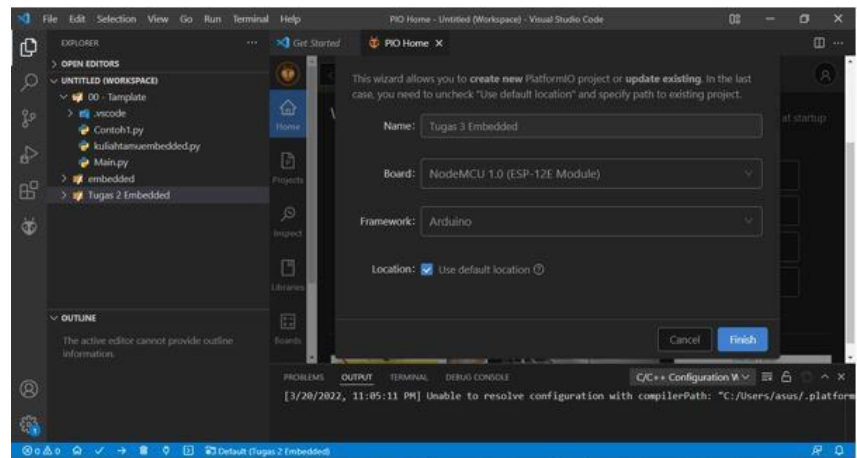
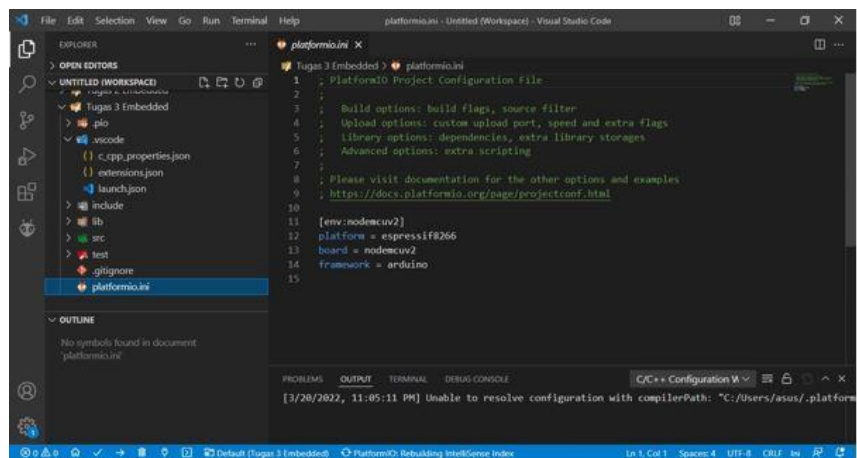


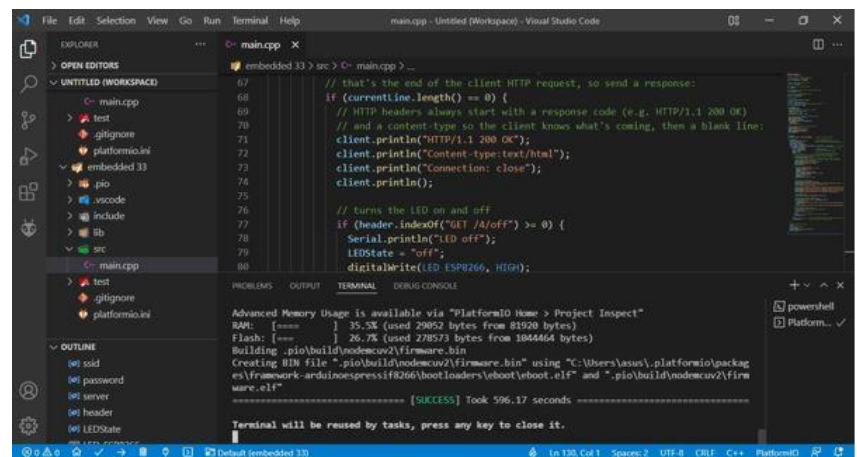
New PlatformIO project dengan nama sesuai user, dengan Board NodeMCU 1.0 untuk ESP8266, dan Framework Arduino.



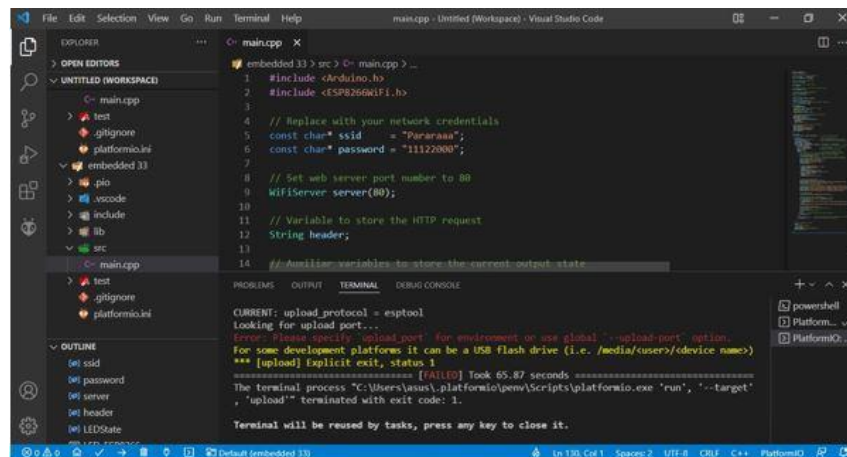
Cek pada file platform.io untuk memastikan bahwa environment yang digunakan sesuai pada platform sudah sesuai untuk ESP8266 yaitu espressif8266, board yaitu nodemcu2, dan framework yaitu arduino.



Build dari kode program yang digunakan.



Upload kode program pada ESP8266.



PENJELASAN KODE PROGRAM

1. Pastikan library yang digunakan adalah ESP8266WiFi.h
2. Berisi ssid yang berisi nama jaringan dari tethering atau modem yang sedang terhubung, dan password yang digunakan. Yang akan digunakan oleh ESP untuk merequest alamat IP
3. Terdapat sebuah server, dengan klasifikasi server pada port 80. Port 80 digunakan untuk HTTP, untuk port 443 HTTPS.
4. Membuat variabel dengan nama header, yang akan digunakan untuk menampung HTTP request.
5. Sebagai opsi tambahan, untuk memberitahu state LED pada port NodeMCU adalah off.
6. Mendefine sebuah konstanta LED_ESP8266, dalam NodeMCU LED build in-nya ada di Digital 4 (D4).
7. Membuat millis, untuk memberikan waktu jeda agar tidak terlalu lama, sehingga terbatas agar tidak terjadi lag
8. Void setup adalah fungsi yang pertama kali dipanggil, menggunakan serial 9600 untuk memberikan status pada server NodeMCU. pinMode untuk memberikan mode pada digital 4 untuk output. digitalWrite untuk menyalakan.
9. Selama NodeMCU belum mendapat alamat IP, tidak ada koneksi maka akan menunggu, sehingga ada perulangan while, dengan delay 500. Apabila sudah connected, antara NodeMCU dan router atau internet kemudian akan lanjut ke print local IP.
10. Kemudian akan di println pada serial, bahwa sekarang mendapatkan alamat IP sedemikian. Setelah mendapat alamat IP, maka langsung menjalankan server dengan server.begin().
11. Pada void loop akan membuat sebuah halaman websitenya. Membuat variabel dengan nama client, dimana variabel ini nantinya akan menunggu adakah client yang connect atau adakah client yang masuk pada NodeMCU atau adakah client yang merequest halaman web. Apabila ada akan dibuat halamannya.
12. Kemudian dibuat timeout agar tidak terlalu lama.
13. Apabila belum ada halaman websitenya, maka dibuat sebuah header yang akan dikirimkan untuk client.
14. Membuat sebuah logika, apabila client mengakses alamat IP on maka akan memanggil header. Index dan serial akan memberikan respon nyala. Apabila client mengakses alamat IP off maka NodeMCU maka LED akan off dan LED pada ESP akan mati. Logika sederhana yang digunakan untuk handle client request berdasarkan alamat yang dikirim pada server yaitu NodeMCU.

15. Membuat head menggunakan html.
16. Membuat style, seperti button berwarna biru, terdapat tulisan
17. Jika LED state off, apabila LED di build in NodeMCU off maka akan menampilkan button dengan tulisan on, yang artinya siap di on-kan. Kalau tidak off, maka akan menampilkan button dengan tulisan off, yang siap untuk di off-kan.
18. Akhir dari halaman HTML.
19. Break berarti menunggu perintah selanjutnya.
20. Apabila sudah tidak ada request lagi pada NodeMCU maka akan menutup koneksinya. Pada serial akan ditambikan bahwa client sudah tidak terkoneksi lagi.