MIKROPROSESOR & MIKROKONTROLLER

UTS

Mata Kuliah : Sistem Tertanam

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

Disusun oleh

Nama   : Ilham Indra Atmaja

NIM   : 4.33.23.2.16

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI SEMARANG

- **Code**

  Kode Untuk C

```c
const int led1 = A2;
const int led2 = 3;
const int led3 = 8;
const int buttonPin = 7;

unsigned long prevMillis2 = 0, prevMillis3 = 0;
const long interval2 = 200, interval3 = 1000;

bool buttonPressed = false;

void setup() {
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
  pinMode(buttonPin, INPUT_PULLUP);
}

void loop() {
  if (digitalRead(buttonPin) == LOW) {
    delay(50); // Debounce
    buttonPressed = !buttonPressed;
    digitalWrite(led1, buttonPressed ? HIGH : LOW);
    if (!buttonPressed) { // Jika tombol dilepas, matikan semua LED
      digitalWrite(led2, LOW);
      digitalWrite(led3, LOW);
    }
    while (digitalRead(buttonPin) == LOW); // Tunggu tombol dilepas
  }

  if (buttonPressed) {
    unsigned long currMillis = millis();
    if (currMillis - prevMillis2 >= interval2) {
      prevMillis2 = currMillis;
      digitalWrite(led2, !digitalRead(led2)); // Toggle LED2
    }
    if (currMillis - prevMillis3 >= interval3) {
      prevMillis3 = currMillis;
      digitalWrite(led3, !digitalRead(led3)); // Toggle LED3
    }
  }
}
```

Kode Untuk Assembly

```
.global _start

.section .text
_start:
    // Set LED1 (A2/PORTC.2), LED2 (PD3/PORTD.3), LED3 (PB0/PORTB.0) sebagai OUTPUT
    ldi r16, 0b00000100
    out DDRC, r16           // LED1
    ldi r16, 0b00001000
    out DDRD, r16           // LED2
    ldi r16, 0b00000001
    out DDRB, r16           // LED3

    // Set Button (PD7) sebagai INPUT_PULLUP
    ldi r16, 0b01111111
    out DDRD, r16           // Button
    ldi r16, 0b10000000
    out PORTD, r16          // Pull-up Button

main_loop:
    in r17, PIND            // Baca tombol
    sbrs r17, 7             // Jika tombol ditekan
    rjmp button_pressed
    rjmp main_loop          // Ulangi

button_pressed:
    ldi r16, 0b00000100
    out PORTC, r16          // Nyalakan LED1
    call blink_led2         // Kedip LED2 cepat
    call blink_led3         // Kedip LED3 lambat
    rjmp main_loop          // Kembali ke loop

blink_led2:
    ldi r18, 200
    call delay              // Delay 200ms
    in r17, PORTD
    eor r17, 0b00001000     // Toggle LED2
    out PORTD, r17
    ret
```

```
blink_led2:
    ldi r18, 200
    call delay              // Delay 200ms
    in r17, PORTD
    eor r17, 0b00001000     // Toggle LED2
    out PORTD, r17
    ret

blink_led3:
    ldi r18, 1000
    call delay_long         // Delay 1000ms
    in r17, PORTB
    eor r17, 0b00000001     // Toggle LED3
    out PORTB, r17
    ret

delay:
    push r18
delay_loop:
    dec r18
    brne delay_loop
    pop r18
    ret

delay_long:
    push r24
    push r25
long_delay_loop:
    dec r25
    brne long_delay_loop
    dec r24
    brne long_delay_loop
    pop r25
    pop r24
    ret
```

```asm
.global _start

.section .text
_start:
    // Set LED1 (A2/PORTC.2), LED2 (PD3/PORTD.3), LED3 (PB0/PORTB.0)
sebagai OUTPUT
    ldi r16, 0b00000100
    out DDRC, r16          // LED1
    ldi r16, 0b00001000
    out DDRD, r16          // LED2
    ldi r16, 0b00000001
    out DDRB, r16          // LED3


    // Set Button (PD7) sebagai INPUT_PULLUP
    ldi r16, 0b01111111
    out DDRD, r16          // Button
    ldi r16, 0b10000000
    out PORTD, r16         // Pull-up Button


main_loop:
    in r17, PIND          // Baca tombol
    sbrs r17, 7           // Jika tombol ditekan
    rjmp button_pressed
    rjmp main_loop        // Ulangi


button_pressed:
    ldi r16, 0b00000100
    out PORTC, r16         // Nyalakan LED1
    call blink_led2        // Kedip LED2 cepat
    call blink_led3        // Kedip LED3 lambat
    rjmp main_loop         // Kembali ke loop


blink_led2:
    ldi r18, 200
```

```asm
        call delay              // Delay 200ms
        in r17, PORTD
        eor r17, 0b00001000     // Toggle LED2
        out PORTD, r17
        ret

blink_led3:
        ldi r18, 1000
        call delay_long         // Delay 1000ms
        in r17, PORTB
        eor r17, 0b00000001     // Toggle LED3
        out PORTB, r17
        ret

delay:
        push r18
delay_loop:
        dec r18
        brne delay_loop
        pop r18
        ret

delay_long:
        push r24
        push r25
long_delay_loop:
        dec r25
        brne long_delay_loop
        dec r24
        brne long_delay_loop
        pop r25
        pop r24
        ret
```
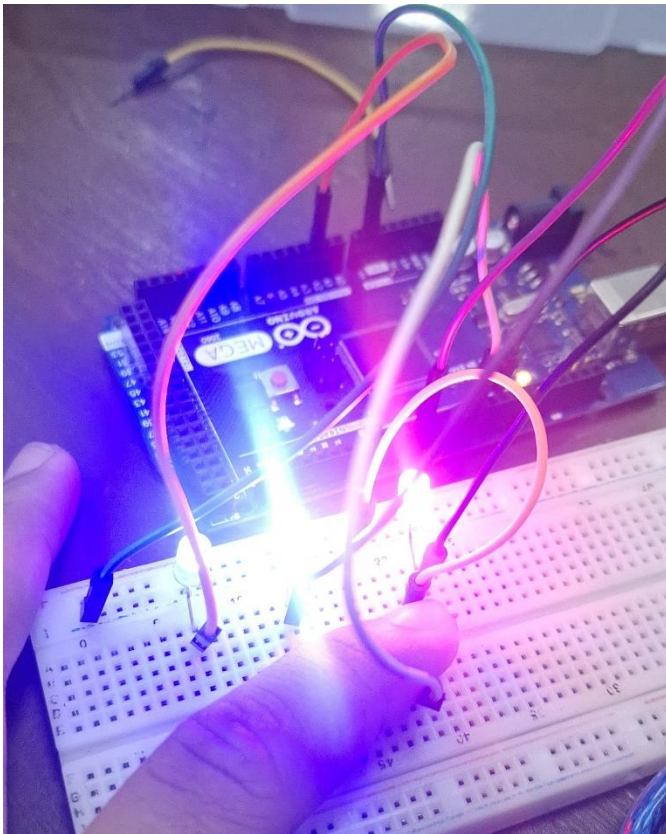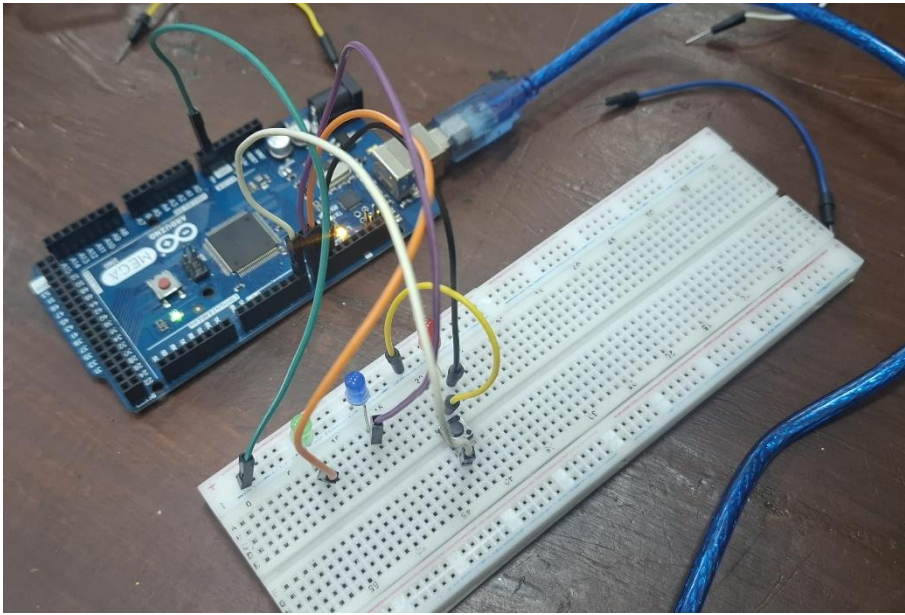
- **Rangkaian**

- **Kendala**

  Kesulitan dalam pengerjaan uts adalah membuat kode assembly yang sesuai dan juga kode C
  untuk dapat Adruino IDE membaca kode assembly