

LAPORAN TUGAS MATA KULIAH MIKROPROSESOR DAN ANTARMUKA
PENGUNAAN BOARD STM32 UNTUK MENGONTROL NYALA / MATI LED
MELALUI SAKLAR

Dosen Pengampu : Dr. Eng. ASIF AWALUDIN, S.T., M.Sc.



Disusun oleh :

Sahrul Ridho Firdaus	1103223009
Vinsensius Jonathan Fajarai	1103220179
Dewo Antonioly	1103223092
Ferensia Fransisca Agatha	1103223036
Grace Agustina Hutagalung	1103223089

PROGRAM STUDI TEKNIK KOMPUTER
FAKULTAS TEKNIK ELEKTRO
UNIVERSITAS TELKOM
BANDUNG
2024

BAB I

1.1 Latar Belakang

Perkembangan teknologi embedded system yang semakin pesat telah meningkatkan kebutuhan akan perangkat mikrokontroler yang mampu menjalankan berbagai fungsi dengan efisien dan fleksibel. Dalam banyak aplikasi modern, terutama yang memerlukan kontrol real-time, mikrokontroler dengan kemampuan komputasi tinggi, konsumsi daya rendah, dan beragam fitur antarmuka sangat dibutuhkan. Salah satu platform yang semakin populer dan banyak digunakan dalam proyek embedded system adalah STM32 board.

STM32 board, yang dikembangkan oleh STMicroelectronics, adalah keluarga mikrokontroler berbasis ARM Cortex-M yang dirancang untuk mendukung aplikasi yang membutuhkan performa tinggi dan keandalan. Berbagai seri STM32, seperti STM32F, STM32L, dan STM32H, menawarkan kombinasi fitur seperti kecepatan prosesor tinggi, kemampuan pemrosesan sinyal digital, antarmuka komunikasi yang luas (I2C, SPI, UART, USB, Ethernet), serta konsumsi daya yang rendah. Hal ini menjadikan STM32 board sebagai pilihan utama dalam aplikasi di berbagai bidang, seperti otomasi industri, perangkat medis, IoT, dan sistem kontrol motor.

Namun, penggunaan STM32 board memerlukan pemahaman yang mendalam tentang konfigurasi perangkat keras dan perangkat lunak. Laporan ini disusun dengan tujuan untuk memberikan gambaran menyeluruh tentang fitur, cara penggunaan, sehingga diharapkan dapat menjadi referensi yang berguna dalam memahami dan mengimplementasikan perangkat ini.

1.2 Tujuan Penulisan

Tujuan Penulisan laporan ini adalah sebagai berikut :

1. Menjelaskan Prinsip Dasar Mikrokontroler STM32
2. Memberikan Panduan Langkah demi Langkah
3. Mengimplementasikan saklar pada STM32

BAB 2

2.1 Metode

GPIO, atau *General Purpose Input/Output*, adalah pin digital yang dapat diprogram pada berbagai jenis mikrokontroler atau prosesor untuk menerima atau mengirim sinyal. GPIO memungkinkan kontrol fleksibel terhadap perangkat keras eksternal, seperti LED, saklar, sensor, dan motor, yang terhubung ke sistem. Setiap GPIO dapat dikonfigurasi sebagai input atau output tergantung pada kebutuhan aplikasi.

Sebagai input, GPIO dapat membaca status atau sinyal digital dari perangkat eksternal, misalnya mendeteksi apakah sebuah saklar ditekan. Sebagai output, GPIO dapat digunakan untuk mengontrol perangkat, seperti menghidupkan atau mematikan LED. Pada beberapa sistem, GPIO juga dapat mengirimkan interupsi yang memungkinkan prosesor untuk merespons perubahan input secara instan, yang sangat berguna dalam aplikasi real-time.

2.2 Alat dan Bahan

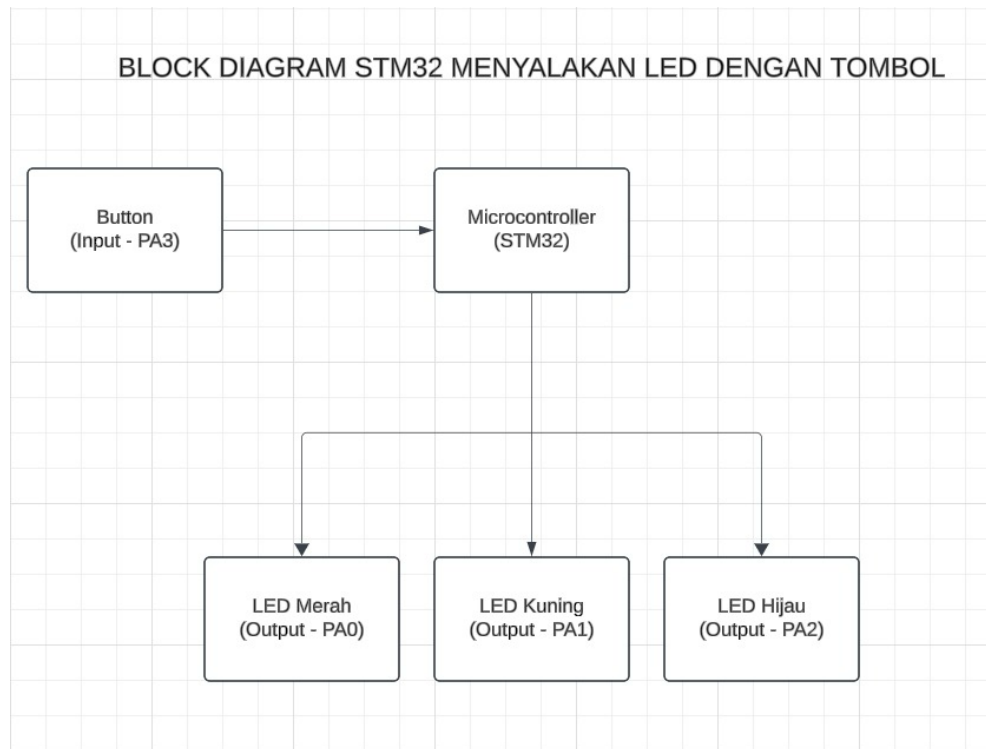
I. Alat

- a) STM32F103C8T6 - Bluepill
- b) STLINK
- c) Breadboard

II. Bahan

- a) Led
- b) Resistor 220 ohm
- c) Kabel jumper
- d) Tombol atau saklar

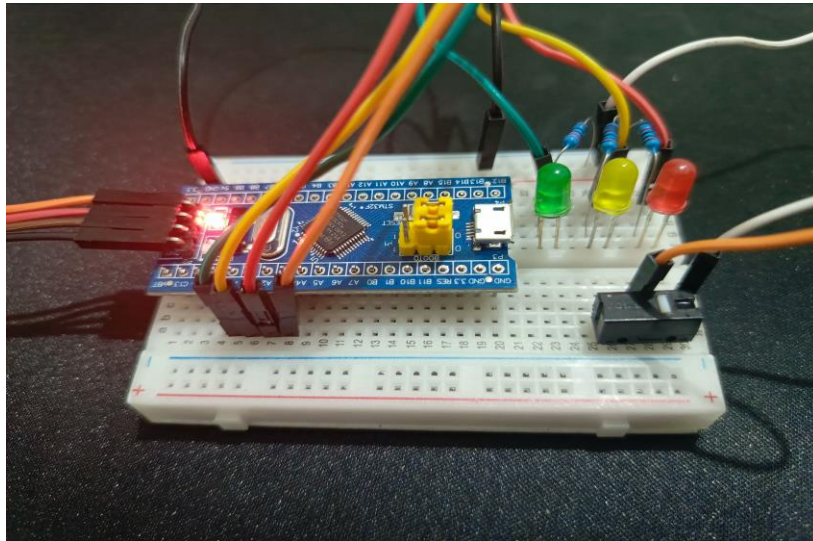
2.3 Blok Diagram



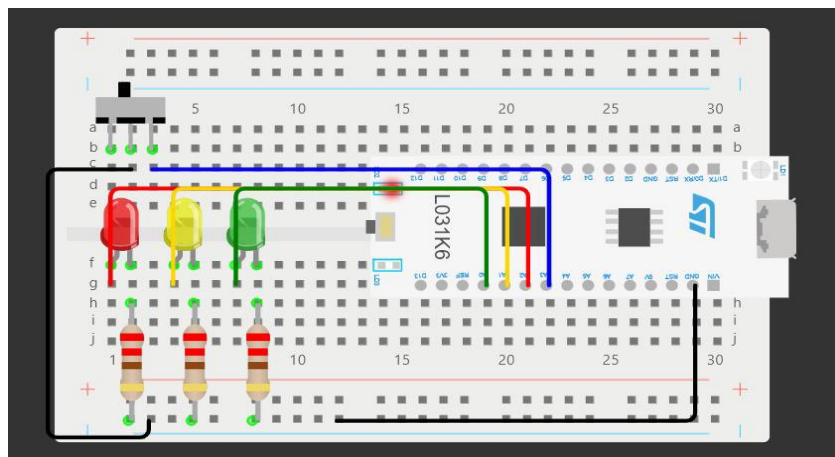
BAB 3

3.1 Hasil Eksperimen

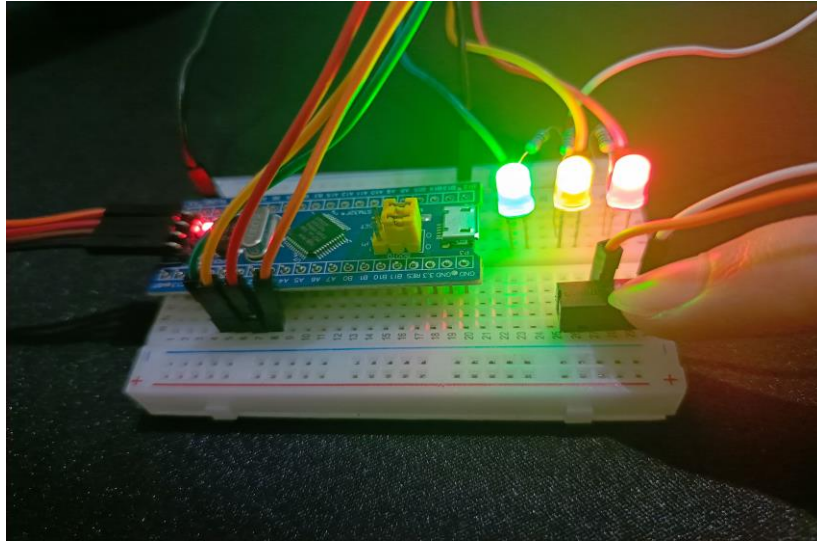
Berikut adalah lampiran gambar yang diambil saat eksperimen berlangsung



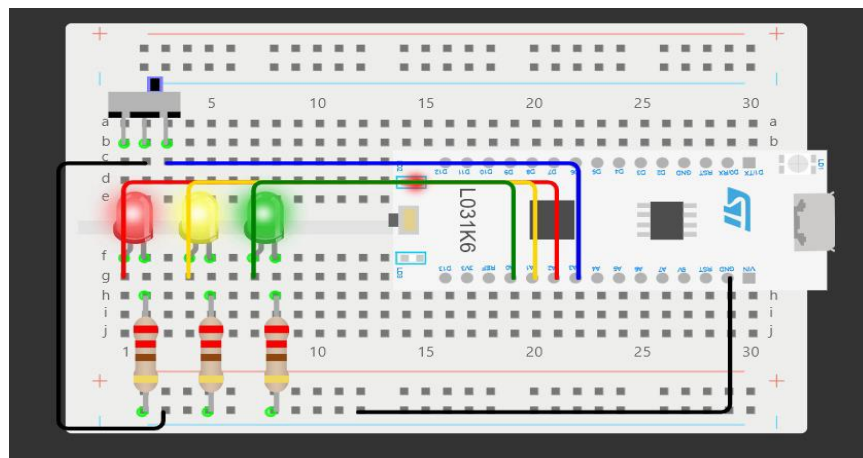
Eksperimen 1- Tombol off



Eksperimen 2- Tombol off simulasi



Eksperimen 3-Tombol on



Eksperimen 4-Tombol off simulasi

3.2 Lampiran Source Code

```
1 int ledmerah = PA0; // Mendefinisikan pin untuk LED merah
2 int ledkuning = PA1; // Mendefinisikan pin untuk LED kuning
3 int ledhijau = PA2; // Mendefinisikan pin untuk LED hijau
4 int buttonPin = PA3; // Mendefinisikan pin untuk tombol
5
6 void setup() {
7     pinMode(ledmerah, OUTPUT); // Mengatur pin LED merah sebagai OUTPUT
8     pinMode(ledkuning, OUTPUT); // Mengatur pin LED kuning sebagai OUTPUT
9     pinMode(ledhijau, OUTPUT); // Mengatur pin LED hijau sebagai OUTPUT
10    pinMode(buttonPin, INPUT_PULLUP); // Mengatur pin tombol sebagai INPUT dengan resistor pull-up internal
11
12    digitalWrite(ledmerah, LOW); // Memastikan LED merah dimatikan
13    digitalWrite(ledkuning, LOW); // Memastikan LED kuning dimatikan
14    digitalWrite(ledhijau, LOW); // Memastikan LED hijau dimatikan
15 }
16
17 void loop() {
18     // Mengecek status tombol; jika tombol ditekan, pin akan membaca LOW
19     if (digitalRead(buttonPin) == LOW) {
20         digitalWrite(ledmerah, HIGH); // Menghidupkan LED merah
21         digitalWrite(ledkuning, HIGH); // Menghidupkan LED kuning
22         digitalWrite(ledhijau, HIGH); // Menghidupkan LED hijau
23     }
24     else {
25         digitalWrite(ledmerah, LOW); // Mematikan LED merah
26         digitalWrite(ledkuning, LOW); // Mematikan LED kuning
27         digitalWrite(ledhijau, LOW); // Mematikan LED hijau
28     }
29 }
```

3.3 Penjelasan Source Code Program

Kodingan ini mengontrol tiga LED (merah, kuning, dan hijau) menggunakan tombol. Pertama, variabel `ledmerah`, `ledkuning`, `ledhijau`, dan `buttonPin` digunakan untuk menetapkan pin PA0, PA1, PA2, dan PA3 untuk masing-masing LED dan tombol. Di dalam `setup()`, setiap pin LED diatur sebagai OUTPUT dengan `pinMode()`, sedangkan pin tombol diatur sebagai `INPUT_PULLUP`, yang menggunakan resistor pull-up internal untuk memastikan pin membaca HIGH saat tombol tidak ditekan.

Kemudian, setiap LED dimatikan di awal dengan `digitalWrite()` pada LOW. Dalam `loop()`, kondisi tombol dicek menggunakan `digitalRead(buttonPin)`: jika tombol ditekan (membaca LOW), ketiga LED diberi sinyal HIGH dengan `digitalWrite()` untuk menyala. Jika tombol dilepas (membaca HIGH), LED dimatikan kembali dengan sinyal LOW.