

MODUL AJAR DASAR MIKROKONTROLLER



Disusun oleh:
Tim CV. Kreasi Industri Nusantara
2025

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan penyusunan Modul Praktikum Mikrokontroler ini dengan baik.

Modul praktikum ini disusun untuk memberikan panduan kepada mahasiswa dalam memahami konsep dasar dan penerapan mikrokontroler khususnya ESP32, yang merupakan salah satu perangkat populer dalam pengembangan sistem embedded system dan Internet of Things (IoT).

Trainer kit yang digunakan dalam praktikum ini telah dilengkapi dengan berbagai komponen sensor dan aktuator, antara lain: relay, motor DC, servo, buzzer, potensiometer, rotary encoder, ultrasonic sensor, DS18B20, MAX30102, LED traffic light, LCD, OLED, serta modul komunikasi CAN Bus. Dengan kelengkapan tersebut, diharapkan mahasiswa dapat mempelajari berbagai aplikasi nyata mikrokontroler dalam bidang otomasi, monitoring, dan komunikasi data.

Adapun modul ini terdiri dari beberapa bab yang tersusun secara sistematis mulai dari pengenalan ESP32, persiapan lingkungan pemrograman, percobaan dasar input/output, penggunaan sensor, pengendalian aktuator, hingga pembuatan mini project. Setiap bab dilengkapi dengan tujuan praktikum, teori singkat, peralatan yang digunakan, langkah kerja, serta contoh program yang dapat langsung diuji pada trainer.

Penulis menyadari bahwa modul ini masih jauh dari sempurna. Oleh karena itu, saran dan kritik yang membangun sangat diharapkan demi penyempurnaan modul ini di masa mendatang. Semoga modul ini dapat memberikan manfaat yang sebesar-besarnya bagi mahasiswa, dosen, maupun pihak lain yang ingin mempelajari mikrokontroler ESP32.

Akhir kata, semoga modul praktikum ini dapat menjadi sarana pembelajaran yang efektif, interaktif, dan aplikatif dalam mendukung penguasaan teknologi mikrokontroler.

Surabaya, 28 Agustus 2025

Tim Kreasi Industries

DAFTAR ISI

KATA PENGANTAR.....	1
DAFTAR ISI.....	2
BAB I Pendahuluan.....	4
1.1 Latar Belakang.....	4
1.2 Tujuan.....	4
1.3 Alat dan Bahan.....	5
1.4 Sistematika Modul.....	5
BAB II Persiapan Lingkungan Pemrograman.....	6
2.1 Instalasi Arduino IDE.....	6
2.2 Menambahkan Board ESP32.....	6
2.3 Menghubungkan ESP32 ke Komputer.....	6
2.4 Instalasi Library Tambahan.....	7
2.5 Uji Coba Program Dasar.....	7
BAB III Pengenalan ESP32.....	8
3.1 Gambaran ESP32.....	8
3.2 PinOut ESP32 DEV Kit V1.....	8
3.3 Diagram Pinout ESP32 DevKit V1.....	9
3.4 Kelebihan ESP32.....	10
BAB IV Praktikum Dasar GPIO.....	12
4.1 Pengantar.....	12
4.2 Tujuan.....	12
4.3 Kontrol Buzzer.....	12
4.4 Kontrol Lampu Lalu Lintas (Traffic Light) Sederhana.....	14
4.5 Input Push Button.....	17
BAB V Praktikum Dasar Sensor.....	21
5.1 Pengantar.....	21
5.2 Tujuan.....	21
5.3 Penggunaan Sensor Ultrasonic.....	21
5.4 Interface Sensor MAX30102 Heart Rate Saturation Sensor.....	24
5.5 Interface sensor suhu water proof DS18B20.....	27
5.6 Input Potensiometer.....	29
5.7 Input Rotary Encoder.....	32
BAB VI Praktikum Aktuator.....	35
6.1 Pengantar.....	35
6.2 Tujuan.....	35
6.3 Kontrol Relay Module.....	36
6.4 Control Servo Emax.....	38
6.5 Control Motor DC + Driver.....	40
BAB VII Praktikum Display.....	44

7.1 Pengantar.....	44
7.2 Tujuan.....	44
7.3 Interface LCD 16x2.....	44
7.4 Interface OLED 2".....	46
BAB VIII Praktikum Komunikasi.....	49
8.1 Pengantar.....	49
8.2 Tujuan.....	49
8.3 TJA1050 (CAN Bus).....	49



BAB I

Pendahuluan

1.1 Latar Belakang

Perkembangan teknologi sistem tertanam (*embedded system*) semakin pesat seiring dengan kebutuhan otomasi, Internet of Things (IoT), dan perangkat cerdas. Mikrokontroler menjadi salah satu komponen utama dalam pengembangan teknologi tersebut karena berfungsi sebagai pengendali utama yang menghubungkan sensor, aktuator, dan sistem komunikasi.

ESP32 adalah salah satu mikrokontroler modern yang banyak digunakan. Mikrokontroler ini memiliki prosesor dual-core, konektivitas WiFi dan Bluetooth bawaan, serta mendukung berbagai protokol komunikasi (UART, SPI, I2C, CAN Bus). ESP32 sangat cocok digunakan baik dalam penelitian, pembelajaran, maupun pengembangan produk IoT.

Trainer kit yang digunakan dalam praktikum ini telah dilengkapi dengan berbagai komponen sensor dan aktuator seperti relay, motor DC, servo, buzzer, LCD, OLED, ultrasonic, rotary encoder, serta sensor detak jantung MAX30102. Dengan kelengkapan tersebut, mahasiswa dapat mempelajari berbagai aplikasi sistem mikrokontroler, mulai dari pengendalian sederhana hingga sistem monitoring berbasis IoT. Pada modul ini berisi dasar dari pemrograman dan wiring dari setiap komponen yang ada dalam Trainer kit.

1.2 Tujuan

Tujuan penyusunan modul ini adalah agar mahasiswa mampu:

1. Memahami dasar pemrograman mikrokontroler menggunakan ESP32.
 2. Menggunakan input/output digital maupun analog.
 3. Mengakses data sensor (ultrasonic, suhu, rotary encoder, MAX30102, dsb).
 4. Mengendalikan aktuator (relay, servo, motor DC, buzzer, lampu lalu lintas).
 5. Menampilkan data menggunakan LCD dan OLED.
-

-
6. Membuat mini project sederhana yang relevan dengan dunia IoT dan otomasi.

1.3 Alat dan Bahan

Modul kontroler yang digunakan dalam praktikum ini terdiri dari:

- **Mikrokontroler:** ESP32
- **Sensor:** Ultrasonic HC-SR04, DS18B20 (suhu), MAX30102 (heart rate & SpO₂), Rotary Encoder, Potensiometer 5K
- **Aktuator:** Relay Module, Servo Emax, Motor DC + Driver, Buzzer 5V
- **Display:** LCD 16x2, OLED 2"
- **Komunikasi:** TJA1050 (CAN Bus)
- **GPIO :** LED Traffic Light, Buzzer, 8 Push Button,
- **Tambahan:** Header, Adaptor 5V 5A, Kabel USB, Kabel Jumper

1.4 Sistematika Modul

Modul ini disusun dalam beberapa bab sebagai berikut:

- Bab I : Pendahuluan
- Bab II : Persiapan Lingkungan Pemrograman
- Bab III : Pengenalan ESP32
- Bab IV : Praktikum Dasar GPIO
- Bab V : Praktikum Sensor
- Bab VI : Praktikum Aktuator
- Bab VII : Praktikum Display
- Bab VIII : Praktikum Komunikasi

BAB II

Persiapan Lingkungan Pemrograman

2.1 Instalasi Arduino IDE

Untuk memprogram ESP32, digunakan **Arduino IDE** karena mudah digunakan dan memiliki dukungan library yang luas.

Langkah instalasi:

1. Unduh Arduino IDE dari <https://www.arduino.cc/en/software>.
 2. Install sesuai sistem operasi (Windows, Linux, Mac).
 3. Buka Arduino IDE setelah instalasi selesai.
-

2.2 Menambahkan Board ESP32

Agar Arduino IDE dapat mengenali ESP32, tambahkan Board Manager URL.

1. Buka menu File → Preferences.
Pada Additional Board Manager URLs, masukkan:
https://dl.espressif.com/dl/package_esp32_index.json
 2. Klik OK.
 3. Buka Tools → Board → Board Manager.
 4. Cari ESP32 by Espressif Systems → klik Install.
-

2.3 Menghubungkan ESP32 ke Komputer

1. Hubungkan ESP32 ke komputer menggunakan kabel USB data.
 2. Pastikan driver USB (CH340/CP2102) sudah terinstal.
 3. Pada Arduino IDE, pilih port:
 - Tools → Port → COMx (ESP32)
 4. Pilih board sesuai modul:
 - Tools → Board → ESP32 Dev Module
-

2.4 Instalasi Library Tambahan

Untuk mendukung semua sensor dan aktuator, tambahkan library berikut melalui Sketch → Include Library → Manage Libraries:

- LiquidCrystal_I2C → LCD 16x2
- Adafruit SSD1306 & GFX → OLED 2"
- NewPing → Ultrasonic
- ServoESP32 → Servo
- OneWire & DallasTemperature → DS18B20
- Adafruit MAX30102 → Heart Rate & SpO₂
- Encoder library → Rotary Encoder
- mcp_can → CAN Bus TJA1050



2.5 Uji Coba Program Dasar

Untuk memastikan instalasi berhasil, jalankan program Blink LED berikut:

Code :

```
int ledPin = 2; // GPIO2 (LED internal ESP32)
void setup() {
    pinMode(ledPin, OUTPUT);
}
void loop() {
    digitalWrite(ledPin, HIGH); // LED ON
    delay(1000);
    digitalWrite(ledPin, LOW); // LED OFF
    delay(1000); }
```

Jika LED internal ESP32 berkedip setiap 1 detik, maka instalasi berhasil.

BAB III

Pengenalan ESP32

3.1 Gambaran ESP32

ESP32 adalah sebuah mikrokontroler berbasis SoC (System on Chip) yang dikembangkan oleh Espressif Systems. ESP32 memiliki keunggulan dibandingkan mikrokontroler generasi sebelumnya (misalnya ESP8266 maupun Arduino UNO) karena dilengkapi dengan:

- Dual-core 32-bit Xtensa LX6 processor dengan kecepatan hingga 240 MHz.
- Wi-Fi 802.11 b/g/n terintegrasi.
- Bluetooth 4.2 dan BLE (Bluetooth Low Energy).
- RAM internal hingga 520 KB, serta dukungan eksternal flash.
- Banyak fitur periferal seperti ADC (Analog to Digital Converter), DAC (Digital to Analog Converter), PWM (Pulse Width Modulation), SPI, I2C, I2S, UART, dan sebagainya.

Karena kelengkapan fitur ini, ESP32 banyak digunakan pada aplikasi IoT (Internet of Things), home automation, sensor network, hingga kendali motor.

3.2 PinOut ESP32 DEV Kit V1

ESP32 DevKit V1 adalah salah satu board pengembangan yang umum digunakan. Board ini menggunakan chip ESP32-WROOM-32 dan menyediakan 30–38 pin GPIO (General Purpose Input/Output).

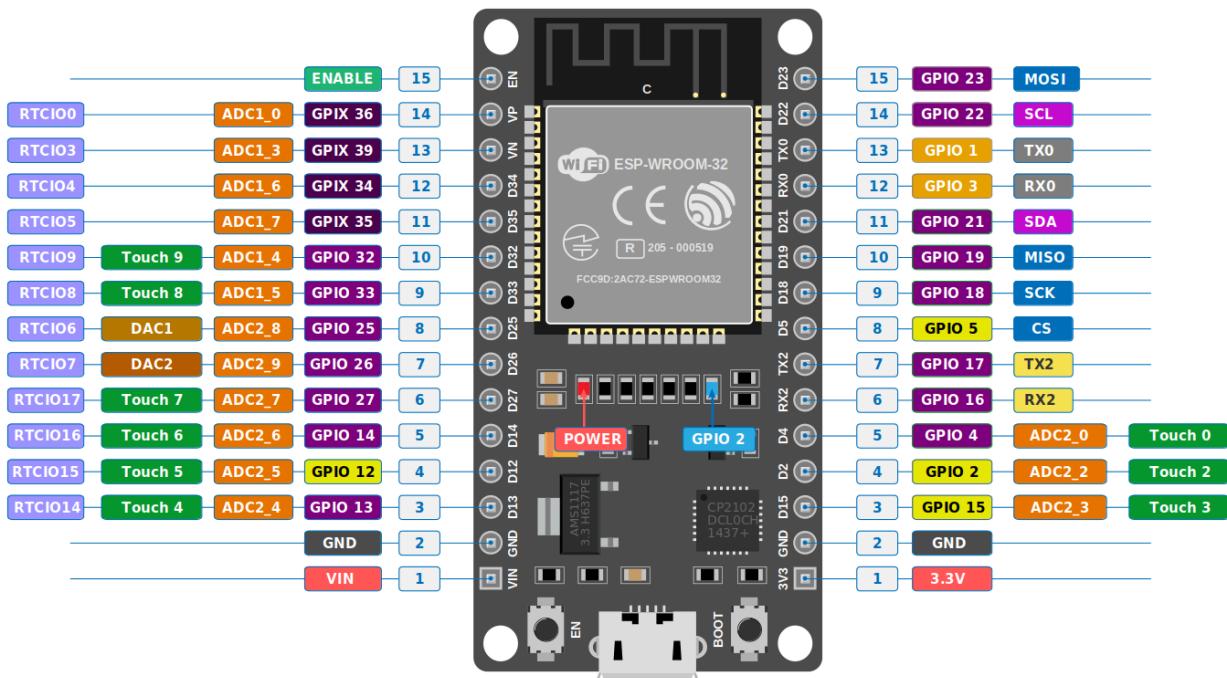
Berikut adalah kategori pin yang tersedia pada ESP32:

1. Power Pin
 - 3.3V → catu daya utama untuk ESP32.
 - 5V (VCC / VIN) → input daya dari USB atau adaptor.
 - GND → ground.

2. GPIO (General Purpose Input Output)
 - ESP32 memiliki lebih dari 30 pin GPIO, namun tidak semuanya bisa dipakai bebas.
 - Setiap GPIO dapat diprogram untuk fungsi berbeda (digital input, digital output, PWM, ADC, dll).
3. ADC (Analog to Digital Converter)
 - Tersedia hingga 18 channel ADC (GPIO 32–39, 0–15).
 - Resolusi hingga 12-bit.
 - Bisa membaca sensor analog seperti potensiometer, sensor cahaya, dll.
4. DAC (Digital to Analog Converter)
 - Tersedia pada GPIO 25 dan GPIO 26.
 - Menghasilkan tegangan analog (0–3.3V).
5. PWM (Pulse Width Modulation / LEDC)
 - Semua pin GPIO bisa digunakan untuk PWM.
 - Berguna untuk mengatur kecepatan motor DC, mengatur kecerahan LED, atau menghasilkan suara di buzzer.
6. Komunikasi
 - UART → komunikasi serial (GPIO1 TX, GPIO3 RX). Bisa juga diubah ke pin lain.
 - SPI → komunikasi dengan sensor/module kecepatan tinggi.
 - I2C → komunikasi dengan sensor sederhana (default: GPIO21 SDA, GPIO22 SCL).
 - I2S → komunikasi audio.

3.3 Diagram Pinout ESP32 DevKit V1

Berikut diagram pinout dari ESP32 Devkit V1



Sumber : <https://esp32.implrust.com/esp32-intro/pinout.html>

Gambar 1

Pin ESP3	Fungsi Utama	Fungsi Lain
GPIO0	Boot mode	Input/Output
GPIO2	Output default	PWM
GPIO4	GPIO bebas	PWM/ADC
GPIO12	Input/output	PWM
GPIO14	Input/output	PWM
GPIO21	SDA (I2C)	GPIO
GPIO22	SCL (I2C)	GPIO
GPIO25	DAC1	PWM
GPIO26	DAC2	PWM
GPIO32–39	ADC	Input only

3.4 Kelebihan ESP32

- Fitur WiFi dan Bluetooth terintegrasi → cocok untuk IoT.

- GPIO yang lebih banyak dibandingkan ESP8266.
 - Dapat menghasilkan PWM di hampir semua pin.
 - Konsumsi daya rendah (Low Power Mode).
 - Harga relatif murah untuk fitur yang ditawarkan.
-



BAB IV

Praktikum Dasar GPIO

4.1 Pengantar

GPIO (*General Purpose Input Output*) pada ESP32 digunakan untuk mengendalikan perangkat eksternal (sebagai output) maupun membaca sinyal dari luar (sebagai input). Pada praktikum dasar ini, kita akan mencoba beberapa percobaan sederhana menggunakan GPIO ESP32, yaitu:

1. Tombol (*push button*).
2. Buzzer.
3. Simulasi traffic light.
4. Input Push Button

Praktikum ini bertujuan agar mahasiswa memahami cara kerja input-output digital pada ESP32 sebelum melanjutkan ke penggunaan sensor maupun aktuator yang lebih kompleks.

4.2 Tujuan

Setelah mengikuti praktikum ini, mahasiswa diharapkan dapat:

1. Memahami konsep dasar GPIO (*General Purpose Input Output*) pada ESP32)
2. Mampu mengendalikan perangkat output (buzzer, LED traffic light).
3. Mampu membaca input dari push button.
4. Menerapkan kombinasi input-output sederhana.

4.3 Kontrol Buzzer

➤ Latar Belakang

Buzzer adalah komponen elektronika yang menghasilkan bunyi saat diberikan tegangan. Ada dua jenis: **buzzer aktif** (langsung bunyi saat diberi tegangan) dan **buzzer pasif** (harus diberi

sinyal frekuensi PWM agar berbunyi). Pada praktikum ini digunakan buzzer 5V yang langsung dikontrol dari pin output ESP32.

➤ Tujuan

- Mahasiswa memahami cara mengontrol buzzer dengan GPIO ESP32.
- Mahasiswa dapat memprogram buzzer untuk berbunyi sesuai logika input.

➤ Alat dan Bahan

- ESP32 DevKit V1 (Sudah terpasang dalam box modul mikrokontroler)
- Kabel jumper
- Laptop yang terdapat aplikasi Arduino IDE
- Kabel USB
- Buzzer 5V (Sudah terpasang dalam box modul mikrokontroler)

➤ Prosedur

a. Penyusunan Rangkaian

- Hubungkan kabel jumper dari salah satu pin input pada kotak modul buzzer ke pin D25 (GPIO 25) ESP32 pada baris S (lihat Gambar 4).



Gambar 4

b. Persiapan Arduino IDE

- Buka Arduino IDE di laptop.
- Pastikan Board ESP32 Dev Module sudah dipilih.
- Pilih port COM sesuai ESP32 yang terhubung.
- Gunakan fungsi `digitalWrite` atau `tone()` (jika buzzer aktif).

c. Program ESP32

- Salin program yang ditunjukkan pada Gambar 5 ke ESP32 menggunakan kabel Micro USB.

```
#define BUZZER 25

void setup() {
    pinMode(BUZZER, OUTPUT);
}

void loop() {
    digitalWrite(BUZZER, HIGH); // buzzer ON
    delay(1000);
    digitalWrite(BUZZER, LOW); // buzzer OFF
    delay(1000);
}
```

Gambar 5

- Hubungkan Kabel USB dari Laptop ke ESP32 dan Upload Program (Sesuaikan dengan port masing masing Laptop)
- Tunggu hingga proses upload selesai dan pastikan tidak ada error pada Arduino IDE.
- Amati perilaku buzzer: buzzer akan menyala dan mati secara bergantian sesuai program.

➤ Tugas

Modifikasi program agar buzzer berbunyi dengan pola SOS kode morse (titik–titik–titik, garis–garis–garis, titik–titik–titik).

4.4 Kontrol Lampu Lalu Lintas (*Traffic Light*) Sederhana

➤ Latar Belakang

Lampu lalu lintas merupakan contoh penerapan logika GPIO output. ESP32 dapat digunakan untuk mengontrol beberapa LED yang menyimulasikan sistem lalu lintas sederhana.. Traffic light adalah sistem lampu LED dengan tiga warna utama:

- Merah → berhenti
- Kuning → hati-hati/transisi
- Hijau → jalan

Dalam mikrokontroler, LED disusun dalam rangkaian sederhana dan dikontrol menggunakan pin GPIO. Dengan kombinasi waktu yang berbeda, sistem ini dapat mensimulasikan lampu lalu lintas nyata.

➤ Tujuan

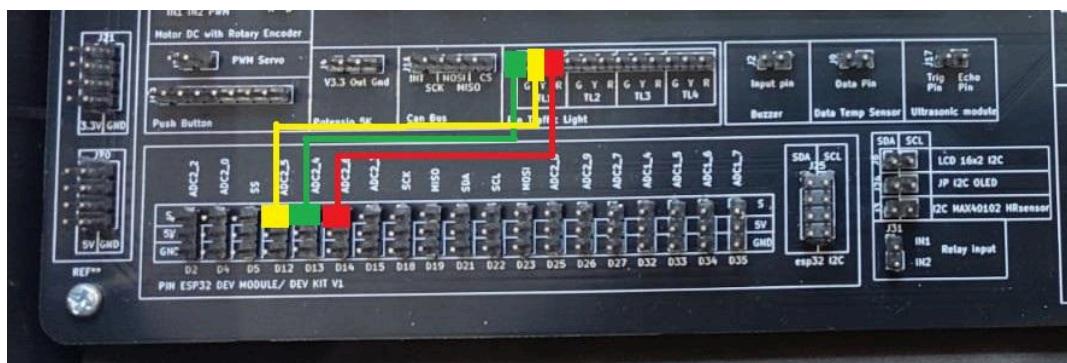
- Mahasiswa memahami kontrol multiple GPIO output.
- Mahasiswa dapat membuat simulasi sederhana lampu lalu lintas.

➤ Alat dan Bahan

- ESP32 DevKit V1 (Sudah terpasang dalam box modul mikrokontroler)
- Kabel jumper
- Laptop yang terdapat aplikasi Arduino IDE
- Kabel USB
- Modul *Traffic Light* (Sudah terpasang dalam box modul mikrokontroler)

➤ Prosedur

a. Penyusunan Rangkaian



Gambar 6

- Hubungkan Pin "G" di kotak modul Traffic Light dengan PinOut D13 di ESP32 pada baris S sesuai gambar 6
- Hubungkan Pin "Y" di kotak modul Traffic Light dengan PinOut D12 di ESP32 pada baris S sesuai gambar 6
- Hubungkan Pin "R" di kotak modul Traffic Light dengan PinOut D14 di ESP32 pada baris S sesuai gambar 6

b. Persiapan Arduino IDE

- Buka Arduino IDE di laptop.
- Pastikan Board ESP32 Dev Module sudah dipilih.
- Pilih port COM sesuai ESP32 yang terhubung.
- Gunakan `digitalWrite` untuk mengatur ON/OFF LED.

c. Program ESP32

- Salin program yang ditunjukkan pada Gambar 7 ke ESP32 menggunakan kabel Micro USB.

```
#define RED 14
#define YELLOW 12
#define GREEN 13

void setup() {
    pinMode(RED, OUTPUT);
    pinMode(YELLOW, OUTPUT);
    pinMode(GREEN, OUTPUT);
}
```



```

void loop() {
    digitalWrite(RED, HIGH);      // Merah nyala
    delay(2000);
    digitalWrite(RED, LOW);

    digitalWrite(GREEN, HIGH);   // Hijau nyala
    delay(2000);
    digitalWrite(GREEN, LOW);

    digitalWrite(YELLOW, HIGH); // Kuning nyala
    delay(1000);
    digitalWrite(YELLOW, LOW);
}

```

Gambar 7

- Hubungkan Kabel USB dari Laptop ke ESP32 dan Upload Program (Sesuaikan dengan port masing masing Laptop)
- Tunggu hingga proses upload selesai dan pastikan tidak ada error pada Arduino IDE.
- Amati traffic light: traffic light akan menyala dan mati secara bergantian sesuai program.

➤ Tugas

Buat program 4 arah traffic light dengan kondisi:

Traffic light 1 → Hijau menyala.

Traffic light 2, 3, 4 → Merah menyala.

Semua arah bergantian sesuai logika lalu lintas nyata.

4.5 Input Push Button

➤ Latar Belakang

Push button adalah salah satu komponen input digital sederhana yang banyak digunakan dalam sistem mikrokontroler. Fungsinya sebagai saklar yang dapat menghubungkan atau memutuskan arus listrik saat ditekan.

Pada ESP32, push button biasanya dihubungkan ke salah satu pin GPIO dengan konfigurasi pull-up atau pull-down resistor:

- Pull-up → Saat tombol tidak ditekan, input bernilai HIGH (1), ketika ditekan bernilai LOW (0).
- Pull-down → Saat tombol tidak ditekan, input bernilai LOW (0), ketika ditekan bernilai HIGH (1).

Dalam praktikum ini digunakan internal pull-up ESP32 sehingga wiring lebih sederhana.

➤ Tujuan

- Mahasiswa memahami prinsip kerja input digital menggunakan push button.
- Mahasiswa dapat membaca input push button pada ESP32.
- Mahasiswa dapat memprogram logika dasar nyala/mati LED interna; ESP32 menggunakan push button.

➤ Alat dan Bahan

- ESP32 DevKit V1 (Sudah terpasang dalam box modul mikrokontroler)
- Kabel jumper
- Laptop yang terdapat aplikasi Arduino IDE
- Kabel USB
- *Push Button* (Sudah terpasang dalam box modul mikrokontroler)

➤ Prosedur

a. Penyusunan Rangkaian

- Hubungkan Salah Satu Pin Push Button di kotak modul Push Button dengan Pin D14 di ESP32 di baris S sesuai gambar 8.



Gambar 8

b. Persiapan Arduino IDE

- Buka Arduino IDE di laptop.
- Pastikan Board ESP32 Dev Module sudah dipilih.
- Pilih port COM sesuai ESP32 yang terhubung.
- Gunakan fungsi `pinMode(pin, INPUT_PULLUP)` untuk push button.

c. Pemrograman ESP32

- Salin program yang ditunjukkan pada Gambar 9 ke ESP32 menggunakan kabel Micro USB.

```
#define BUTTON_PIN 14
#define LED_PIN 2 // LED internal ESP32 (bisa diganti LED eksternal)

void setup() {
    pinMode(BUTTON_PIN, INPUT_PULLUP);
    pinMode(LED_PIN, OUTPUT);
}

void loop() {
    int buttonState = digitalRead(BUTTON_PIN);

    if (buttonState == LOW) {
        digitalWrite(LED_PIN, HIGH); // LED nyala kalau tombol ditekan
    } else {
        digitalWrite(LED_PIN, LOW); // LED mati kalau tombol dilepas
    }
}
```

Gambar 9

- Hubungkan Kabel USB dari Laptop ke ESP32 dan Upload Program (Sesuaikan dengan port masing masing Laptop)
- Tunggu hingga proses upload selesai dan pastikan tidak ada error pada Arduino IDE.
- Amati LED pada ESP32 akan menyala saat *push button* ditekan dan mati saat *push button* dilepas sesuai program.

➤ Tugas

Buat Program dengan menggunakan 2 *push button* yang sebagai on off untuk menyalakan dan mematikan LED pada ESP32



BAB V

Praktikum Dasar Sensor

5.1 Pengantar

Sensor merupakan komponen penting dalam sistem berbasis mikrokontroler karena berfungsi sebagai alat untuk mendeteksi atau mengukur besaran fisik dari lingkungan, seperti jarak, suhu, maupun detak jantung, lalu mengubahnya menjadi sinyal listrik yang dapat diproses oleh ESP32.

Pada praktikum sensor ini, mahasiswa akan mempelajari cara menghubungkan sensor ke ESP32, membaca data yang dihasilkan, serta menampilkannya melalui Serial Monitor atau display. Sensor yang digunakan mencakup sensor jarak (Ultrasonic HC-SR04), sensor suhu (DS18B20 waterproof), sensor detak jantung (MAX30102), serta input analog seperti potensiometer dan rotary encoder.

Dengan memahami penggunaan sensor, mahasiswa akan lebih siap untuk mengembangkan sistem monitoring dan kontrol yang kompleks pada proyek-proyek selanjutnya.

5.2 Tujuan

Setelah mengikuti praktikum ini, mahasiswa diharapkan dapat:

1. Memahami prinsip dasar penggunaan sensor dengan ESP32.
 2. Mampu membaca data sensor melalui pin digital/analog/I2C.
 3. Menampilkan data sensor pada Serial Monitor untuk analisis.
-

5.3 Penggunaan Sensor Ultrasonic

➤ Latar Belakang

Sensor ultrasonic digunakan untuk mengukur jarak berdasarkan pantulan gelombang suara ultrasonik. Prinsip kerjanya dengan mengirimkan sinyal (trigger) dan menerima pantulan (echo).. Jarak dihitung dengan persamaan:

$$\text{Jarak (cm)} = \frac{\text{waktu} \times 0.034}{2}$$

➤ Tujuan

- Mahasiswa memahami prinsip kerja sensor ultrasonic.
 - Mahasiswa dapat membaca hasil pengukuran jarak pada Serial Monitor.

➤ Alat dan Bahan

- ESP32 DevKit V1 (Sudah terpasang dalam box modul mikrokontroler)
 - Kabel jumper
 - Laptop yang terdapat aplikasi Arduino IDE
 - Kabel USB
 - Sensor *Ultrasonic* (Sudah terpasang dalam box modul mikrokontroler)

➤ Prosedur

a. Penyusunan Rangkaian



Gambar 10

- Hubungkan Pin Trig pada kotak pin Ultrasonic ke Pin D5 di ESP32 di baris S Sesuai Gambar 10
 - Hubungkan Pin Echo pada kotak pin Ultrasonic ke Pin D18 di ESP32 di baris S Sesuai Gambar 10

b. Persiapan Arduino IDE

- Buka Arduino IDE di laptop.
 - Pastikan Board ESP32 Dev Module sudah dipilih.
 - Pilih port COM sesuai ESP32 yang terhubung.

c. Pemrograman ESP32

- Salin program yang ditunjukkan pada Gambar 11 ke ESP32 menggunakan kabel Micro USB.

```
#define TRIG 5
#define ECHO 18

long duration;
int distance;

void setup() {
    Serial.begin(115200);
    pinMode(TRIG, OUTPUT);
    pinMode(ECHO, INPUT);
}

void loop() {
    digitalWrite(TRIG, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG, LOW);

    duration = pulseIn(ECHO, HIGH);
    distance = duration * 0.034 / 2;

    Serial.print("Jarak: ");
    Serial.print(distance);
    Serial.println(" cm");
    delay(500);
}
```



Gambar 11

- Hubungkan Kabel USB dari Laptop ke ESP32 dan Upload Program (Sesuaikan dengan port masing masing Laptop)

- Tunggu hingga proses upload selesai dan pastikan tidak ada error pada Arduino IDE.
- Buka Serial Monitor pada pojok kanan arduino IDE dan ganti baudrate 115200
- Amati perubahan nilainya dengan meletakkan barang di depan sensor ultrasonic dan ubah jaraknya.

➤ **Tugas**

Buat program untuk mendeteksi jarak tertentu dengan indikator servo, jika pada jarak <3 cm maka servo akan berputar ke kanan dan ke kiri (seperti membuka menutup).

5.4 Interface Sensor *MAX30102 Heart Rate Saturation Sensor*

➤ **Latar Belakang**

MAX30102 adalah sensor optik yang digunakan untuk mengukur detak jantung dan kadar oksigen dalam darah (SpO_2) menggunakan cahaya merah dan inframerah.

➤ **Tujuan**

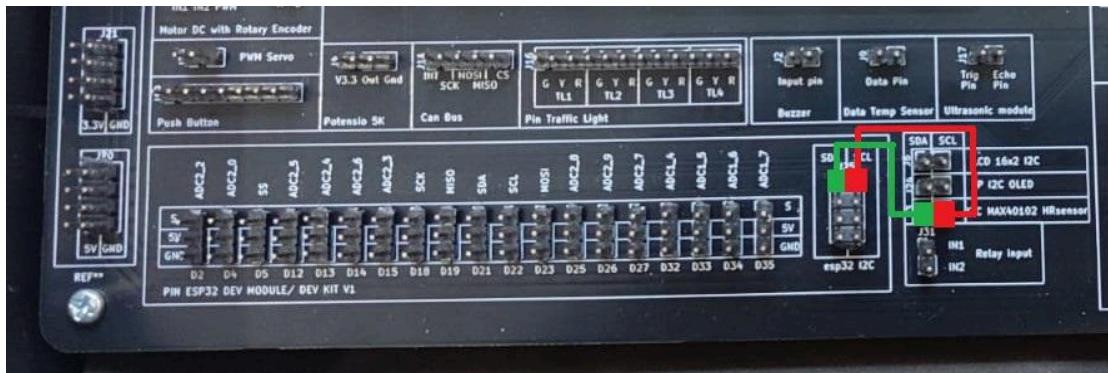
- Mahasiswa memahami penggunaan sensor biomedis berbasis optik.
- Mahasiswa dapat menampilkan hasil heart rate di Serial Monitor.

➤ **Alat dan Bahan**

- ESP32 DevKit V1 (Sudah terpasang dalam box modul mikrokontroler)
- Kabel jumper
- Laptop yang terdapat aplikasi Arduino IDE
- Kabel USB
- Sensor MAX30102 (Sudah terpasang dalam box modul mikrokontroler)

➤ **Prosedur**

- a. Penyusunan Rangkaian



Gambar 12

- Hubungkan pin SDA MAX30102 pada pin SDA ESP32 dengan kabel jumper Sesuai Gambar 12
 - Hubungkan pin SCL MAX30102 pada pin SCL ESP32 dengan kabel jumper Sesuai Gambar 12
- b. Persiapan Arduino IDE
- Buka Arduino IDE di Laptop
 - Install library "MAX3010x" dari Arduino Library Manager.
 - Pastikan Board ESP32 Dev Module sudah dipilih.
 - Pilih port COM sesuai ESP32 yang terhubung.
- c. Pemrograman ESP32
- Salin program yang ditunjukkan pada Gambar 13 ke ESP32 menggunakan kabel Micro USB.

```

#include <Wire.h>
#include "MAX30105.h"
#include "heartRate.h"

MAX30105 particleSensor;

void setup() {
    Serial.begin(115200);
    if (!particleSensor.begin(Wire, I2C_SPEED_STANDARD)) {
        Serial.println("MAX30102 tidak terdeteksi. Periksa wiring!");
        while (1);
    }
    particleSensor.setup();
}

void loop() {
    long irValue = particleSensor.getIR();
    if (checkForBeat(irValue) == true) {
        long delta = millis();
        static long lastBeat = 0;
        delta = millis() - lastBeat;
        lastBeat = millis();

        int bpm = 60 / (delta / 1000.0);
        Serial.print("Detak Jantung: ");
        Serial.println(bpm);
    }
}

```

Gambar 13

- Hubungkan Kabel USB dari Laptop ke ESP32 dan Upload Program (Sesuaikan dengan port masing masing Laptop)
- Tunggu hingga proses upload selesai dan pastikan tidak ada error pada Arduino IDE.
- Buka Serial Monitor pada pojok kanan arduino IDE dan ganti baudrate 115200
- Amati perubahan nilainya dengan menempelkan jari telunjuk ke tengah tengah sensor MAX30102 .

➤ Tugas

Tampilkan status: “Detak Jantung Terdeteksi” jika IR > 50000, sebaliknya tampilkan “Tidak Ada Jari

5.5 Interface sensor suhu water proof DS18B20

➤ Latar Belakang

DS18B20 adalah sensor suhu digital berbasis protokol 1-Wire. Sensor ini mampu membaca suhu dengan akurasi tinggi dan dapat digunakan dalam aplikasi monitoring suhu. Tahan air sehingga cocok untuk aplikasi biomedis atau lingkungan.

➤ Tujuan

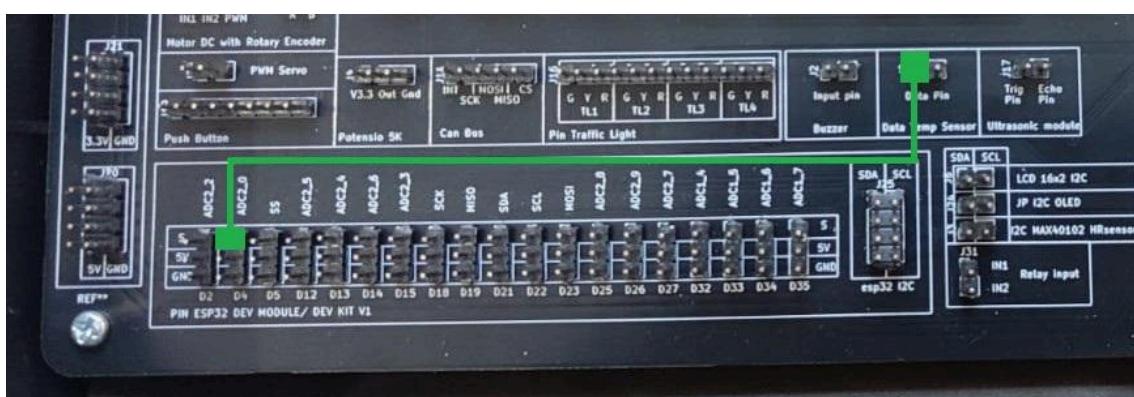
- Mahasiswa memahami penggunaan sensor digital 1-Wire.
- Mahasiswa dapat membaca data suhu melalui Serial Monitor.

➤ Alat dan Bahan

- ESP32 DevKit V1 (Sudah terpasang dalam box modul mikrokontroler)
- Kabel jumper
- Laptop yang terdapat aplikasi Arduino IDE
- Kabel USB
- Sensor DS18B20 (Sudah terpasang dalam box modul mikrokontroler)
- Resistor 4.7KΩ (Sudah terpasang dalam box modul mikrokontroler)
- Gelas kecil berisi Air hangat / Air Dingin

➤ Prosedur

- a. Penyusunan Rangkaian



Gambar 14

- Hubungkan white housing pada temperature sensor ke modul temperature sensor.
- Hubungkan data pin pada kotak data temp sensor ke pinout esp32.

Pin data → D4 di baris S (Gambar 14).

- Siapkan pemantik api/air dingin
- b. Persiapan Arduino IDE
- Buka Arduino IDE di Laptop
 - Install OneWire dan DallasTemperature dari Arduino Library Manager.
 - Pastikan Board ESP32 Dev Module sudah dipilih.
 - Pilih port COM sesuai ESP32 yang terhubung.
- c. Pemrograman ESP32
- Salin program yang ditunjukkan pada Gambar 15 ke ESP32 menggunakan kabel Micro USB.

```
#include <OneWire.h>
#include <DallasTemperature.h>

#define ONE_WIRE_BUS 4

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

void setup() {
    Serial.begin(115200);
    sensors.begin();
}

void loop() {
    sensors.requestTemperatures();
    float suhu = sensors.getTempCByIndex(0);
    Serial.print("Suhu: ");
    Serial.print(suhu);
    Serial.println(" °C");
    delay(1000);
}
```

Gambar 15

- Hubungkan Kabel USB dari Laptop ke ESP32 dan Upload Program (Sesuaikan dengan port masing masing Laptop)
- Tunggu hingga proses upload selesai dan pastikan tidak ada error pada Arduino IDE.
- Buka Serial Monitor pada pojok kanan arduino IDE dan ganti baudrate 115200
- Amati perubahan nilainya dengan memasukkan sensor temperature ke dalam air dingin/pemantik api dan di luar air dingin/pemantik api.

➤ **Tugas**

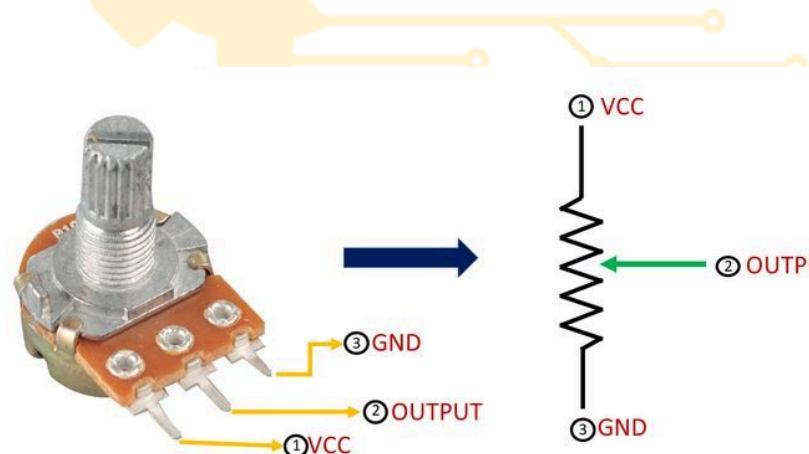
Buat program untuk membaca suhu dari sensor dan buzzer sebagai indikator. Atur buzzer agar berbunyi jika suhu melebihi batas tertentu (misalnya $>30^{\circ}\text{C}$) dan mati jika suhu normal.

5.6 Input Potensiometer

➤ **Latar Belakang**

Potensiometer adalah sensor analog berupa resistor variabel yang digunakan untuk menghasilkan perubahan tegangan berdasarkan posisi putaran porosnya.

- Potensiometer 5K memiliki tiga pin:



Gambar 16

1. VCC (3.3V dari ESP32)
 2. Pin tengah → output analog yang dihubungkan ke pin ADC ESP32 (misalnya GPIO 34).
-

3. GND

- ESP32 memiliki ADC (Analog to Digital Converter) 12-bit, sehingga nilai input analog akan terbaca antara 0 – 4095.

Contoh pemanfaatan: kontrol volume, pengaturan kecepatan motor, atau input user interface.

➤ Tujuan

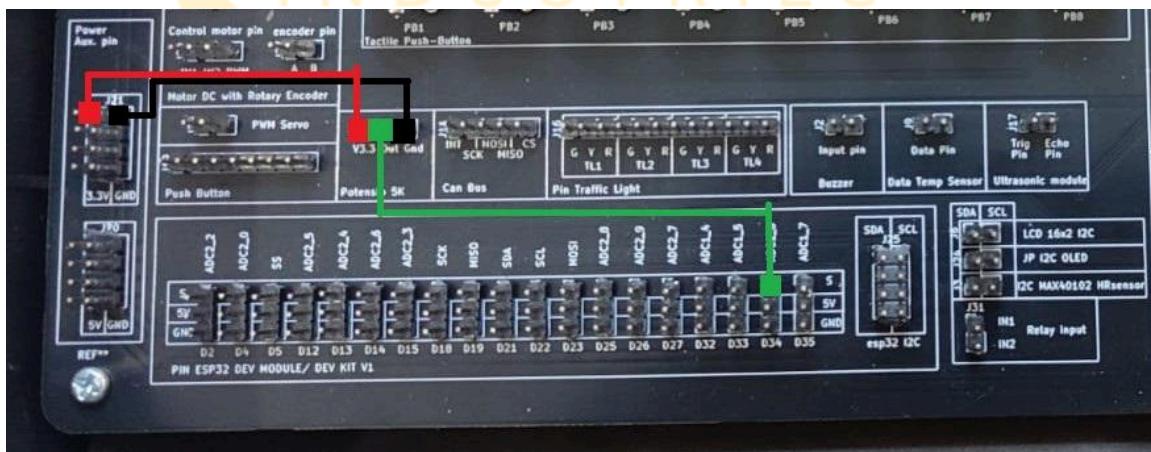
- Mahasiswa memahami pembacaan sinyal analog pada ESP32.
- Mahasiswa dapat menggunakan nilai potensiometer untuk mengatur parameter tertentu.

➤ Alat dan Bahan

- ESP32 DevKit V1 (Sudah terpasang dalam box modul mikrokontroler)
- Kabel jumper
- Laptop yang terdapat aplikasi Arduino IDE
- Kabel USB
- Potensiometer 5k (Sudah terpasang dalam box modul mikrokontroler)

➤ Prosedur

a. Penyusunan Rangkaian



Gambar 17

- Hubungkan VCC pada kotak Potensiometer ke 3.3V pada ESP32 (Gambar 17)
- Hubungkan GND pada kotak potensiometer ke GND pada ESP32 (Gambar 17)
- Hubungkan pin data pada kotak potensiometer ke pin D34 pada ESP32 di baris S (Gambar 17)

b. Persiapan Arduino IDE

- Buka Arduino IDE di Laptop
- Pastikan Board ESP32 Dev Module sudah dipilih.
- Pilih port COM sesuai ESP32 yang terhubung.

c. Pemrograman ESP32

- Salin program yang ditunjukkan pada Gambar 18 ke ESP32 menggunakan kabel Micro USB.

```
#define POT_PIN 34

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

void loop() {
    int potValue = analogRead(POT_PIN);
    Serial.print("Nilai Potensiometer: ");
    Serial.println(potValue);
    delay(200);
}
```



Gambar 18

- Hubungkan Kabel USB dari Laptop ke ESP32 dan Upload Program (Sesuaikan dengan port masing masing Laptop)
- Tunggu hingga proses upload selesai dan pastikan tidak ada error pada Arduino IDE.
- Buka Serial Monitor pada pojok kanan arduino IDE dan ganti baudrate 115200
- Amati perubahan nilainya dengan memutar ke potensiometer ke kanan dan ke kiri.

➤ Tugas

Buat program untuk mengontrol servo dengan potensiometer, serta membaca nilai potensiometer dan konversikan ke sudut servo ($0^\circ - 180^\circ$). Putar potensiometer perlahan dan amati perubahan sudut servo sesuai nilai potensiometer.

5.7 Input Rotary Encoder

➤ Latar Belakang

Rotary encoder digunakan untuk membaca arah putaran dan jumlah langkah rotasi. Encoder ini menghasilkan dua sinyal digital (A dan B) yang dibaca secara kuadratur.

➤ Tujuan

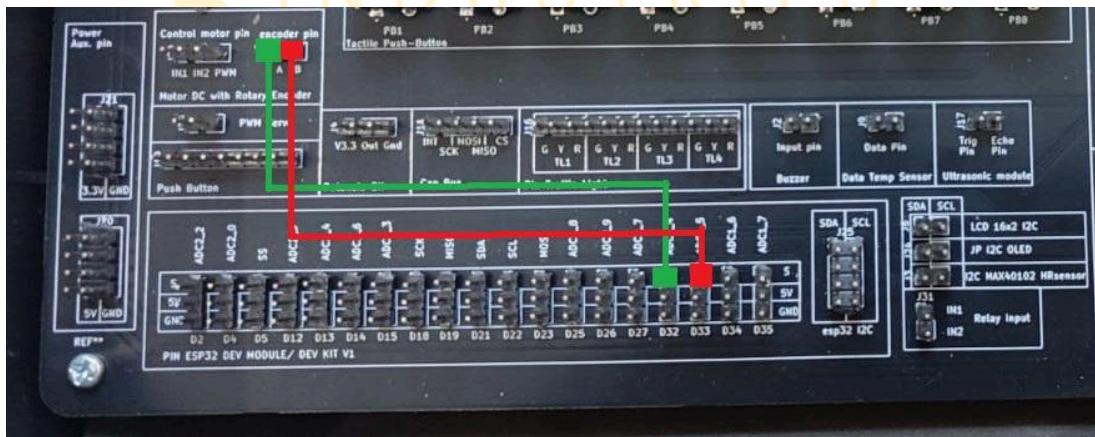
- Mahasiswa memahami prinsip kerja rotary encoder.
- Mahasiswa dapat membaca arah dan jumlah putaran.

➤ Alat dan Bahan

- ESP32 DevKit V1 (Sudah terpasang dalam box modul mikrokontroler)
- Kabel jumper
- Laptop yang terdapat aplikasi Arduino IDE
- Kabel USB
- Rotary Encoder (Sudah terpasang dalam box modul mikrokontroler)

➤ Prosedur

a. Penyusunan Rangkaian



Gambar 19

- Hubungkan Pin “A” pada kotak pin rotary encoder ke Pin D32 di ESP32 di baris S Sesuai Gambar 19
- Hubungkan Pin “B” pada kotak pin rotary encoder ke Pin D33 di ESP32 di baris S Sesuai Gambar 19

b. Persiapan Arduino IDE

- Buka Arduino IDE di Laptop
 - Pastikan Board ESP32 Dev Module sudah dipilih.
 - Pilih port COM sesuai ESP32 yang terhubung.
- c. Pemrograman ESP32
- Salin program yang ditunjukkan pada Gambar 20 ke ESP32 menggunakan kabel Micro USB.

```
#define B 33
#define A 32

int counter = 0;
int lastStateB;

void setup() {
    pinMode(B, INPUT);
    pinMode(A, INPUT);
    Serial.begin(115200);
    lastStateB = digitalRead(B);
}

void loop() {
    int currentStateB = digitalRead(B);
    if (currentStateB != lastStateB && currentStateB == 1) {
        if (digitalRead(A) != currentStateB) {
            counter++; // tadinya counter--
            Serial.println("Putar Kanan");
        } else {
            counter--; // tadinya counter++
            Serial.println("Putar Kiri");
        }
        Serial.print("Counter: ");
        Serial.println(counter);
    }
    lastStateB = currentStateB;
}
```

Gambar 20

- Hubungkan Kabel USB dari Laptop ke ESP32 dan Upload Program (Sesuaikan dengan port masing masing Laptop)
- Tunggu hingga proses upload selesai dan pastikan tidak ada error pada Arduino IDE.
- Buka Serial Monitor pada pojok kanan arduino IDE dan ganti baudrate 115200

- Amati perubahan nilainya dengan memutar encoder pada motor DC ke kanan dan ke kiri.

➤ Tugas

1. Buat program untuk membaca arah putaran (kanan/kiri) dan tampilkan di Serial Monitor.
 2. Tambahkan counter yang bertambah saat diputar kanan dan berkurang saat diputar kiri
-



BAB VI

Praktikum Aktuator

6.1 Pengantar

Aktuator merupakan perangkat yang berfungsi untuk mengubah sinyal listrik dari mikrokontroler menjadi aksi fisik. Jika sensor berperan sebagai input untuk membaca kondisi lingkungan, maka aktuator adalah output yang mengeksekusi perintah dari sistem.

Dalam sistem berbasis ESP32, aktuator digunakan untuk menggerakkan atau mengendalikan perangkat eksternal, misalnya menyalakan lampu, menggerakkan motor, mengaktifkan relay, atau mengontrol posisi servo. Dengan memanfaatkan aktuator, ESP32 dapat berinteraksi secara nyata dengan lingkungan sekitarnya.

Pada praktikum ini, mahasiswa akan mencoba beberapa aktuator dasar yang umum digunakan dalam sistem embedded dan IoT, yaitu:

- Relay Module → untuk mengontrol perangkat dengan tegangan lebih tinggi.
- Servo Motor → untuk mengatur sudut posisi tertentu.
- Motor DC dengan Driver → untuk mengatur kecepatan dan arah putaran motor.

6.2 Tujuan

- Memahami prinsip kerja aktuator yang digunakan pada ESP32.
 - Mampu mengendalikan relay, motor DC, dan servo melalui program ESP32.
 - Mengetahui cara menghasilkan sinyal PWM untuk mengatur kecepatan motor atau sudut servo.
 - Mengintegrasikan buzzer sebagai alat peringatan atau indikator.
 - Mengembangkan dasar sistem kontrol berbasis aktuator untuk aplikasi yang lebih kompleks.
-

6.3 Kontrol Relay Module

➤ Latar Belakang

Relay adalah sebuah saklar elektrik yang dikendalikan oleh arus listrik, berfungsi untuk memutus atau menyambungkan arus listrik ke beban. Pada modul relay yang umum digunakan bersama mikrokontroler (seperti ESP32), terdapat kumparan elektromagnetik dan kontak mekanis yang berpindah ketika kumparan dialiri arus.

Relay sangat berguna ketika mikrokontroler hanya mampu mengeluarkan tegangan dan arus kecil (misalnya 3,3V dengan arus puluhan mA), sedangkan beban yang dikendalikan seperti motor DC, lampu, atau peralatan rumah tangga memerlukan tegangan dan arus jauh lebih besar (5V, 12V, bahkan 220V).

Bagian Utama Relay:

1. Kumparan (Coil)

Bagian ini dikendalikan oleh sinyal dari ESP32. Saat diberi tegangan, coil akan menciptakan medan magnet yang menarik tuas mekanis.

2. Kontak NO (Normally Open)

- Dalam keadaan default (relay mati), kontak ini terbuka (tidak terhubung).
- Saat relay aktif, kontak NO akan tertutup dan menghubungkan arus ke beban.
- Biasanya dipakai untuk menghidupkan motor hanya ketika relay diaktifkan.

3. Kontak NC (Normally Closed)

- Dalam keadaan default, kontak ini tertutup (arus mengalir).
- Saat relay aktif, kontak ini akan terbuka (arus terputus).
- Cocok digunakan jika beban (misalnya motor) ingin dalam keadaan nyala terus, lalu mati hanya ketika relay aktif.

4. Common (COM)

Titik tengah kontak yang dihubungkan ke beban (motor, lampu, dsb).

Prinsip Kerja Relay Motor:

- ESP32 memberi sinyal logika HIGH ke input relay → coil aktif → kontak berpindah.
- Jika beban motor dihubungkan ke NO → motor hanya menyala ketika relay aktif.
- Jika beban motor dihubungkan ke NC → motor akan selalu menyala, dan hanya mati ketika relay aktif.

Manfaat Relay untuk Motor:

- Mengisolasi rangkaian mikrokontroler (3,3V) dengan rangkaian motor (5V/12V).
- Mengontrol motor dengan aman tanpa membebani langsung pin GPIO ESP32.
- Memberi fleksibilitas penggunaan NO atau NC sesuai kebutuhan aplikasi.

➤ Tujuan

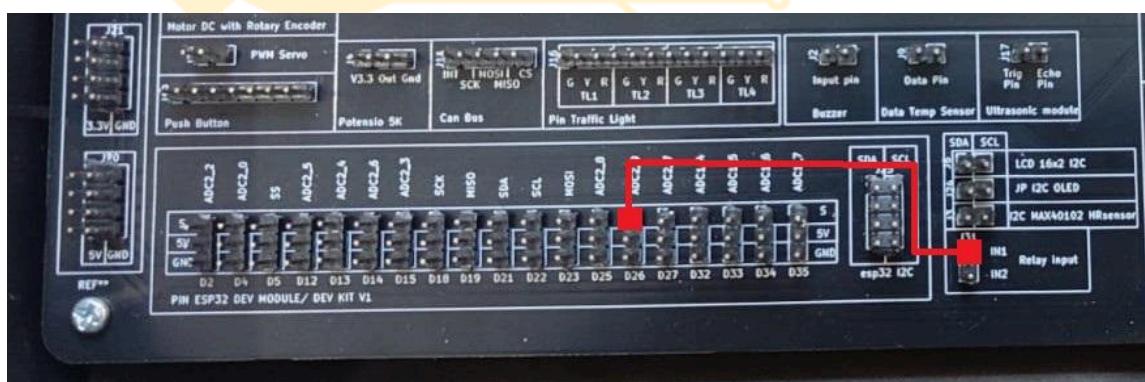
- Memahami prinsip kerja relay dan fungsinya sebagai aktuator.
- Menghubungkan modul relay 2-channel ke ESP32 dengan aman.
- Mengendalikan dua relay secara digital (ON/OFF) menggunakan ESP32.
- Mengimplementasikan kontrol lewat perintah Serial.

➤ Alat dan Bahan

- ESP32 DevKit V1 (Sudah terpasang dalam box modul mikrokontroler)
- Kabel jumper
- Laptop yang terdapat aplikasi Arduino IDE
- Kabel USB
- Relay 3.3v 2 channel (Sudah terpasang dalam box modul mikrokontroler)

➤ Prosedur

a. Penyusunan Rangkaian



Gambar 21

- Hubungkan pin “int” pada kotak relay modul ke Pin D26 di ESP32 di baris S Sesuai Gambar 21

b. Persiapan Arduino IDE

- Buka Arduino IDE di Laptop

- Pastikan Board ESP32 Dev Module sudah dipilih.
 - Pilih port COM sesuai ESP32 yang terhubung.
- c. Pemrograman ESP32
- Salin program yang ditunjukkan pada Gambar 22 ke ESP32 menggunakan kabel Micro USB.

```
#define RELAY_PIN 26

void setup() {
    pinMode(RELAY_PIN, OUTPUT);
}

void loop() {
    digitalWrite(RELAY_PIN, HIGH); // Relay ON
    delay(2000);
    digitalWrite(RELAY_PIN, LOW); // Relay OFF
    delay(2000);
}
```

Gambar 22

- Hubungkan Kabel USB dari Laptop ke ESP32 dan Upload Program (Sesuaikan dengan port masing masing Laptop)
- Tunggu hingga proses upload selesai dan pastikan tidak ada error pada Arduino IDE.
- Amati perubahan LED pada relay motor sebagai indikatornya yang mana jika LED menyala → relay dalam kondisi aktif (kontak dalam relay berubah posisi → NO jadi terhubung).

➤ Tugas

Buat program untuk menghidupkan 2 relay selama 2 detik lalu mematikannya 2 detik secara berulang.

6.4 Control Servo Emax

➤ Latar Belakang

Servo motor adalah aktuator yang bergerak pada sudut tertentu (0° – 180°) sesuai sinyal PWM. Banyak digunakan untuk robotika, sistem kontrol, dan aplikasi biomedis.

➤ Tujuan

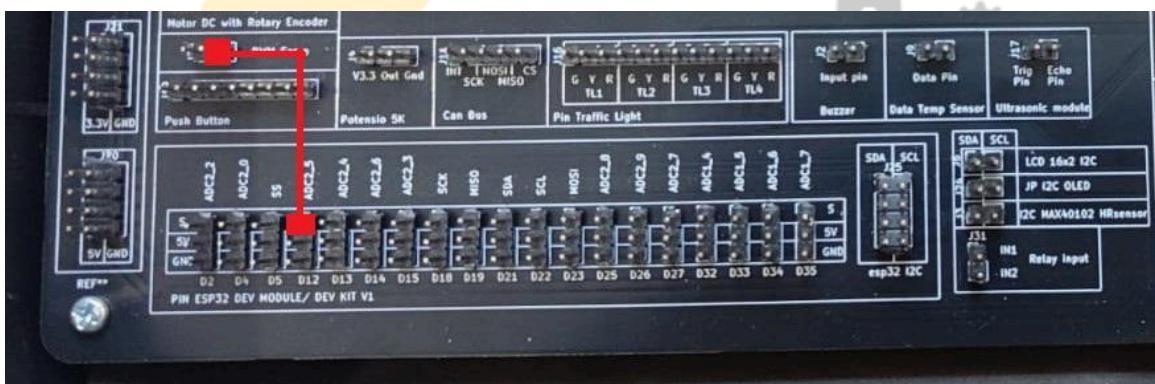
- Mahasiswa memahami prinsip kerja servo motor.
- Mahasiswa dapat mengatur sudut servo menggunakan ESP32

➤ Alat dan Bahan

- ESP32 DevKit V1 (Sudah terpasang dalam box modul mikrokontroler)
- Kabel jumper
- Laptop yang terdapat aplikasi Arduino IDE
- Kabel USB
- Servo EMAX (Sudah terpasang dalam box modul mikrokontroler)

➤ Prosedur

a. Penyusunan Rangkaian



- Hubungkan salah satu pin pwm pada kotak servo ke pin D12 pada kotak esp32 di baris S (Gambar 22)

b. Persiapan Arduino IDE

- Buka Arduino IDE di Laptop
- Pastikan Board ESP32 Dev Module sudah dipilih.
- Pilih port COM sesuai ESP32 yang terhubung.
- Install Library ESP32Servo by kevin harrington

c. Pemrograman ESP32

- Salin program yang ditunjukkan pada Gambar 23 ke ESP32 menggunakan kabel Micro USB.

```

#include <ESP32Servo.h>

Servo myservo;

void setup() {
    myservo.attach(12);
}

void loop() {
    for (int pos = 0; pos <= 180; pos++) {
        myservo.write(pos);
        delay(15);
    }
    for (int pos = 180; pos >= 0; pos--) {
        myservo.write(pos);
        delay(15);
    }
}

```

Gambar 23

- Hubungkan Kabel USB dari Laptop ke ESP32 dan Upload Program (Sesuaikan dengan port masing masing Laptop)
- Tunggu hingga proses upload selesai dan pastikan tidak ada error pada Arduino IDE.
- Amati perubahan arah dari baling baling servo, dan akan berputar 180 derajat ke kanan dan ke kiri.

➤ Tugas

Buat program untuk menggerakkan servo dari $0^\circ \rightarrow 90^\circ \rightarrow 180^\circ$ dengan jeda 1 detik tiap sudut.

6.5 Control Motor DC + Driver

➤ Latar Belakang

Motor DC adalah salah satu aktuator paling umum yang digunakan dalam sistem kendali, robotika, dan otomasi karena dapat menghasilkan gerakan rotasi sederhana dengan kecepatan

yang dapat dikendalikan. Namun, motor DC membutuhkan arus lebih besar dibandingkan kemampuan keluaran pin mikrokontroler seperti ESP32.

Untuk itu, diperlukan driver motor sebagai perantara, salah satunya adalah IC L293DD yang merupakan motor driver berbasis dual H-Bridge yang berfungsi untuk mengendalikan dua motor DC atau satu motor stepper dengan arah maju-mundur dan kecepatan variabel. Versi L293DD merupakan varian SMD dengan kapasitas arus lebih besar dibanding L293D standar, sehingga lebih stabil digunakan untuk motor dengan beban yang lebih tinggi.

Dengan menggunakan ESP32, IC L293DD dapat dikendalikan melalui sinyal digital untuk arah putaran dan PWM untuk kecepatan. Hal ini membuat rangkaian lebih aman karena arus besar dari motor tidak langsung melewati mikrokontroler.

➤ Tujuan

- Memahami prinsip kerja motor DC dan kebutuhan penggunaan driver.
- Menjelaskan fungsi IC L293D sebagai pengendali motor berbasis H-Bridge.
- Melatih keterampilan menyusun rangkaian motor DC dengan ESP32.
- Mengimplementasikan kontrol arah dan kecepatan motor menggunakan PWM.

➤ Alat dan Bahan

- ESP32 DevKit V1 (Sudah terpasang dalam box modul mikrokontroler)
- Kabel jumper
- Laptop yang terdapat aplikasi Arduino IDE
- Kabel USB
- Motor DC (Sudah terpasang dalam box modul mikrokontroler)
- Driver Motor DC L293DD (Sudah terpasang dalam box modul mikrokontroler)

➤ Prosedur

- a. Penyusunan Rangkaian



Gambar 24

- Hubungkan pin PWM pada kotak kontrol motor dc ke pin D12 ESP32 (Gambar 24).
 - Hubungkan pin IN1 pada kotak kontrol motor dc ke pin D4 ESP32 (Gambar 24).
 - Hubungkan pin IN2 pada kotak kontrol motor dc ke pin D4 ESP32 (Gambar 24).
- b. Persiapan Arduino IDE
- Buka Arduino IDE di laptop.
 - Pilih board ESP32 Dev Module pada menu *Tools > Board*.
 - Pilih port COM sesuai ESP32 yang terhubung.
- c. Pemrograman ESP32
- Salin program yang ditunjukkan pada Gambar 25 ke ESP32 menggunakan kabel Micro USB.

```

// use 8 bit precision for motor timer
#define Timer 8

// use 50 Hz as a PWM base frequency
#define FREQ 50

// PWM atau EN pins
#define PWM 12

// motor channel that will be used instead of automatic selection.
#define MOTOR_CHANNEL 0

void setup() {
    // Use single motor channel 0 for both pins
    ledcAttachChannel(PWM, FREQ, Timer, MOTOR_CHANNEL);
    pinMode(4,OUTPUT);
    pinMode(5,OUTPUT);
}

void loop() {
    // set the speed on motor channel 0
    ledcWriteChannel(MOTOR_CHANNEL, 200);
    digitalWrite(4,HIGH);
    digitalWrite(5,LOW);
}

```

Gambar 25

- Hubungkan Kabel USB dari Laptop ke ESP32 dan Upload Program (Sesuaikan dengan port masing masing Laptop)
- Tunggu hingga proses upload selesai dan pastikan tidak ada error pada Arduino IDE.
- Amati kecepatan dari motor DC.

➤ Tugas

Buat program agar motor bergerak maju 2 detik, berhenti 1 detik, lalu mundur 2 detik, berulang terus.

BAB VII

Praktikum Display

7.1 Pengantar

Display merupakan salah satu media output yang digunakan untuk menampilkan informasi dari mikrokontroler kepada pengguna. Pada praktikum ini, mahasiswa akan mempelajari penggunaan dua jenis display yang umum digunakan, yaitu LCD 16x2 dengan modul I2C dan OLED.

Display LCD 16x2 banyak digunakan karena mudah diprogram dan mampu menampilkan karakter alfanumerik sederhana. Sedangkan OLED memiliki resolusi lebih tinggi, kontras lebih baik, serta mendukung tampilan grafik. Dengan menggunakan display, mahasiswa dapat menampilkan data sensor, status aktuator, maupun pesan sistem secara lebih interaktif.

7.2 Tujuan

Setelah mengikuti praktikum ini, mahasiswa diharapkan dapat:

1. Memahami prinsip kerja LCD I2C dan OLED.
2. Menghubungkan dan mengkonfigurasi display dengan ESP32.
3. Menampilkan teks atau data dari sensor ke layar.
4. Membuat tampilan sederhana untuk aplikasi monitoring.

7.3 *Interface LCD 16x2*

➤ Latar Belakang

LCD (Liquid Crystal Display) 16x2 banyak digunakan untuk menampilkan teks (16 kolom × 2 baris). Dengan modul I2C backpack, penggunaan pin lebih efisien (hanya SDA dan SCL).

➤ Tujuan

- Mahasiswa memahami prinsip kerja LCD dengan komunikasi I2C.

- Mahasiswa dapat menampilkan teks ke LCD menggunakan ESP32.

➤ Alat dan Bahan

- ESP32 DevKit V1 (Sudah terpasang dalam box modul mikrokontroler)
- Kabel jumper
- Laptop yang terdapat aplikasi Arduino IDE
- Kabel USB
- LCD I2C 16X2 (Sudah terpasang dalam box modul mikrokontroler)

➤ Prosedur

a. Penyusunan Rangkaian



Gambar 26

- Hubungkan pin SDA pada kotak Oled 128x64 ke pin SDA pada kotak ESP32 (Sesuai Gambar 26)
- Hubungkan pin SCL pada kotak Oled 128x64 ke pin SCL pada kotak ESP32 (Sesuai Gambar 26)
- b. Persiapan Arduino IDE
 - Buka Arduino IDE di Laptop
 - Pastikan Board ESP32 Dev Module sudah dipilih.
 - Pilih port COM sesuai ESP32 yang terhubung.
 - Install library LiquidCrystal_I2C
- c. Pemrograman ESP32
 - Salin program yang ditunjukkan pada Gambar 27 ke ESP32 menggunakan kabel Micro USB.

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2); // Alamat I2C LCD biasanya 0x27 atau 0x3F

void setup() {
    lcd.init();
    lcd.backlight();
    lcd.setCursor(0,0);
    lcd.print("Hello ESP32!");
    lcd.setCursor(0,1);
    lcd.print("LCD 16x2 I2C");
}

void loop() {
}

```

Gambar 27

- Hubungkan Kabel USB dari Laptop ke ESP32 dan Upload Program (Sesuaikan dengan port masing masing Laptop)
- Amati LCD 16X2 apakah muncul tulisan “Hello ESP32 !” dan “LCD 16x2 I2C”

➤ Tugas

Tampilkan teks bergerak dari kiri ke kanan.

7.4 *Interface OLED 2"*

➤ Latar Belakang

OLED (Organic Light Emitting Diode) adalah display dengan kontras tinggi, resolusi kecil (misal 128×64), dan hemat daya. OLED berbasis chip SSD1306 populer digunakan bersama ESP32.

➤ Tujuan

- Mahasiswa memahami penggunaan OLED dengan komunikasi I2C.
- Mahasiswa dapat menampilkan teks dan grafis sederhana pada OLED.

➤ Alat dan Bahan

- ESP32 DevKit V1 (Sudah terpasang dalam box modul mikrokontroler)
- Kabel jumper
- Laptop yang terdapat aplikasi Arduino IDE
- Kabel USB
- OLED 128x64 (Sudah terpasang dalam box modul mikrokontroler)

➤ Prosedur

a. Penyusunan Rangkaian



Gambar 28

- Hubungkan SDA pada kotak Oled 128x64 ke pin SDA pada kotak ESP32 (Sesuai Gambar 28)
- Hubungkan SCL pada kotak Oled 128x64 ke pin SCL pada kotak ESP32 (Sesuai Gambar 28)

b. Persiapan Arduino IDE

- Buka Arduino IDE di Laptop
- Pastikan Board ESP32 Dev Module sudah dipilih.
- Pilih port COM sesuai ESP32 yang terhubung.
- Install library Adafruit SSD1306
- Install library Adafruit GFX

c. Pemrograman ESP32

- Salin program yang ditunjukkan pada Gambar 29 ke ESP32 menggunakan kabel Micro USB.

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

void setup() {
    if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
        for(;;); // Gagal inisialisasi
    }
    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(0,0);
    display.println("Hello ESP32!");
    display.display();
}

void loop() {
```

Gambar 29

- Hubungkan Kabel USB dari Laptop ke ESP32 dan Upload Program (Sesuaikan dengan port masing masing Laptop)
- Amati OLED apakah muncul tulisan “Hello ESP32 !”

➤ Tugas

Buat tampilan grafik sederhana (misalnya garis, lingkaran, atau bar level).

BAB VIII

Praktikum Komunikasi

8.1 Pengantar

Komunikasi data merupakan aspek penting dalam sistem embedded dan IoT karena memungkinkan pertukaran informasi antara mikrokontroler dengan perangkat lain. Pada praktikum ini, mahasiswa akan mempelajari komunikasi menggunakan modul CAN Bus (TJA1050) yang banyak digunakan pada sistem otomotif, robotik, dan aplikasi industri.

CAN Bus memungkinkan banyak perangkat saling terhubung dalam satu jaringan dengan tingkat keandalan tinggi dan toleransi terhadap gangguan. Dengan memanfaatkan modul TJA1050, ESP32 dapat mengirimkan dan menerima data melalui protokol CAN.

8.2 Tujuan

Setelah mengikuti praktikum ini, mahasiswa diharapkan dapat:

1. Memahami konsep dasar komunikasi CAN Bus.
2. Mengetahui cara kerja modul TJA1050 sebagai interface CAN.
3. Mampu melakukan inisialisasi komunikasi CAN pada ESP32.
4. Mengirim data sederhana menggunakan protokol CAN.

8.3 TJA1050 (CAN Bus)

➤ Latar Belakang

CAN (Controller Area Network) merupakan salah satu protokol komunikasi serial yang banyak digunakan di dunia otomotif, industri, dan perangkat embedded karena keandalannya dalam mengirimkan data antar perangkat dengan kecepatan tinggi dan toleransi terhadap noise. ESP32 sendiri tidak memiliki transceiver CAN secara langsung, sehingga diperlukan modul tambahan TJA1050.

TJA1050 adalah CAN transceiver yang berfungsi sebagai penghubung antara mikrokontroler (dengan sinyal logika TTL) dan jalur komunikasi CAN (CANH dan CANL). Modul ini bekerja sebagai antarmuka fisik yang memastikan data dari ESP32 dapat dikirimkan dalam format tegangan diferensial (CAN bus), sekaligus menerima data dari jaringan CAN agar dapat dibaca kembali oleh mikrokontroler.

Penggunaan TJA1050 sangat penting pada sistem komunikasi terdistribusi yang membutuhkan multi-node (lebih dari satu perangkat saling terhubung), misalnya dalam ECU kendaraan, sistem kontrol robot, sensor jarak jauh, atau otomasi industri. Dengan mempelajari praktikum ini, mahasiswa akan memahami dasar komunikasi CAN menggunakan ESP32 dan modul TJA1050 sebagai transceiver.

➤ Tujuan

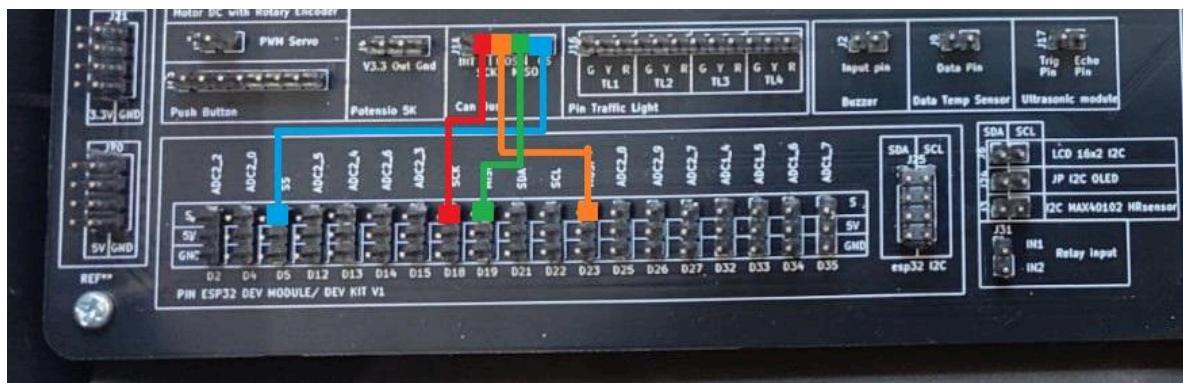
- Mengenalkan konsep komunikasi CAN bus pada ESP32.
- Menunjukkan peran TJA1050 sebagai transceiver dalam sistem CAN.
- Melatih mahasiswa mengirim data sederhana melalui jaringan CAN.
- Mempersiapkan mahasiswa memahami aplikasi CAN di dunia otomotif dan industri.

➤ Alat dan Bahan

- ESP32 DevKit V1 (Sudah terpasang dalam box modul mikrokontroler)
- Kabel jumper
- Laptop yang terdapat aplikasi Arduino IDE
- Kabel USB
- CAN bus TJA1050 (Sudah terpasang dalam box modul mikrokontroler)

➤ Prosedur

- a. Penyusunan Rangkaian



Gambar 30

- Hubungkan pin SCK pada kotak can bus ke pin D18 ESP32 (gambar 30)
 - Hubungkan pin MISO pada kotak can bus ke pin D19 ESP32 (gambar 30)
 - Hubungkan pin MOSI pada kotak can bus ke pin D23 ESP32 (gambar 30)
 - Hubungkan pin CS pada kotak can bus ke pin D5 ESP32 (gambar 30)
- b. Persiapan Arduino IDE
- Buka Arduino IDE di laptop.
 - Download library sebagai ZIP untuk MCP2515 di GitHub melalui link "<https://github.com/autowp/arduino-mcp2515>"
 - Tambahkan library dalam bentuk zip ke Arduino IDE dengan pilih Sketch > Include library > Add ZIP Library
 - Pastikan Board ESP32 Dev Module sudah dipilih.
 - Pilih port COM sesuai ESP32 yang terhubung.
- c. Pemrograman ESP32
- Salin program yang ditunjukkan pada Gambar 31 ke ESP32 menggunakan kabel Micro USB.

```

#include <mcp2515.h>
#include <SPI.h>

MCP2515 mcp2515(5); // pastikan ini CS pin yang benar
struct can_frame canMsg;

void setup() {
    Serial.begin(115200);
    SPI.begin(18,19,23,5);
    mcp2515.reset();
    // ganti MCP_8MHZ ke MCP_16MHZ kalau modulmu punya kristal 16MHz
    if (mcp2515.setBitrate([CAN_100KBPS, MCP_16MHZ]) != MCP2515::ERROR_OK) {
        Serial.println("Gagal set bitrate!");
    } else Serial.println("Bitrate set OK");
    mcp2515.setNormalMode();
    Serial.println("Ketik pesan dan tekan ENTER:");
}

void loop() {
    if (Serial.available()) {
        String s = Serial.readStringUntil('\n');
        s.trim();
        if (s.length() == 0) return;

        // kirim max 8 byte satu frame
        int len = min(8, (int)s.length());
        canMsg.can_id = 0x100;
        canMsg.can_dlc = len;
        memcpy(canMsg.data, s.c_str(), len);

        // kirim
        mcp2515.sendMessage(&canMsg);
        Serial.print("Terkirim frame ID 0x");
        Serial.print(canMsg.can_id, HEX);
        Serial.print(" data: ");
        for (int i=0;i<len;i++) Serial.print((char)canMsg.data[i]);
        Serial.println();
    }
}

```

Gambar 31

- Hubungkan Kabel USB dari Laptop ke ESP32 dan Upload Program (Sesuaikan dengan port masing masing Laptop)
- Tunggu hingga proses upload selesai dan pastikan tidak ada error pada Arduino IDE.
- Buka Serial Monitor pada pojok kanan arduino IDE dan ganti baudrate 115200

- Ketik “h” dan tekan enter pada serial monitor maka akan muncul “ Terkirim frame ID 0x : h”

➤ Tugas

1. Ubah ID CAN menjadi 0x100 dan isi data dengan angka 1–8.
 2. Buat sistem sederhana: jika data pertama bernilai 0x01, nyalakan LED bawaan ESP32.
-

