Logbook 1 : ESP32 dan Penerapan Blink Sketch

Nama : Albi Akhsanul hakim
NPM : 22081010194

Pada logbook pertama ini, saya mempelajari dasar penggunaan ESP32 dan bagaimana menerapkan program sederhana untuk membuat LED berkedip menggunakan blink sketch. ESP32 merupakan mikrokontroler berbiaya rendah dan hemat daya yang dilengkapi dengan kemampuan Wi-Fi dan Bluetooth bawaan. Perangkat ini sangat cocok digunakan untuk berbagai proyek IoT (Internet of Things).

Berikut adalah contoh kode yang digunakan pada ESP32:

```
// Pin LED yang akan digunakan
const int ledPin = 2;

void setup() {
  // Inisialisasi pin LED sebagai output
  pinMode(ledPin, OUTPUT);
  Serial.begin(115200);
}

void loop() {
  // Menyalakan LED
  digitalWrite(ledPin, HIGH);
  Serial.println("LED ON");
  delay(500); // Tunda selama 0,5 detik

  // Mematikan LED
  digitalWrite(ledPin, LOW);
  Serial.println("LED OFF");
  delay(1000); // Tunda selama 1 detik

  // Variasi delay untuk efek berkedip yang berbeda
  digitalWrite(ledPin, HIGH);
  Serial.println("LED ON - Cepat");
  delay(100); // Tunda singkat
  digitalWrite(ledPin, LOW);
  Serial.println("LED OFF - Cepat");
  delay(100);

  digitalWrite(ledPin, HIGH);
  Serial.println("LED ON - Sedang");
  delay(250); // Tunda sedang
  digitalWrite(ledPin, LOW);
  Serial.println("LED OFF - Sedang");
  delay(250);
}
```

Analisis:
Program di atas merupakan implementasi dari konsep dasar blink sketch, yang bertujuan untuk membuat LED berkedip secara berulang. Dalam penerapan ini, saya menggunakan pin GPIO2 sebagai output untuk mengontrol LED. Fungsi digitalWrite() digunakan untuk memberikan sinyal HIGH dan LOW secara bergantian, sedangkan delay() digunakan untuk mengatur durasi nyala dan mati LED.

Selain pola berkedip standar, saya juga menambahkan beberapa variasi delay untuk menciptakan pola kedipan yang berbeda, seperti cepat dan sedang. Hal ini bertujuan untuk menguji respons LED terhadap perubahan kecepatan sinyal. Output dari setiap status LED juga ditampilkan melalui Serial Monitor menggunakan Serial.println(), sehingga memudahkan proses monitoring dan debugging.

Penerapan ini berhasil dijalankan pada ESP32, dan menjadi langkah awal yang penting dalam memahami cara kerja dasar mikrokontroler serta integrasi antara perangkat keras dan perangkat lunak.