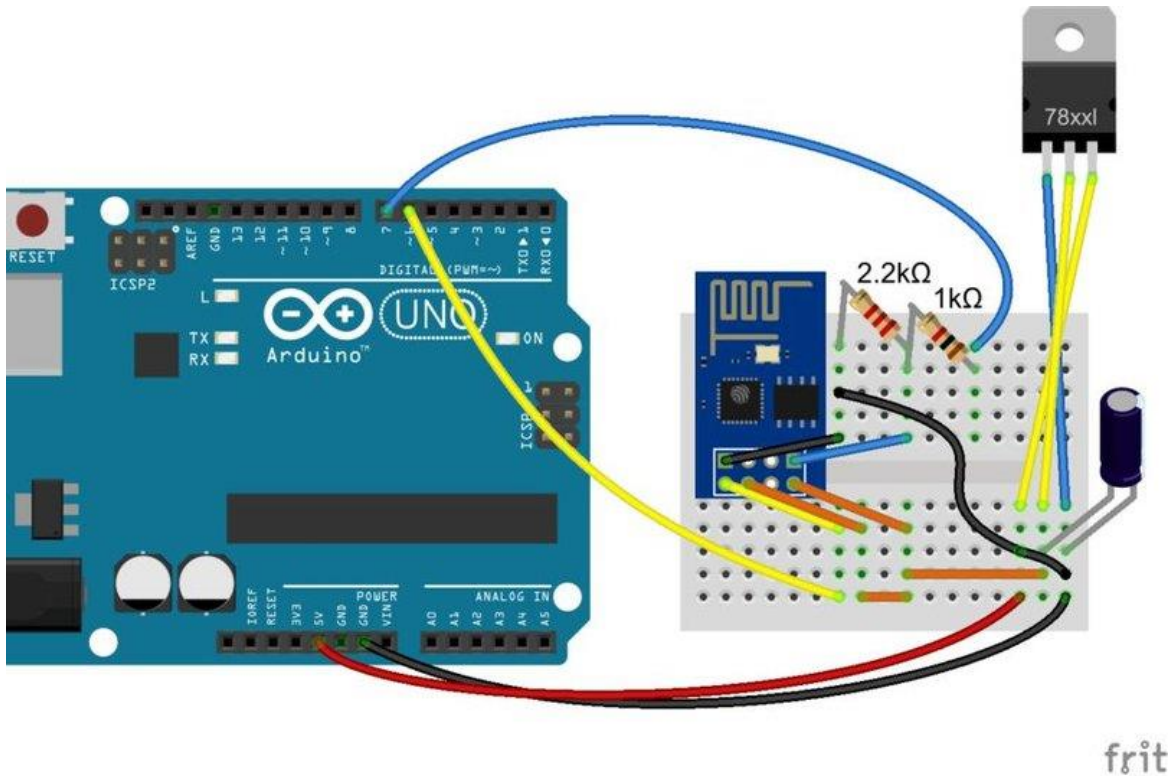
Kelas Server membuat server yang menyediakan fungsionalitas ke program atau perangkat lain, yang disebut klien. Klien terhubung ke memutuskan untuk mengirim dan menerima data dan mengakses fungsionalitas yang disediakan.

### [Library Class ESP866 sebagai Server](#)



## E. Praktikum

1. Buat rangkaian sehingga seperti dalam skematik di bawah ini



2. Dalam skematis dijelaskan cara menghubungkan semua komponen menggunakan Arduino Uno dan mini breadboard.
  - a. Pin VCC ESP didukung oleh pin output 3.3V dari regulator tegangan (AMS1117). Kapasitor 10uF terhubung ke pin output untuk menstabilkan regulator. Pin CH\_PD juga harus terhubung ke 3.3V. Pin GND jelas terhubung ke ground.
  - b. Pin TXD ESP dapat dihubungkan langsung ke pin RX Arduino (diemulasikan pada pin 6).
  - c. Pin RXD ESP terhubung ke pin TX Arduino (ditiru pada pin 7) melalui tingkat shifter.
3. Untuk mencoba koneksi maka tulis kode sumber sebagai berikut

```
// EspDebug - Test sketch for ESP8266 module

// emulate Serial1 if not present
#ifndef HAVE_HWSERIAL1
  #include "SoftwareSerial.h"
  SoftwareSerial Serial1(6, 7); // RX, TX
#endif

void setup()
{
```

```

Serial.begin(115200); // serial port used for debugging
Serial1.begin(9600); // your ESP's baud rate might be different
}

void loop()
{
  if(Serial1.available()) // check if the ESP is sending a message
  {
    while(Serial1.available())
    {
      int c = Serial1.read(); // read the next character
      Serial.write((char)c); // writes data to the serial monitor
    }
  }

  if(Serial.available())
  {
    // wait to let all the input command in the serial buffer
    delay(10);

    // read the input command in a string
    String cmd = "";
    while(Serial.available())
    {
      cmd += (char)Serial.read();
    }

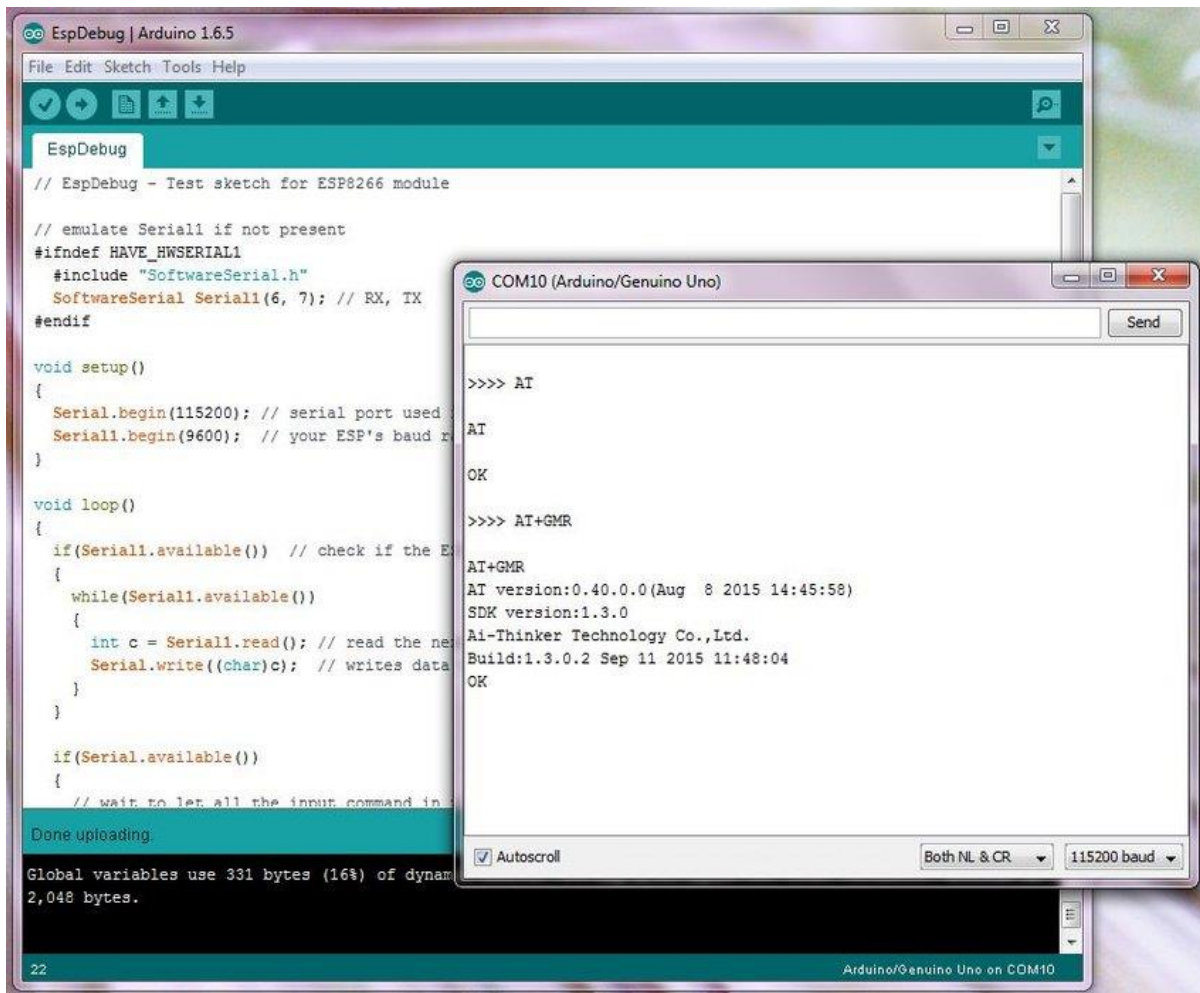
    // print the command and send it to the ESP
    Serial.println();
    Serial.print(">>>> ");
    Serial.println(cmd);

    // send the read character to the ESP
    Serial1.print(cmd);
  }
}

```

4. Buka serial port monitor dan ketikkan AT command yaitu: AT dan AT+GMR, hingga seperti pada gambar berikut:





5. Buat konfigurasi ESP826 ke dalam mode Client dan masukkan contoh sebagai berikut.

```
#include <ESP8266WiFi.h>

const char* ssid = "*****";
const char* password = "*****";

const char* host = "www.example.com";

void setup()
{
  Serial.begin(115200);
  Serial.println();

  Serial.printf("Connecting to %s ", ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
```

```

    {
        delay(500);
        Serial.print(".");
    }
    Serial.println(" connected");
}

void loop()
{
    WiFiClient client;

    Serial.printf("\n[Connecting to %s ... ", host);
    if (client.connect(host, 80))
    {
        Serial.println("connected");

        Serial.println("[Sending a request]");
        client.print(String("GET /") + " HTTP/1.1\r\n" +
            "Host: " + host + "\r\n" +
            "Connection: close\r\n" +
            "\r\n"
        );

        Serial.println("[Response:]");
        while (client.connected())
        {
            if (client.available())
            {
                String line = client.readStringUntil('\n');
                Serial.println(line);
            }
        }
        client.stop();
        Serial.println("\n[Disconnected]");
    }
    else
    {
        Serial.println("connection failed!");
        client.stop();
    }
    delay(5000);
}

```

## F. Tugas

Buat secara berkelompok aplikasi Arduino yang dapat digunakan untuk membaca masukan (input) dari sensor dan mengirimkan data tersebut dengan bersama dengan alamat Arduino dan ESP8266 pada mode Client.