

```

;=====
; RESET and INTERRUPT VECTORS
;=====

;
.org $00
rjmp MAIN
.org $01
rjmp INTERRUPT_LIBUR
.org $02
rjmp INTERRUPT_DEFENSE
.org $0E
rjmp INTERRUPT_SWITCH_JADWAL

;=====
; CODE SEGMENT
;=====

MAIN:

```

.org \$01	set alamat \$01 untuk mengatur instruksi selanjutnya di alamat tersebut
rjmp INTERRUPT_LIBUR	jump ke INTERRUPT_LIBUR
.org \$02	set alamat \$02 untuk mengatur instruksi selanjutnya di alamat tersebut
rjmp INTERRUPT_DEFENSE	jump ke INTERRUPT_DEFENSE
.org \$0E	set alamat \$0E untuk mengatur instruksi selanjutnya di alamat tersebut
rjmp INTERRUPT_SWITCH_JADWAL	jump ke INTERRUPT_SWITCH_JADWAL

```

;=====
; CODE SEGMENT
;=====

MAIN:
    ldi count_jadwal, 0 ;todo

INIT_STACK:
    ldi temp, low(RAMEND)
    out SPL, temp
    ldi temp, high(RAMEND)
    out SPH, temp

INIT_LED:

```

C:\Users\deme\OneDrive\Documents\S1\Sem 2\DDAK\7\Lab7_TOP_B_ClementSamuelMarly_2206082114_Genap.asm

MAIN:

ldi count_jadwal, 0 set count_jadwal dengan nilai 0

INIT_STACK: inisiasi stack pointer

ldi temp, low(RAMEND) ambil nilai byte rendah dari alamat RAMEND dan set temp dengan nilai tersebut

out SPL, temp menyimpan nilai byte rendah tersebut ke dalam register SPL, mengatur byte rendah dari stack pointer

ldi temp, high(RAMEND) ambil nilai byte tinggi dari alamat RAMEND dan set temp dengan nilai tersebut

out SPH, temp menyimpan nilai byte tinggi tersebut ke dalam register SPL, mengatur byte tinggi dari stack pointer

```

INIT_LED:
    ser temp
    out DDRA, temp
    out DDRB, temp
    out DDRC, temp

INIT_EXT_INTERRUPT:
    ldi temp, 0b00001111
    out MCUCR, temp
    ldi temp, 0b11000000
    out GICR, temp

INIT_INT_INTERRUPT:

```

C:\Users\deme\OneDrive\Documents\S1\Sem 2\DDAK\7\Lab7_TOP_B_ClementSamuelMarly_2206082114_Genap.asm

INIT_LED:

ser temp set semua bit temp menjadi 1

out DDRA, temp set semua LED yang terhubung dengan PORTA dengan nilai temp.

out DDRB, temp set semua LED yang terhubung dengan PORTB dengan nilai temp.

out DDRC, temp set semua LED yang terhubung dengan PORTC dengan nilai temp.

INIT_EXIT_INTERRUPT:

ldi temp, 0b00001111 set variabel temp dengan nilai 0b00001111

out MCUCR, temp mengirimkan nilai temp ke register MCUCR untuk set interrupt

ldi temp, 0b11000000 set variabel temp dengan nilai 0b11000000

out GICR, temp mengirimkan nilai temp ke register GICR untuk set interrupt

```

INIT_INT_INTERRUPT:
    ldi temp, (1 << CS02) | (1 << CS00)
    out TCCR0, temp
    ldi temp, (1 << TOV0) | (1 << OCF0)
    out TIFR, temp
    ldi temp, (1 << OCIE0)
    out TIMSK, temp
    ldi temp, 128
    out OCR0, temp
    sei

```

C:\Users\cleme\OneDrive\Documents\S1\Sem 2\DDAK\7\Lab7_TOP_B_ClementSamuelMarly_2206082114_Genap.asm

ldi temp, (1 << CS02) (1 << CS00)	set temp dengan bit CS02 atau CS00
out TCCR0, temp	mengirimkan nilai temp ke register TCCR0 untuk mengatur timer
ldi temp, (1 << TOV0) (1 << OCF0)	set nilai temp dengan hasil operasi bit shift (1 << OCF0) atau (1 << TOV0).
out TIFR, temp	mengirimkan nilai temp ke TIFR untuk menghapus status flag overflow timer dan flag compare match.
ldi temp, (1 << OCIE0)	set nilai temp dengan hasil operasi bit shift (1 << OCIE0)
out TIMSK, temp	mengirimkan nilai temp ke TIMSK untuk mengatur interrupt ketika terjadi compare match pada timer
ldi temp, 128	set nilai temp = 128
out OCR0, temp	mengirimkan nilai temp ke OCR0, 128 digunakan sebagai nilai compare match pada timer
sei	Set Enable Interrupt

```

NORMAL:
    clr libur_flag
    rjmp CHECK_JADWAL

CHECK_JADWAL:
    cpi count_jadwal, 1
    breq JADWAL_02
    brlo JADWAL_01
    brne JADWAL_03

```

C:\Users\cleme\OneDrive\Documents\S1\Sem 2\DDAK\7\Lab7_TOP_B_ClementSamuelMarly_2206082114_Genap.asm

NORMAL:	
clr libur_flag	clear nilai libur_flag menjadi 0
CHECK_JADWAL:	
cpi count_jadwal, 1	cek nilai count_jadwal
breq JADWAL_02	jika nilai count_jadwal = 1, lompat ke JADWAL_02
brlo JADWAL_01	jika nilai count_jadwal < 1, lompat ke JADWAL_01
brne JADWAL_03	jika nilai count_jadwal != 1, lompat ke JADWAL_03

```

JADWAL_01:
    sbrc persiapan, 0
    rcall PERSIAPAN_JADWAL_01
    ldi temp, 0b00000100
    out PORTA, temp
    out PORTB, temp
    ldi temp, 0b00010000
    out PORTC, temp
    rcall DELAY
    rjmp CHECK_STATUS

```

JADWAL_02:

C:\Users\cleme\OneDrive\Documents\S1\Sem 2\DDAK\7\Lab7_TOP_B_ClementSamuelMarly_2206082114_Genap.asm

sbrc persiapan, 0	cek bit 0 pada variabel persiapan.
--------------------------	------------------------------------

rcall PERSIAPAN_JADWAL_01

panggil label PERSIAPAN_JADWAL_01 apabila bit 0 pada variabel persiapan adalah 1, jika 0 line ini dilewati

ldi temp, 0b00000100

set nilai temp dengan 0b00000100

out PORTA, temp

mengirimkan nilai temp ke PORTA (mengatur LED di yang terhubung PORTA)

out PORTB, temp

mengirimkan nilai temp ke PORTB (mengatur LED di yang terhubung PORTB)

ldi temp, 0b00010000

set nilai temp dengan 0b00010000

out PORTC, temp

mengirimkan nilai temp ke PORTC (mengatur LED di yang terhubung PORTC)

```
JADWAL_02:
    sbrc persiapan, 0
    rcall PERSIAPAN_JADWAL_02
    ldi temp, 0b00010000
    out PORTA, temp
    ldi temp, 0b00000100
    out PORTB, temp
    out PORTC, temp
    rcall DELAY
    rjmp CHECK_STATUS
```

C:\Users\deme\OneDrive\Documents\S1\Sem 2\DDAK\7\Lab7_TOP_B_ClementSamuelMarly_2206082114_Genap.asm

sbrc persiapan, 0

cek bit 0 pada variabel persiapan.

rcall PERSIAPAN_JADWAL_02

panggil label PERSIAPAN_JADWAL_02 apabila bit 0 pada variabel persiapan adalah 1, jika 0 line ini dilewati

ldi temp, 0b00010000

set nilai temp dengan 0b00010000

out PORTA, temp

mengirimkan nilai temp ke PORTA (mengatur LED di yang terhubung PORTA)

ldi temp, 0b00000100

set nilai temp dengan 0b00000100

out PORTB, temp

mengirimkan nilai temp ke PORTB (mengatur LED di yang terhubung PORTB)

out PORTC, temp

mengirimkan nilai temp ke PORTC (mengatur LED di yang terhubung PORTC)

rjmp CHECK_STATUS

lompat ke CHECK_STATUS

```
JADWAL_03:
    sbrc persiapan, 0
    rcall PERSIAPAN_JADWAL_03
    ldi temp, 0b00010000
    out PORTB, temp
    ldi temp, 0b00000100
    out PORTC, temp
    out PORTA, temp
    rcall DELAY
    rjmp CHECK_STATUS
```

PERSIAPAN_JADWAL_01:

C:\Users\deme\OneDrive\Documents\S1\Sem 2\DDAK\7\Lab7_TOP_B_ClementSamuelMarly_2206082114_Genap.asm

sbrc persiapan, 0

cek bit 0 pada variabel persiapan.

rcall PERSIAPAN_JADWAL_03

panggil label PERSIAPAN_JADWAL_03 apabila bit 0 pada variabel persiapan adalah 1, jika 0 line ini dilewati

ldi temp, 0b00010000

set nilai temp dengan 0b00010000

out PORTB, temp

mengirimkan nilai temp ke PORTB (mengatur LED di yang terhubung PORTB)

ldi temp, 0b00000100

set nilai temp dengan 0b00000100

out PORTC, temp

mengirimkan nilai temp ke PORTC (mengatur LED di yang terhubung PORTC)

out PORTA, temp

mengirimkan nilai temp ke PORTA (mengatur LED di yang terhubung PORTA)

rjmp CHECK_STATUS

lompat ke CHECK_STATUS

```
PERSIAPAN_JADWAL_01:
    ldi temp, 0b00001000
    out PORTB, temp
    rcall DELAY
    clr persiapan
    ret

PERSIAPAN_JADWAL_02:
    ldi temp, 0b00001000
    out PORTC, temp
    rcall DELAY
    clr persiapan
    ret

PERSIAPAN_JADWAL_03:
    ldi temp, 0b00001000
    out PORTA, temp
    rcall DELAY
    clr persiapan
    ret
```

DEFENSE:

C:\Users\cleme\OneDrive\Documents\S1\Sem 2\DDAK\7\Lab7_TOP_B_ClementSamuelMarly_2206082114_Genap.asm

PERSIAPAN_JADWAL_01: set LED Persiapan B

ldi temp, 0b00001000 set nilai temp dengan 0b00001000

out PORTB, temp mengirimkan nilai temp ke PORTB (mengatur LED di yang terhubung PORTB)

PERSIAPAN_JADWAL_02: set LED Persiapan C

ldi temp, 0b00001000 set nilai temp dengan 0b00001000

out PORTC, temp mengirimkan nilai temp ke PORTC (mengatur LED di yang terhubung PORTC)

PERSIAPAN_JADWAL_03: set LED Persiapan A

ldi temp, 0b00001000 set nilai temp dengan 0b00001000

out PORTA, temp mengirimkan nilai temp ke PORTA (mengatur LED di yang terhubung PORTA)

```
DEFENSE:
    ldi temp, 0b01100011
    out PORTA, temp
    out PORTB, temp
    out PORTC, temp
    rcall DELAY
    ldi temp, 0b00000000
    out PORTA, temp
    out PORTB, temp
    out PORTC, temp
    rcall DELAY
    rjmp CHECK_STATUS
```

C:\Users\cleme\OneDrive\Documents\S1\Sem 2\DDAK\7\Lab7_TOP_B_ClementSamuelMarly_2206082114_Genap.asm

ldi temp, 0b01100011 set nilai temp dengan 0b01100011

out PORTA, temp mengirimkan nilai temp ke PORTA (mengatur LED di yang terhubung PORTA)

out PORTB, temp mengirimkan nilai temp ke PORTB (mengatur LED di yang terhubung PORTB)

out PORTC, temp mengirimkan nilai temp ke PORTC (mengatur LED di yang terhubung PORTC)

rcall DELAY panggil label DELAY

ldi temp, 0b00000000 set nilai temp dengan 0b00000000

out PORTA, temp	mengirimkan nilai temp ke PORTA (mengatur LED di yang terhubung PORTA)
out PORTB, temp	mengirimkan nilai temp ke PORTB (mengatur LED di yang terhubung PORTB)
out PORTC, temp	mengirimkan nilai temp ke PORTC (mengatur LED di yang terhubung PORTC)

```
LIBUR:
    dec count_libur
    ldi temp, 0b00010000
    out PORTA, temp
    out PORTB, temp
    out PORTC, temp
    rcall DELAY
    ldi temp, 0b00000000
    out PORTA, temp
    out PORTB, temp
    out PORTC, temp
    rcall DELAY
    rjmp CHECK_STATUS
```

C:\Users\cleme\OneDrive\Documents\S1\Sem 2\DDAK\7\Lab7_TOP_B_ClementSamuelMarly_2206082114_Genap.asm

dec count_libur	decrement nilai count_libur (nilai dikurang 1)
ldi temp, 0b00010000	set nilai temp dengan 0b00010000
out PORTA, temp	mengirimkan nilai temp ke PORTA (mengatur LED di yang terhubung PORTA)
out PORTB, temp	mengirimkan nilai temp ke PORTB (mengatur LED di yang terhubung PORTB)
out PORTC, temp	mengirimkan nilai temp ke PORTC (mengatur LED di yang terhubung PORTC)
rcall DELAY	panggil label DELAY
ldi temp, 0b00000000	set nilai temp dengan 0b00000000
out PORTA, temp	mengirimkan nilai temp ke PORTA (mengatur LED di yang terhubung PORTA)
out PORTB, temp	mengirimkan nilai temp ke PORTB (mengatur LED di yang terhubung PORTB)
out PORTC, temp	mengirimkan nilai temp ke PORTC (mengatur LED di yang terhubung PORTC)

```
INTERRUPT_SWITCH_JADWAL:
    sbrc count_libur, 0
    rjmp EXIT_INTERRUPT
    inc count_jadwal
    inc persiapan
    cpi count_jadwal, 3
    breq RESET
    brne EXIT_INTERRUPT

RESET:
    clr count_jadwal
    rjmp EXIT_INTERRUPT
```

C:\Users\cleme\OneDrive\Documents\S1\Sem 2\DDAK\7\Lab7_TOP_B_ClementSamuelMarly_2206082114_Genap.asm

INTERRUPT_SWITCH_JADWAL:

sbrc count_libur, 0	cek bit 0 pada variabel count_libur
rjmp EXIT_INTERRUPT	panggil label EXIT_INTERRUPT apabila bit 0 pada variabel count_libur adalah 1, jika 0 line ini dilewati
inc count_jadwal	increment variabel count_jadwal (count_jadwal + 1)
inc persiapan	increment variabel persiapan (persiapan + 1)
cpi count_jadwal, 3	cek nilai count_jadwal = 3

breq RESET

brne EXIT_INTERRUPT

RESET:

clr count_jadwal

rjmp EXIT_INTERRUPT

jika nilai count_jadwal = 3, masuk ke RESET

jika nilai count_jadwal != 3, masuk ke EXIT_INTERRUPT

reset count_jadwal apabila sudah mencapai 3

clear nilai count_jadwal

lompat ke EXIT_INTERRUPT

```
INTERRUPT_LIBUR:
    ldi count_libur, 10
    inc libur_flag
    rjmp EXIT_INTERRUPT
```

```
INTERRUPT_DEFENSE:
    inc defense_flag
    clr count_jadwal
    sbrc defense_flag, 1
    clr defense_flag
    rjmp EXIT_INTERRUPT
```

C:\Users\ceme\OneDrive\Documents\S1\Sem 2\DDAK\7\Lab7_TOP_B_ClementSamuelMarly_2206082114_Genap.asm

INTERRUPT_LIBUR:

ldi count_libur, 10

inc libur_flag

rjmp EXIT_INTERRUPT

set nilai count_libur = 10

increment nilai libur_flag (libur_flag + 1)

lompat ke EXIT_INTERRUPT

INTERRUPT_DEFENSE:

inc defense_flag

clr count_jadwal

sbrc defense_flag, 1

clr defense_flag

increment nilai defense_flag (defense flag + 1)

clear nilai count_jadwal

cek bit 1 pada variabel count_libur

clear nilai defense_flag apabila bit 1 pada variable count_libur adalah 1,
jika 0 line ini dilewati

rjmp EXIT_INTERRUPT

lompat ke EXIT_INTERRUPT