

Lab Tutorial DDAK 7.0

Deadline: **29 Desember 2020 Pkl 23.55 WIB**

File yang dikumpulkan :

1. File AVR (.asm)
2. File Hapsim (.xml)
3. PDF

Format pengumpulan: semua file dikumpulkan menjadi **sat folder lalu di zip**

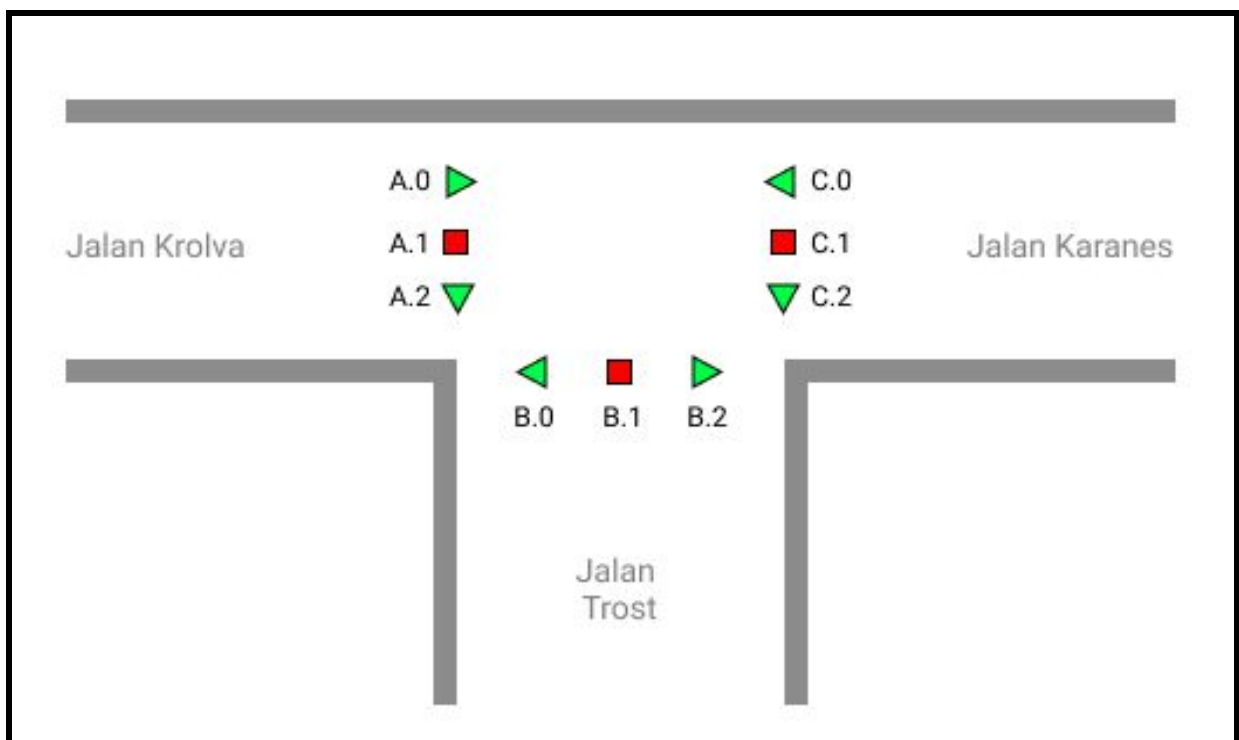
Format Penamaan:

1. Lab7-[Kode Asdos]-[Kelas]-[Nama]-[NPM].asm
2. Lab7-[Kode Asdos]-[Kelas]-[Nama]-[NPM].xml
3. Lab7-[Kode Asdos]-[Kelas]-[Nama]-[NPM].pdf
4. Lab7-[Kode Asdos]-[Kelas]-[Nama]-[NPM].zip

Ketentuan keterlambatan Pengumpulan Lab : **Pengurangan nilai sebanyak 10% per 1 jam keterlambatan** terhitung dari waktu deadline yang sudah ditentukan dengan maksimal pengurangan sebesar 50%. (Contoh : deadline 23.55 , maka pengumpulan di atas 23.55 sampai 00.55 mendapatkan pengurangan nilai sebesar 10%)

Tutorial 7- Internal Interrupt with AVR

Pada sebuah area di Distrik Royal Capital, akan dibangun pertigaan jalan yang menghubungkan Jalan Krolva, Jalan Trost, dan Jalan Karanes. Untuk itu, harus dibuat sebuah sistem pengaturan lalu lintas tersebut dengan menggunakan AVR ATmega8515. Ilustrasinya dapat dilihat seperti di bawah ini.



Pada setiap ruas jalan, memiliki 3 lampu lalu lintas, yaitu 2 lampu hijau dan 1 lampu merah. Jika ingin berjalan, pengendara harus menunggu hingga lampu hijau di ruas tersebut menyala. Jika lampu merah menyala, maka semua kendaraan di ruas jalan itu harus berhenti, baik yang ingin mengambil jalur lurus, ke kiri atau ke kanan. Khusus untuk lampu hijau pada **Jalan Krolva**, lampu hijau untuk arah lurus akan selalu menyala walaupun lampu merah di **Jalan Krolva** menyala.

Setiap lampu terhubung dengan PORT pada AVR sesuai konfigurasi gambar. Format LED di setiap mode akan disimpan dalam sebuah variabel mode. Dengan keterangan 0x00 untuk mode A, 0x01 untuk mode B dan 0x02 untuk mode C. Sistem ini dilengkapi oleh 2 tombol SWITCH_MODE dan menyebrang. Apabila tombol SWITCH_MODE ditekan, maka lampu hijau

untuk ruas tersebut menyala secara bergantian mulai dari A-B lalu B-C dan C-A dan seterusnya. Apabila tombol menyebrang ditekan, maka sistem akan masuk pada mode CROSS dan semua lampu merah akan menyala di setiap mode tanpa terkecuali. Sistem ini juga dilengkapi oleh SWITCH_MODE secara otomatis dengan menggunakan timer. Sistem ini memanfaatkan fungsi internal interrupt dengan menggunakan timer 1 Overflow. Untuk dapat berubah mode secara otomatis, terdapat counter untuk menghitung lama waktu pergantian mode. Apabila counter menunjukkan waktu 60 detik, maka sistem secara otomatis akan berganti mode sesuai dengan urutannya.

1. Lengkapi cuplikan kode di bawah ini yang telah disisipkan. Kode tersebut merupakan template program yang akan kalian ubah sesuai dengan implementasi pada narasi di atas.
2. Buatlah refleksi pribadi mengenai Lab 7 ini. Kumpulkan dengan format .pdf digabungkan dengan file lainnya dalam .zip.



```
include "m8515def.inc"
.def temp =r18
.def mode=r19
.def counter=r20

; Lengkapi bagian ini
.org $00 rjmp START
.org $01 rjmp SWITCH_MODE
.org $02 rjmp CROSS
.org $06 rjmp .....

START:
ldi temp,low(RAMEND)
out SPL,temp
ldi temp,high(RAMEND)
out SPH,temp
ldi temp,$ff
out DDRA, temp ; Set port A as output
out DDRB, temp ; Set port B as output
out DDRC, temp ; Set port C as output

SETUP_EXT_INTERRUPT:
ldi r16, 0b00001010
out MCUCR, r16
ldi r16, 0b11000000
out GICR, r16

; Lengkapi bagian ini
SETUP_TIMER_INTERRUPT:
ldi r16, (1<<CS10) | (1<<CS12)
out .....,r16
ldi r16,.....
out TIFR,r16
ldi r16,.....
out TIMSK,r16
Sei

MAIN:
cpi mode, 0x00
```



```
beq ModeA
cpi mode, 0x01
beq ModeB
cpi mode, 0x02
beq ModeC
rjmp MAIN
```

; Lengkapi bagian ini

ModeA :

```
ldi r16, .....
out PORTA, r16
ldi r16, .....
out PORTB, r16
ldi r16, .....
out PORTC, r16
rjmp MAIN
```

; Lengkapi bagian ini

ModeB:

```
ldi r16, .....
out PORTA, r16
ldi r16, .....
out PORTB, r16
ldi r16, .....
out PORTC, r16
rjmp MAIN
```

; Lengkapi bagian ini

ModeC:

```
ldi r16, .....
out PORTA, r16
ldi r16, .....
out PORTB, r16
ldi r16, .....
out PORTC, r16
rjmp MAIN
```

; Lengkapi bagian ini

SWITCH_MODE:



```
ldi r16, 1
add mode, r16
cpi mode, .....
brne Exit_Int
ldi mode, .....
```

```
Exit_Int:
reti
```

```
; Lengkapi bagian ini
CROSS:
ldi r16, .....
out PORTA, r16
ldi r16, .....
out PORTB, r16
ldi r16, .....
out PORTC, r16
rcall DELAY_SHORT
reti
```

```
; Lengkapi bagian ini
DELAY_SHORT:
.....
```

TODO: Lengkapi cuplikan program diatas. Anda bebas menambahkan asumsi dan memodifikasi program.