

LAPORAN UTS MIKROPROCESSOR



Dosen : Nurseno Bayu Aji, S.Kom.,M.Kom.

Mata Kuliah : Mikroprosesor & Mikrokontroller

Nama : Faiz Akmal Nurhakim

Nim : 4.33.23.2.11

Program Studi Teknologi Rekayasa Komputer
Jurusan Teknik Elektro Politeknik Negeri Semarang

1. Rangkaian
LED 1 = A2
LED 2 = Pin 10 karena nomor absen saya 11 - 1
LED 3 = Pin 11
Button = Pin 2
2. Kode C

```
// Definisi pin untuk LED
const int led1 = A2; // LED 1 di pin A2
const int led2 = 10; // LED 2 di pin 10
const int led3 = 11; // LED 3 di pin 11
const int buttonPin = 2; // Button di pin digital 2

// Variable untuk timing LED
unsigned long previousMillis2 = 0; // untuk LED 2
unsigned long previousMillis3 = 0; // untuk LED 3
const long interval2 = 200; // interval kedip LED 2 (cepat) - 200ms
const long interval3 = 1000; // interval kedip LED 3 (lambat) - 1000ms

// Variable untuk status
bool buttonPressed = false;
bool led2State = false;
bool led3State = false;

void setup() {
  // Set pin LED sebagai output
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);

  // Set pin button sebagai input dengan pull-up resistor internal
  pinMode(buttonPin, INPUT_PULLUP);

  // Matikan semua LED di awal
  digitalWrite(led1, LOW);
  digitalWrite(led2, LOW);
  digitalWrite(led3, LOW);
}

void loop() {
  // Baca status button
  if (digitalRead(buttonPin) == LOW) {
    delay(50); // Debounce delay
    buttonPressed = !buttonPressed; // Toggle status

    if (buttonPressed) {
```

```

    digitalWrite(led1, HIGH); // Nyalakan LED 1
} else {
    // Matikan semua LED
    digitalWrite(led1, LOW);
    digitalWrite(led2, LOW);
    digitalWrite(led3, LOW);
}

// Tunggu sampai button dilepas
while(digitalRead(buttonPin) == LOW);
}

// Jika button sudah ditekan, jalankan pola kedip
if (buttonPressed) {
    unsigned long currentMillis = millis();

    // LED 2 kedip cepat
    if (currentMillis - previousMillis2 >= interval2) {
        previousMillis2 = currentMillis;
        led2State = !led2State;
        digitalWrite(led2, led2State);
    }

    // LED 3 kedip lambat
    if (currentMillis - previousMillis3 >= interval3) {
        previousMillis3 = currentMillis;
        led3State = !led3State;
        digitalWrite(led3, led3State);
    }
}
}
}

```

3. Kode Assembly

```

.global _start

.section .text
_start:
    // Inisialisasi pin
    // Set pin LED 1 (A2), LED 2 (pin 10), LED 3 (pin 11) sebagai OUTPUT
    // Set pin button (pin 2) sebagai INPUT_PULLUP
    // Cek pin button dan status LED sesuai tombol ditekan

    // Mengatur pin mode: LED1, LED2, LED3 sebagai output
    ldi r16, 0b11111100    // Bitwise untuk set pin sebagai output: A2, 10, 11 (misal)
    out DDRC, r16          // Mengatur register DDR untuk A2
    out DDRB, r16          // Mengatur register DDR untuk pin 10 dan pin 11

    // Set pin button sebagai input_pullup
    ldi r16, 0b11111101    // Set pin 2 sebagai input dengan pull-up (untuk pin 2)
    out DDRD, r16          // Set DDR untuk pin 2

main_loop:
    // Baca status tombol
    in r17, PIND           // Baca register input untuk pin D (pin 2)
    sbrc r17, 2            // Cek jika tombol ditekan (pin 2 rendah)
    rjmp button_pressed    // Jika tombol ditekan, lompat ke button_pressed

    rjmp main_loop         // Kembali ke loop utama jika tombol tidak ditekan

```

```

button_pressed:
    // Toggle status LED
    // LED1 menyala (A2)
    ldi r16, 0b00000100    // LED 1 nyala pada bit ke-2 (A2)
    out PORTC, r16        // Mengaktifkan LED 1

    // Cek untuk kedipan LED2 (kecepatan tinggi)
    call blink_led2        // Panggil fungsi untuk kedipan cepat LED 2

    // Cek untuk kedipan LED3 (kecepatan rendah)
    call blink_led3        // Panggil fungsi untuk kedipan pelan LED 3

    // Kembali ke loop utama setelah tombol dilepas
    rjmp main_loop

blink_led2:
    // Fungsi kedipan LED 2 (pin 10), cepat (200ms)
    ldi r18, 200            // Interval 200ms untuk LED 2
    call delay              // Panggil fungsi delay
    // Toggle LED 2
    in r17, PORTB           // Baca port pin 10
    eor r17, 0b00000001    // Toggle LED 2 (flip bit ke-0)
    out PORTB, r17          // Set status LED 2
    ret                    // Kembali ke caller

blink_led3:
    // Fungsi kedipan LED 3 (pin 11), pelan (1000ms)
    ldi r24, 3              // Load high byte of 1000 (1000 >> 8)
    ldi r25, 232            // Load low byte of 1000 (1000 & 0xFF)
    call delay_long         // Panggil fungsi delay panjang
    // Toggle LED 3
    in r17, PORTB           // Baca port pin 11
    eor r17, 0b00000010    // Toggle LED 3 (flip bit ke-1)
    out PORTB, r17          // Set status LED 3
    ret                    // Kembali ke caller

delay:
    // Fungsi untuk delay dalam satuan waktu
    push r18                // Simpan register r18 untuk interval
delay_loop:
    dec r18                 // Kurangi interval
    brne delay_loop         // Jika interval belum habis, ulangi loop
    pop r18                 // Kembalikan register r18
    ret                    // Kembali ke caller

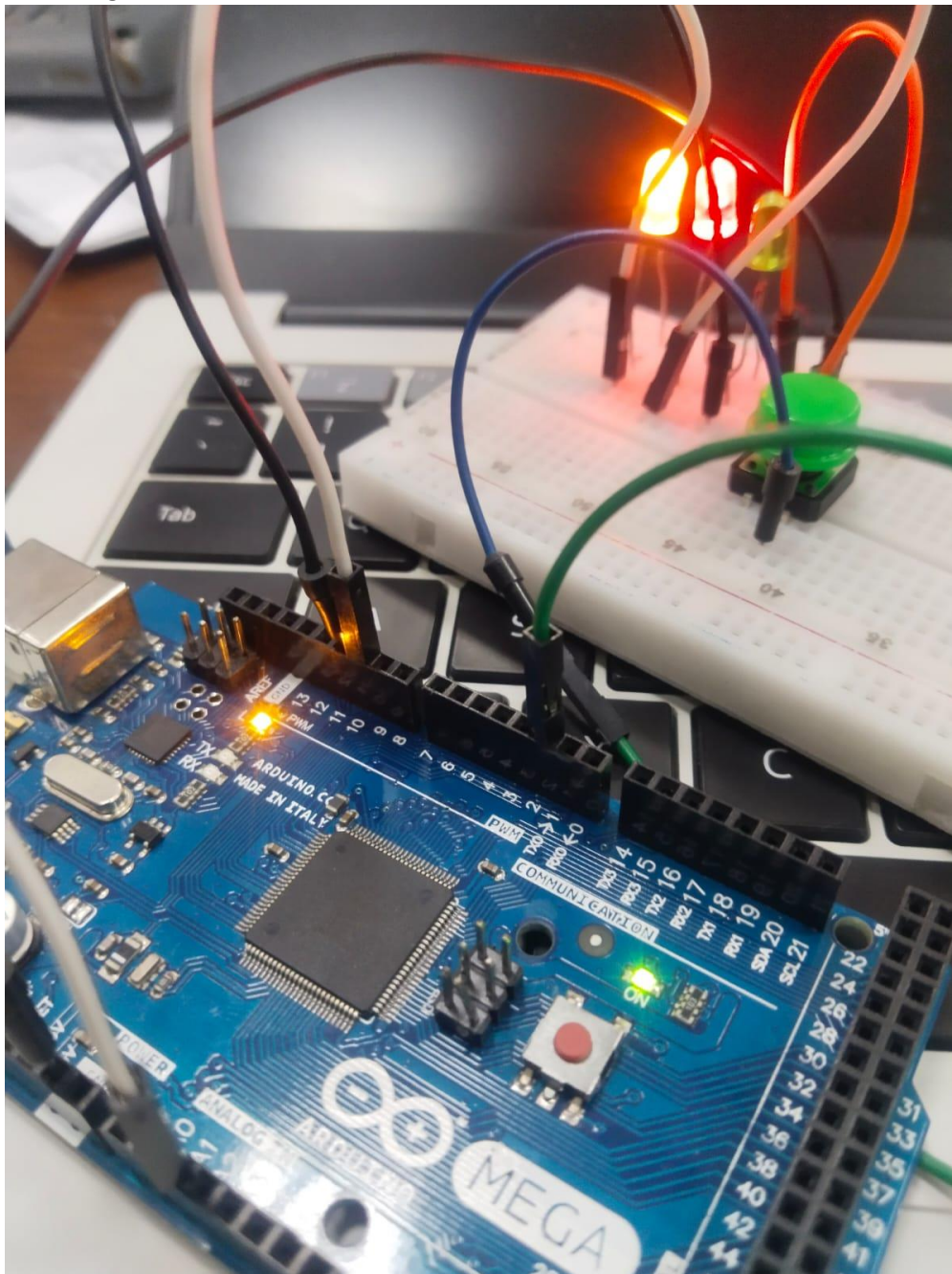
```

```

delay_long:
    // Fungsi untuk delay lebih panjang (untuk interval 1000ms)
    push r24                // Simpan register r24
    push r25                // Simpan register r25
long_delay_loop:
    dec r25                 // Kurangi byte rendah
    brne long_delay_loop    // Jika belum habis, ulangi
    dec r24                 // Kurangi byte tinggi
    brne long_delay_loop    // Jika byte tinggi belum habis, ulangi
    pop r25                 // Kembalikan register r25
    pop r24                 // Kembalikan register r24
    ret                     // Kembali ke caller

```

4. Foto Rangkaian & Hasil



5. Kesulitan

Kurang memahami Bahasa assembly