

PRAKTIK DASAR ESP32 BOARD

A. Tujuan

1. Memahami dasar mikrokontroler.
2. Menggunakan fitur Serial Monitor di Arduino IDE.
3. Menerapkan pemrograman dasar pada mikrokontroler untuk mengendalikan perangkat elektronik sederhana.
4. Menganalisis bagaimana kendali perangkat elektronik dapat diterapkan dalam kehidupan sehari-hari.

B. Teori Dasar

Mikrokontroler mempunyai sejumlah pin GPIO yang dapat dipilih dan difungsikan sebagai input maupun *output*. Pada bab ini akan dilakukan praktik akses pin GPIO sebagai *output* untuk mengendalikan LED. Penggunaan fitur Serial Monitor di Arduino IDE untuk menampilkan *output* virtual dengan pesan yang telah diprogram serta mengendalikan LED melalui Serial Monitor.

Struktur pemrograman dasar dari Arduino yaitu seperti gambar berikut. Terdapat dua perbedaan utama antara keduanya, sebagai berikut.

- `void setup()` digunakan untuk inisialisasi awal (pin, komunikasi, sensor), disini program dieksekusi hanya sekali.
- `void loop()` digunakan untuk menulis logika utama program, disini program dieksekusi berulang selama perangkat aktif.

```
1 void setup() {  
2     // put your setup code here, to run once:  
3  
4 }  
5  
6 void loop() {  
7     // put your main code here, to run repeatedly:  
8  
9 }  
10
```

C. Alat & Bahan

- a. ESP32
- b. LED 5mm
- c. Breadboard
- d. Kabel jumper
- e. Laptop terinstal Arduino IDE
- f. Kabel USB Type C

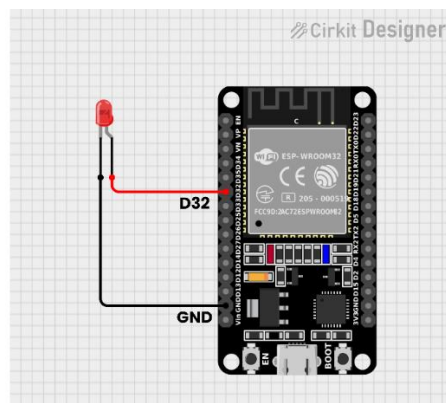
D. Keselamatan Kerja

1. Lakukan, atau laksanakan praktikum di tempat yang bersih dan kering.
2. Menggunakan laptop, atau komputer dengan baik, dan benar.
3. Jangan sentuh langsung bagian kit praktikum yang dialiri listrik.
4. Pastikan *power supply* kit praktikum dalam keadaan mati ketika proses instalasi, atau perakitan kit praktikum.
5. Selalu perhatikan *port* kit praktikum. Jangan sampai tertukar, terbalik, atau salah penempatan dalam instalasi, atau perakitan pada kit praktikum.
6. Gunakan komponen pada kit praktikum sebagaimana mestinya.
7. Jika terjadi *short circuit*, segera matikan *power supply* pada kit praktikum.
8. Mengikuti langkah-langkah praktikum seperti yang tertera pada langkah kerja serta memperhatikan keselamatan kerja.

E. Langkah Kerja

5.1 *Blinking LED*

- a. Buat rangkaian seperti gambar berikut. Pastikan gunakan pin GPIO yang mendukung *output*.



b. Masukkan kode berikut.

```
const int ledPin1 = 32; // inisialisasi led di pin GPIO 32

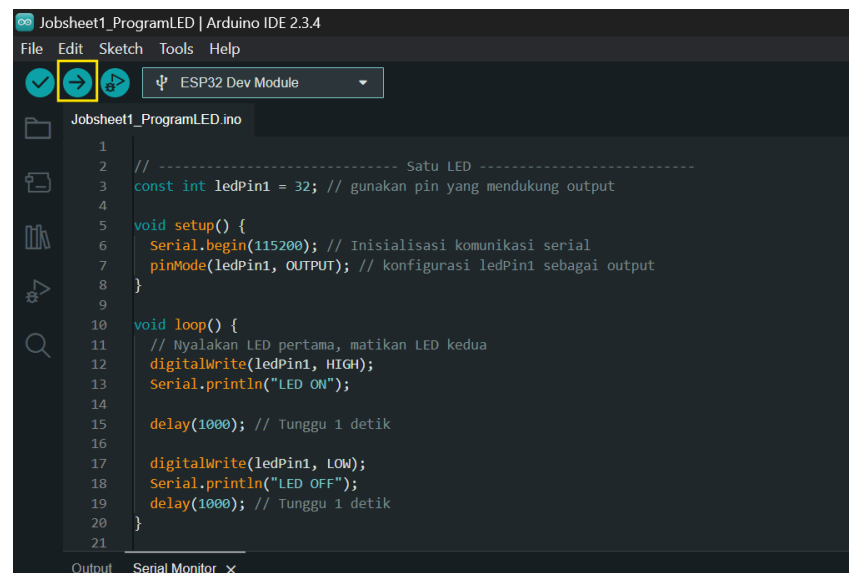
void setup() {
  Serial.begin(115200); // inisialisasi komunikasi serial
  pinMode(ledPin1, OUTPUT); // konfigurasi ledPin1 sebagai output
}

void loop() {
  // Nyalakan LED
  digitalWrite(ledPin1, HIGH);
  Serial.println("LED ON"); // Menampilkan pesan
  delay(1000); // Tunggu 1 detik

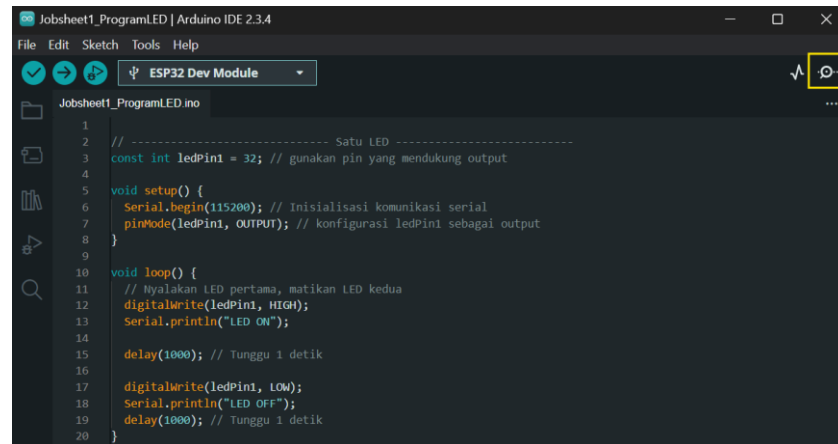
  // Matikan LED
  digitalWrite(ledPin1, LOW);
  Serial.println("LED OFF"); // Menampilkan pesan
  delay(1000); // Tunggu 1 detik
}
```

c. Hubungkan ESP32 ke laptop, sebelum Verify atau Compile code pastikan Board ESP32 sudah terhubung di Arduino IDE.

d. Apabila proses compile tidak ada error, lanjut untuk upload program seperti gambar berikut.

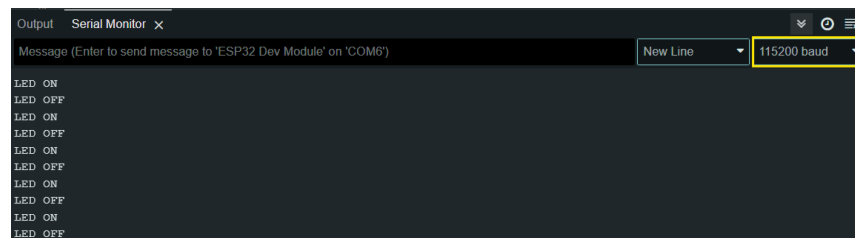


e. Proses upload berhasil, kita bisa melihat hasil program melalui jendela Serial Monitor, ikon sebelah kanan atas.



```
1 // ----- Satu LED -----
2
3 const int ledPin1 = 32; // gunakan pin yang mendukung output
4
5 void setup() {
6   Serial.begin(115200); // Inisialisasi komunikasi serial
7   pinMode(ledPin1, OUTPUT); // konfigurasi ledPin1 sebagai output
8 }
9
10 void loop() {
11   // Nyalakan LED pertama, matikan LED kedua
12   digitalWrite(ledPin1, HIGH);
13   Serial.println("LED ON");
14
15   delay(1000); // Tunggu 1 detik
16
17   digitalWrite(ledPin1, LOW);
18   Serial.println("LED OFF");
19   delay(1000); // Tunggu 1 detik
20 }
```

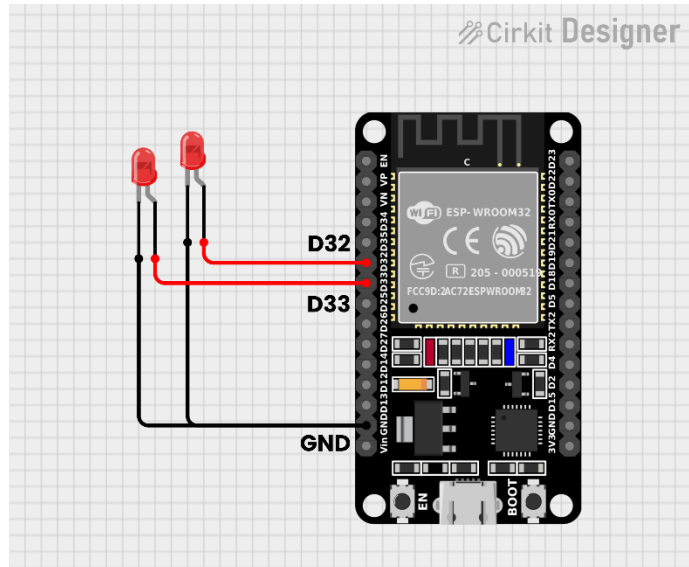
- f. Di jendela Serial Monitor diperlukan sinkronisasi *baud rate* untuk komunikasi serial. Ubah *baud rate* menjadi **115200** agar dapat menampilkan pesan sesuai yang telah diprogram.



```
Output Serial Monitor x
Message (Enter to send message to 'ESP32 Dev Module' on 'COM6')
New Line 115200 baud
LED ON
LED OFF
LED ON
LED OFF
LED ON
LED OFF
LED ON
LED OFF
LED ON
LED OFF
```

5.2 *Blinking* Dua LED

- a. Buat rangkaian seperti gambar berikut. Pastikan gunakan pin GPIO yang mendukung *output*.



- b. Masukkan kode berikut.

```
const int ledPin1 = 33; // inisialisasi LED di pin GPIO 33
const int ledPin2 = 32; // inisialisasi LED di pin GPIO 32

void setup() {
  pinMode(ledPin1, OUTPUT); // led 1 sebagai output
  pinMode(ledPin2, OUTPUT); // led 2 sebagai output
}

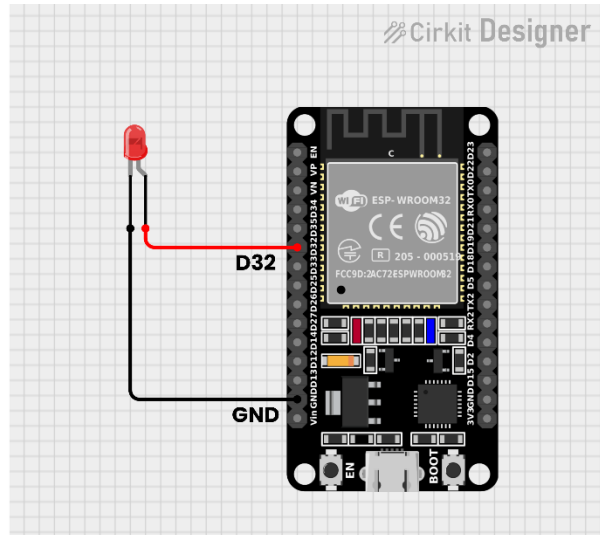
void loop() {
  // Nyalakan LED pertama, matikan LED kedua
  digitalWrite(ledPin1, HIGH);
  digitalWrite(ledPin2, LOW);
  delay(1000); // Tunggu 1 detik

  // Matikan LED pertama, nyalakan LED kedua
  digitalWrite(ledPin1, LOW);
  digitalWrite(ledPin2, HIGH);
  delay(1000); // Tunggu 1 detik
}
```

- c. Hubungkan ESP32 ke laptop, sebelum Verify atau Compile code pastikan Board ESP32 sudah terhubung di Arduino IDE.
- d. Apabila proses compile selesai dan tidak ada error, selanjutnya upload program, lalu tunggu hingga selesai.

5.3 Kontrol LED dari Serial Monitor

- a. Buat rangkaian seperti gambar berikut. Pastikan gunakan pin GPIO yang mendukung *output*.



- b. Masukkan kode berikut.

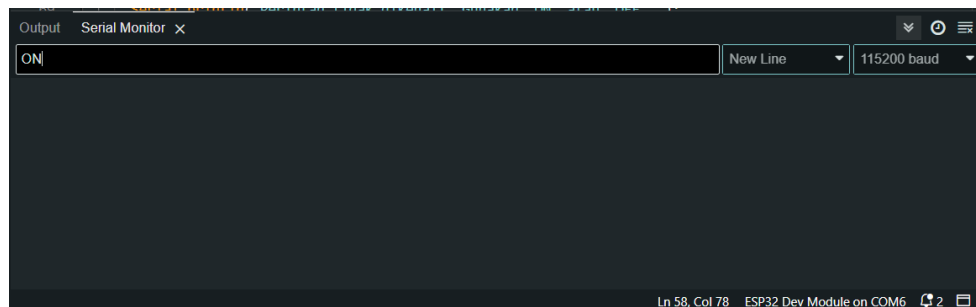
```
const int ledPin1 = 32; // Inisialisasi LED di pin GPIO 32

void setup() {
  pinMode(ledPin1, OUTPUT); // Konfigurasi LED sebagai output
  Serial.begin(115200); // Inisialisasi komunikasi serial
  Serial.println("Ketik 'ON' untuk menyalakan LED, 'OFF' untuk mematikan LED.");
}

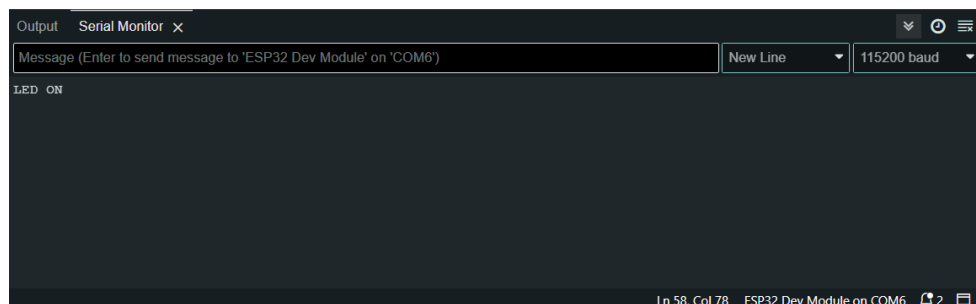
void loop() {
  if (Serial.available() > 0) { // Jika ada data dari Serial Monitor
    String command = Serial.readStringUntil('\n'); // Baca input hingga enter
    command.trim(); // Hapus karakter spasi atau newline

    if (command == "ON") {
      digitalWrite(ledPin1, HIGH); // Nyalakan LED
      Serial.println("LED ON");
    }
    else if (command == "OFF") {
      digitalWrite(ledPin1, LOW); // Matikan LED
      Serial.println("LED OFF");
    }
    else {
      Serial.println("Perintah tidak dikenali. Gunakan 'ON' atau 'OFF'.");
    }
  }
}
```

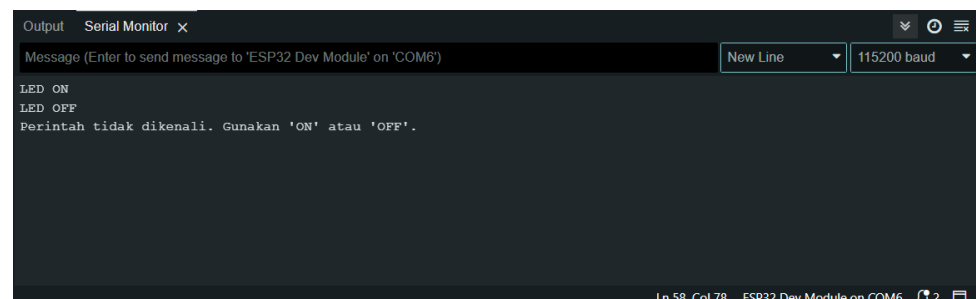
- c. Hubungkan ESP32 ke laptop, sebelum Verify atau Compile code pastikan Board ESP32 sudah terhubung di Arduino IDE.
- d. Apabila proses compile selesai dan tidak ada error, selanjutnya upload program, lalu tunggu hingga selesai.
- e. Masuk ke Serial Monitor, lalu ubah *baud rate* sesuai yang telah diprogram.
- f. Ketik perintah “ON” atau “OFF” seperti yang telah dikonfigurasi, untuk menyalakan atau mematikan LED. Setelah mengetik perintah klik **enter**.



- g. Apabila telah selesai mengirim perintah, maka tampil pesan “LED ON” dan LED akan menyala. Begitu sebaliknya ketika mengetik “OFF” maka akan mematikan LED dan menampilkan pesan “LED OFF”.



- h. Apabila terjadi salah ketik maka perintah tidak dapat dieksekusi, sehingga program akan menampilkan pesan error.



F. Soal

1. Tampilkan output berdasarkan dari instruksi Langkah Kerja! Tambahkan analisis dan prinsip kerjanya!
2. Buatlah program untuk mengontrol LED dengan durasi nyala atau mati yang dapat diatur melalui Serial Monitor!
3. Buatlah program seperti konsep lalu lintas menggunakan 3 buah LED, misalnya:
 - LED 1 menyala selama 3 detik, LED 2 dan 3 mati.
 - LED 2 menyala selama 3 detik, LED 1 dan 3 mati.
 - LED 3 menyala selama 3 detik, LED 1 dan 2 mati.
4. Dokumentasikan hasil pengerjaan pada lembar Kerja Praktikum. Sertakan video hasil praktikum dan source code-nya!
5. Upload lembar kerja Laporan Praktikum (dalam format PDF) dan video dokumentasi di Google Drive.
https://drive.google.com/drive/folders/1I2Dsi14Gmf9mdjtKauldQ7rw9M7A9IMr?usp=drive_link
6. Ketentuan pengumpulan drive sebagai berikut:
 - Gunakan akun students unnes
 - Buat folder dengan format “NIM_Nama Lengkap”
 - Upload lembar kerja Laporan Praktikum dan dokumentasi video