

# **LAMPIRAN C**

## **Perangkat Lunak**

• **Rule-rule *Fuzzy Logic Control* untuk kendaraan yang mengikuti suatu lintasan tertentu dengan software Petrafuz.**

1. IF error IS NB AND derror IS NB THEN selisihgain IS PB
2. IF error IS NB AND derror IS NS THEN selisihgain IS PS
3. IF error IS NB AND derror IS Z THEN selisihgain IS PS
4. IF error IS NB AND derror IS PS THEN selisihgain IS PB
5. IF error IS NB AND derror IS PB THEN selisihgain IS PB
6. IF error IS NS AND derror IS NB THEN selisihgain IS PS
7. IF error IS NS AND derror IS NS THEN selisihgain IS PS
8. IF error IS NS AND derror IS Z THEN selisihgain IS PB
9. IF error IS NS AND derror IS PS THEN selisihgain IS PB
10. IF error IS NS AND derror IS PB THEN selisihgain IS PB
11. IF error IS Z AND derror IS NB THEN selisihgain IS PB
12. IF error IS Z AND derror IS NS THEN selisihgain IS PS
13. IF error IS Z AND derror IS Z THEN selisihgain IS Z
14. IF error IS Z AND derror IS PS THEN selisihgain IS NS
15. IF error IS Z AND derror IS PB THEN selisihgain IS NB
16. IF error IS PS AND derror IS NB THEN selisihgain IS NS
17. IF error IS PS AND derror IS NS THEN selisihgain IS NS
18. IF error IS PS AND derror IS Z THEN selisihgain IS NB
19. IF error IS PS AND derror IS PS THEN selisihgain IS NS
20. IF error IS PS AND derror IS PB THEN selisihgain IS NB
21. IF error IS PB AND derror IS NB THEN selisihgain IS NB
22. IF error IS PB AND derror IS NS THEN selisihgain IS NS
23. IF error IS PB AND derror IS Z THEN selisihgain IS NB
24. IF error IS PB AND derror IS PS THEN selisihgain IS NS
25. IF error IS PB AND derror IS PB THEN selisihgain IS NB

- **Program Mikrokontroler**

Perintah macro pada file avr.inc

```
;-----;
; Load/store word dari/ke memori
;
;   ldsw  Z,mem
;   ldiw  Z,imm
.macro   ldiw
        ldi   @0L,low(@1)
        ldi   @0H,high(@1)
.endm
.macro   ldsw
        lds   @0L,@1
        lds   @0H,@1+1
.endm
.macro   lddw
        ldd   @0L,@1
        ldd   @0H,@1+1
.endm
.macro   stsw
        sts   @0,@1L
        sts   @0+1,@1H
.endm
.macro   stdw
        std   @0,@1L
        std   @0+1,@1H
.endm
.macro   pushw
        push  @0H
        push  @0L
.endm
```

```

    .macro    popw
        pop    @0L
        pop    @0H
    .endm

;-----;
; Operasi Word dengan dua register
;

    .macro    addiw
        subi    @0L,low(-(@1))
        sbci    @0H,high(-(@1))
    .endm

    .macro    subiw
        subi    @0L,low(@1)
        sbci    @0H,high(@1)
    .endm

    .macro    addw
        add    @0L,@1L
        adc    @0H,@1H
    .endm

    .macro    adcw
        adc    @0L,@1L
        adc    @0H,@1H
    .endm

    .macro    subw
        sub    @0L,@1L
        sbc    @0H,@1H
    .endm

    .macro    sbcw
        sbc    @0L,@1L
        sbc    @0H,@1H
    .endm

```

```
.macro    cpw
    cp    @0L,@1L
    cpc   @0H,@1H
.endm

.macro    cpcw
    cpc   @0L,@1L
    cpc   @0H,@1H
.endm

.macro    andw
    and   @0L,@1L
    and   @0H,@1H
.endm

.macro    andiw
    andi  @0L,low(@1)
    andi  @0H,high(@1)
.endm

.macro    orw
    or    @0L,@1L
    or    @0H,@1H
.endm

.macro    oriw
    ori   @0L,low(@1)
    ori   @0H,high(@1)
.endm

.macro    lslw
    lsl   @0L
    rol   @0H
.endm

.macro    lsrw
    lsr   @0H
    ror   @0L
.endm
```

```
.macro    asrw
    asr   @0H
    ror   @0L
.endm

.macro    rolw
    rol   @0L
    rol   @0H
.endm

.macro    rorw
    ror   @0H
    ror   @0L
.endm

.macro    clrw
    clr   @0L
    clr   @0H
.endm

.macro    comw
    com   @0L
    com   @0H
.endm

.macro    movew
    mov   @0L, @1L
    mov   @0H, @1H
.endm

.macro    tstw
    cp    @0L, _0
    cpc   @0H, _0
.endm

.macro    outw
    out   @0H, @1H
    out   @0L, @1L
.endm
```

```
.macro    inw
    in    @0L, @1L
    in    @0H, @1H
.endm

;-----;
; Output port immediate via r16
;    outi  port,var
.macro    outi
    ldi   r16,@1
    out   @0,r16
.endm

;-----;
; Add immediate to register
.macro    addi
    subi  @0,-(@1)
.endm
```

## Program Utama

```
-----;  
; Program Line Follower Robot  
; Internal Oscillator 8 Mhz, dengan low fuse bit untuk bit CKSEL 01000  
dan SUT: 10 (start up time 65ms);  
; Mode sleep yang digunakan power-down, dan ;  
; Timer0 dan Timer2 dipakai dalam mode PWM ;  
-----;
```

```
.include "m16def.inc"  
.include "avr.inc"  
.equ  RAMTOP = 0x60  
.equ  bit0   = 0b00000001  
.equ  bit1   = 0b00000010  
.equ  bit2   = 0b00000100  
.equ  bit3   = 0b00001000  
.equ  bit4   = 0b00010000  
.equ  bit5   = 0b00100000  
.equ  bit6   = 0b01000000  
.equ  bit7   = 0b10000000  
  
.def  simpan1 = r16  
.def  simpan2 = r17  
.def  simpan3 = r21  
.def  err     = r25  
.def  AL     = r18  
.def  AH     = r19  
.def  speed  = r20  
.def  positif = r22  
.def  Esblm  = r29  
.def  E      = r30  
.def  De     = r31
```

```

.equ setpoint = 0b00110011
.equ batasns = 0b00000110
.equ batasnb = 0b00001001
.equ batasps = 0b00001100
.equ bataspb = 0b00010100
.equ batasz = 0x00
.equ gainpb = 128
.equ gainps = 64
.equ gainz = 0
.equ gainns = 64
.equ gainnb = 128
.equ batasdens = 0b00000110
.equ batasdenb = 0b00001001
.equ batasdeps = 0b00001100
.equ batasdepb = 0b00010100
.equ batasdez = 0x00

;-----;
;                Tabel Interupsi                ;
;-----;

.cseg
    jmp RESET        ; interupt RESET
;   jmp 0            ; interupt INT0
;   jmp 0            ; interupt INT1
;   jmp 0            ; interupt TIMER2_COMP
;   jmp 0            ; interupt TIMER2_OVF
;   jmp 0            ; interupt TIMER1_CAPT
;   jmp 0            ; interupt TIMER1_COMPA
;   jmp 0            ; interupt TINER1_COMPB
;   jmp 0            ; interupt TIMER1_OVF
;   jmp TIM0_OVF     ; interupt TIMER0_OVF
;   jmp 0            ; interupt SPI_STC

```

```

;      jmp 0          ; interupt USART_RXC
;      jmp 0          ; interupt USART_UDRE
;      jmp 0          ; interupt USART_TXC
;      jmp 0          ; interupt ADC
;      jmp 0          ; interupt EE_RDY
;      jmp 0          ; interupt ANA_COMP
;      jmp 0          ; interupt TWI
;      jmp 0          ; interupt INT2
;      jmp 0          ; interupt TIMER2_COMP
;      jmp 0          ; interupt SPM_RDY

```

```

;-----;

```

```

;-----;

```

```

;      POWER ON RESET/ POWER ON      ;

```

```

;-----;

```

```

RESET      :

```

```

    ldiw A, RAMEND    ; inialisasi Stack Pointer

```

```

    outw SP, A        ;

```

```

    clr err

```

```

    outi DDRA, 0b00000000 ; inialisasi PORTA

```

```

    outi PORTA, 0b00000000 ; /

```

```

    outi DDRB, 0b01101011 ; Inialisasi PORTB

```

```

    outi PORTB, 0b10010100 ; /

```

```

    outi DDRC, 0b11111111 ; Inialisasi PORTC

```

```

    outi PORTC, 0b00000000 ; /

```

```

    outi DDRD, 0b11111111 ; Inialisasi PORTD

```

```

    outi PORTD, 0b01111111 ; /

```

```

    ldi Esblm, 0b00000000 ; inialisasi Emin awal

```

```
        outi TCCR0, 0b01100101      ; Inisialisasi TIMER 0 untuk motor
kiri
        outi TCCR2, 0b01100111      ; Inisialisasi Timer 2 untuk motor
kanan
```

```
        outi OCR0, 0x7F
        outi OCR2, 0x7F
        clr positif
        clr speed
        outi PORTB, 0b0100010 ; start motor kiri dan kanan
        call delay
        sei
```

```
;-----;
;   Program Utama      ;
;-----;
```

start:

```
        rcall ceksensor
        cpi AL, 0x3F
        breq cekerr
        ldi simpan3, setpoint
        mov simpan1, AL
        cp simpan3, simpan1
        brsh pos
        brlo negatif
```

ceksensor:

```
        outi PORTD, 0b01101010 ; pemancar sensor 1,3 dan 5 aktif
        sbi PORTA, 6           ; penerima aktif
        rcall wait
        in AL, PORTA           ; baca hasil di portA
        mov AH, AL
```

```
    outi PORTA, 0b01010101 ; pemancar sensor 2,4 dan 6 aktif
    sbi PORTA, 6           ; penerima aktif
    rcall wait
    in AL, PORTA
    or AL, AH
    ret
```

cekerr:

```
    inc err
    cpi err, 0x04
    breq enter_SLEEP
    rcall ceksensor
    cpi AL, 0x3F
    breq cekerr
    ret
```

enter\_SLEEP :

```
    outi PORTA, 0b00000000
    outi PORTB, 0b00100010
    outi PORTC, 0b00000000
    outi PORTD, 0b01111111
    in r16, SREG
    clr positif
    outi MCUCR, 0b10110011
    sleep
    ret
```

terus:

```
    mov Esblm, E
    rcall delay
    jmp start
```

pos:

```
sbr positif, 0
sub simpan3, simpan1
mov E, simpan3
cpi E, bataspb
brsh epb
brlo cekeps
```

epb:

```
rcall pb
rjmp terus
```

cekeps:

```
cpi E, batasps
brsh eps
brlo ez
```

eps:

```
rcall ps
rjmp terus
```

negatif:

```
cbr positif, 0
mov simpan2, simpan1
sub simpan2, simpan3
mov E, simpan2
cpi E, batasnb
brsh enb
brlo cekens
```

enb:

```
rcall nb
rjmp terus
```

cekens:

```
cpi E, batasns
brsh ens
brlo ez
```

ens:

```
rcall ns
rjmp terus
```

ez:

```
rcall zero
rjmp terus
```

```
;-----;
```

```
;   Positif Big   ;
```

```
;-----;
```

pb:

```
mov simpan3, E
cp simpan3, Esblm
brsh positifpb
brlo negatifpb
```

positifpb:

```
sub simpan3, Esblm
mov De, simpan3
cpi De, batasdepb
brsh depb_pb
brlo cekdepb_ps
```

negatifpb:

```
mov simpan2, Esblm
sub simpan2, simpan3
mov De, simpan2
cpi De, batasdenb
brsh depb_nb
brlo depb_ns
```

depb\_pb:

```
ldi simpan1, gainnb
rcall kurangkec
ret
```

```

cekdepb_ps:
    cpi E, batasdeps
    brsh depb_ps
    brlo depb_z
depb_ps:
    ldi simpan1, gainns
    rcall kurangkec
    ret
depb_z:
    ldi simpan1, gainnb
    rcall kurangkec
    ret
depb_nb:
    ldi simpan1, gainnb
    rcall kurangkec
    ret
cekdepb_ns:
    cpi E, batasdens
    brsh depb_ns
    brlo depb_z
depb_ns:
    ldi simpan1, gainnb
    rcall kurangkec
    ret
;-----;
;    Positif Small ;
;-----;
ps:
    mov simpan3, E
    cp simpan3, Esblm
    brsh positifps
    brlo negatifps

```

positifps:

```
sub simpan3, Esblm
mov De, simpan3
cpi De, batasdepb
brsh deps_pb
brlo cekdeps_ps
```

negatifps:

```
mov simpan2, Esblm
sub simpan2, simpan3
mov De, simpan2
cpi De, batasdenb
brsh deps_nb
brlo deps_ns
```

deps\_pb:

```
ldi simpan1, gainnb
rcall kurangkec
ret
```

cekdeps\_ps:

```
cpi E, batasdeps
brsh deps_ps
brlo deps_z
```

deps\_ps:

```
ldi simpan1, gainns
rcall kurangkec
ret
```

deps\_z:

```
ldi simpan1, gainnb
rcall kurangkec
ret
```

deps\_nb: ldi simpan1, gainns

```
rcall kurangkec
ret
```

cekdeps\_ns:

    cpi E, batasdens

    brsh deps\_ns

    brlo deps\_z

deps\_ns:

    ldi simpan1, gainns

    rcall kurangkec

    ret

;-----;

;    Z            ;

;-----;

zero:

    mov simpan3, E

    cp simpan3, Esblm

    brsh positifz

    brlo negatifz

positifz:

    sub simpan3, Esblm

    mov De, simpan3

    cpi De, batasdepb

    brsh dez\_pb

    brlo cekdez\_ps

negatifz:

    mov simpan2, Esblm

    sub simpan2, simpan3

    mov De, simpan2

    cpi De, batasdenb

    brsh dez\_nb

    brlo dez\_ns

```
dez_pb:
    ldi simpan1, gainnb
    rcall kurangkec
    ret
```

```
cekdez_ps:
    cpi E, batasdeps
    brsh dez_ps
    brlo dez_z
```

```
dez_ps:
    ldi simpan1, gainns
    rcall kurangkec
    ret
```

```
dez_z:
    ldi simpan1, gainz
    rcall tetap
    ret
```

```
tetap:
    ldi simpan1, gainz
    rjmp tambahkec
    ret
```

```
dez_nb:
    ldi simpan1, gainpb
    rcall tambahkec
    ret
```

```
cekdez_ns:
    cpi E, batasdens
    brsh dez_ns
    brlo dez_z
```

```
dez_ns:
    ldi simpan1, gainps
    rcall tambahkec
    ret
```

```
;-----;  
;      Negatif Big  ;  
;-----;
```

nb:

```
    mov simpan3, E  
    cp simpan3, Esblm  
    brsh positifnb  
    brlo negatifnb
```

positifnb:

```
    sub simpan3, Esblm  
    mov De, simpan3  
    cpi De, batasdepb  
    brsh denb_pb  
    brlo cekdenb_ps
```

negatifnb:

```
    mov simpan2, Esblm  
    sub simpan2, simpan3  
    mov De, simpan2  
    cpi De, batasdenb  
    brsh denb_nb  
    brlo denb_ns
```

denb\_pb:

```
    ldi simpan1, gainpb  
    rcall tambahkec  
    ret
```

cekdenb\_ps:

```
    cpi E, batasdeps  
    brsh denb_ps  
    brlo denb_z
```

```

denb_ps:
    ldi simpan1, gainpb
    rcall tambahkec
    ret

denb_z:
    ldi simpan1, gainps
    rcall tambahkec
    ret

denb_nb:
    ldi simpan1, gainpb
    rcall tambahkec
    ret

cekdenb_ns:
    cpi E, batasdens
    brsh denb_ns
    brlo denb_z

denb_ns:
    ldi simpan1, gainps
    rcall tambahkec
    ret

;-----;
;    Negatif Small;
;-----;

ns:
    mov simpan3, E
    cp simpan3, Esblm
    brsh positifns
    brlo negatifns

positifns:
    sub simpan3, Esblm
    mov De, simpan3
    cpi De, batasdepb

```

```
brsh dens_pb  
brlo cekdens_ps
```

negatifns:

```
mov simpan2, Esblm  
sub simpan2, simpan3  
mov De, simpan2  
cpi De, batasdenb  
brsh dens_nb  
brlo dens_ns
```

dens\_pb:

```
ldi simpan1, gainpb  
rcall tambahkec  
ret
```

cekdens\_ps:

```
cpi E, batasdeps  
brsh dens_ps  
brlo dens_z
```

dens\_ps:

```
ldi simpan1, gainpb  
rcall tambahkec  
ret
```

dens\_z:

```
ldi simpan1, gainpb  
rcall tambahkec  
ret
```

dens\_nb:

```
ldi simpan1, gainps  
rcall tambahkec  
ret
```

cekdens\_ns:

```
cpi E, batasdens  
brsh dens_ns
```

brlo dens\_z

dens\_ns:

ldi simpan1, gainps

rcall tambahkec

ret

tambahkec:

in speed, OCR0

add speed, simpan1

out OCR0, speed

ret

kurangkec:

in speed, OCR0

sub speed, simpan1

out OCR0, speed

ret

delay :

rcall wait

rcall wait

rcall wait

ret

clockwait:

push r18

ldi r18, 0x40

loncat:

dec r18

cpi r18, 0x00

brne loncat

pop r18

ret

wait :

push r16

ldi R16, 0xFF

loncat2:

dec r16

rcall clockwait

cpi r16, 0x00

brne loncat2

pop r16

ret