LAMPIRAN B

LISTING PROGRAM

```
;=================================
;         Program Tampilan LCD
;=================================
;VARIABEL:

          command     equ     31h
        character   equ     32h
        D_BIN       equ     33h
          ASCH        equ     34h
          ASCL        equ     35h
        L_DL0       equ     36h
        L_DL1       equ     37h
        L_DL2       equ     38h
        S_DL0       equ     39h
        S_DL1       equ     3Ah
        S_DL2       equ     3Bh
        RS          equ     P2.4
        EN          equ     P2.5
        LCD         equ     P2
        WR_ADC1             equ     P0.1
        WR_ADC2     equ     P0.3
        INTR_ADC1   equ     P0.0
        INTR_ADC2   equ     P0.2
        jeruk       equ     P0.7
        status1     equ     42h
        status2     equ     43h

;=================================
;         Subrutin Baris ke-1
;=================================

line1:    mov    command,#80h
          lcall  send_command
          mov    r7,#10h
next1:    mov    a,#0h
          movc   a,@a+dptr
          mov    character,a
          lcall  send_character
          inc    dptr
          djnz   r7,next1
          ret
line1B:   mov    command,#85h
          lcall  send_command
          mov    r7,#11
next1B:   mov    a,#0h
          movc   a,@a+dptr
```

```
        mov    character,a
        lcall  send_character
        inc    dptr
        djnz   r7,next1B
        ret

;=================================
;         Subrutin Baris ke-2
;=================================

line2:    mov    command,#0C0h
        lcall  send_command
        mov    r7,#10h
next2:    mov    a,#0h
        movc   a,@a+dptr
        mov    character,a
        lcall  send_character
        inc    dptr
        djnz   r7,next2
        ret
line2B:   mov    command,#0C5h
        lcall  send_command
        mov    r7,#11
next2B:   mov    a,#0h
        movc   a,@a+dptr
        mov    character,a
        lcall  send_character
        inc    dptr
        djnz   r7,next2B
        ret

;=================================
;         Subrutin Inisialisasi LCD
;=================================

init_LCD:
           lcall  SDLY
        lcall  SDLY
        MOV    LCD,#02H;set mode operasi LCD 4bit

        setb   EN
        mov    S_DL0,#00h;delay power on LCD
        MOV    S_DL1,#4h
        lcall  SDLY
        clr    EN

        mov    command,#28h;set mode LCD 16 karakter 2 baris
```

```
        lcall  send_command
        lcall  sdly
        mov    command,#08h;turn off LCD
        lcall  send_command
        lcall  sdly
        mov    command,#0Ch;turn on LCD
        lcall  send_command
        lcall  sdly
        mov    command,#06h;increment address mode
        lcall  send_command
        lcall  sdly
        ret

;=====================================
;         Subrutin Tulis Perintah ke LCD
;=====================================

send_command:

        mov    LCD,#00H
        clr    RS
        mov    a,command
        anl    a,#0f0h
        swap   a
        mov    LCD,a
        setb   EN
        mov    S_DL1,#1h
        lcall  SDLY
        clr    EN
        mov    a,command
        anl    a,#0fh
        mov    LCD,A
        setb   EN
        mov    S_DL1,#1h
        lcall  SDLY
        clr    EN
        ret

;=====================================
;         Subrutin Tulis Karakter ke LCD
;=====================================

send_character:
        mov    LCD,#00H
        mov    a,character
        anl    a,#0F0h
        swap   a
```

```
        mov    LCD,a
        setb   RS
        setb   EN
        mov    S_DL1,#1h
        lcall  SDLY
        clr    EN
        mov    a,character
        anl    a,#0fh
        mov    LCD,a
        setb   RS
        setb   EN
        mov    S_DL1,#1h
        lcall  SDLY
        clr    EN
        ret

;=======================================
;         Subrutin Short Delay
;=======================================

SDLY:
      djnz   S_DL0,SDLY
      djnz   S_DL1,SDLY
      ret

;=======================================
;         Subrutin Long Delay
;=======================================
LDLY:
      djnz   L_DL0,LDLY
      djnz   L_DL1,LDLY
      djnz   L_DL2,LDLY
      ret

;=============================================
;         Subrutin Tulis Karakter ASCII ke LCD
;=============================================

BIN2ASC:   MOV     A,D_BIN
        ANL     A,#0FH
        ADD     A,#0F6H
        JNC     NOADJL
        ADD     A,#07H
NOADJL:    ADD     A,#3AH
        MOV     ASCL,A
        MOV     A,D_BIN
        SWAP    A
```

```
        ANL     A,#0FH
        ADD     A,#0F6H
        JNC     NOADJL2
        ADD     A,#07H
NOADJL2:   ADD     A,#3AH
        MOV     ASCH,A
        RET
;=====================================
;         Subrutin Delay 50 MiliDetik
;=====================================
delay_50_ms:MOV R5,#50
D50MS2: MOV R6,#04H
D50MS3: MOV R7,#0FFH
            DJNZ R7,$
         DJNZ R6,D50MS3
         DJNZ R5,D50MS2
         RET
;=====================================
;         Subrutin Delay 1 Detik
;=====================================
delay_1_s:    lcall    delay_50_ms
              lcall    delay_50_ms
              lcall    delay_50_ms
              lcall    delay_50_ms
              lcall    delay_50_ms
              lcall    delay_50_ms
              lcall    delay_50_ms
              lcall    delay_50_ms
              lcall    delay_50_ms
              lcall    delay_50_ms
              lcall    delay_50_ms
              lcall    delay_50_ms
              lcall    delay_50_ms
              lcall    delay_50_ms
              lcall    delay_50_ms
              lcall    delay_50_ms
              lcall    delay_50_ms
              lcall    delay_50_ms
              lcall    delay_50_ms
              lcall    delay_50_ms
              lcall    delay_50_ms
              ret
```

```
;=================================================
        org     00h           ; alamat awal 00
        ljmp    main_prog
; =================================================
;                 MAIN PROGRAM
; =================================================
        org     100h          ; alamat program

main_prog:

           lcall   delay_1_s;delay
        lcall   init_lcd;inisialisasi LCD

ulang:
        mov     dptr,#pesan1
        acall   line1
        mov     dptr,#pesan2
        acall   line2
        lcall   delay_1_s
        lcall   delaY_1_s
        lcall   delay_1_s
        mov     dptr,#blank
        acall   line1
        mov     dptr,#blank
        acall   line2


        mov     dptr,#pesan3
        acall   line1
        mov     dptr,#pesan4
        acall   line2
        lcall   delay_1_s
        lcall   delaY_1_s
        lcall   delay_1_s

           mov     dptr,#blank
        acall   line1
        mov     dptr,#blank
        acall   line2

           mov     p1,#0FFh
           mov     p3,#0FFh
           mov     p0,#0FFh

loop:
          jnb     jeruk,buah_jeruk   ; P0.7
          ljmp    deteksi_buah_tomat
```

```
buah_jeruk:
          mov     dptr,#jerruk
        lcall    line1B
        ljmp    back


back:
           clr     WR_ADC1            ; WRITE (LOW)
           nop
           setb    WR_ADC1            ; WRITE (High)
here:     jb      INTR_ADC1,here      ; INTR

           mov     D_BIN,P1
           lcall   BIN2ASC
         mov     command,#080h
         lcall   send_command
         mov     character,ASCH
         lcall   send_character

         mov     character,ASCL
         lcall   send_character


           lcall    delay_1_s



; baca dulu ADC

back1:     clr    WR_ADC2             ; WRITE (LOW)
           nop
           setb    WR_ADC2            ; WRITE (High)
here1:     jb     INTR_ADC2,here1      ; INTR

           mov     D_BIN,P3
           lcall   BIN2ASC

         mov     command,#0C0h
         lcall   send_command
         mov     character,ASCH
         lcall   send_character

         mov     character,ASCL
         lcall   send_character
           lcall   delay_1_s
```

```
          mov     A,P3
          cjne    A,#0AFh,deteksi       ; isi

; tampil _kosong
          mov    dptr,#kosong
          lcall   line2B
          lcall   delay_1_s
          ljmp   loop

          clr   c

deteksi:
         subb    A,#0AAh
         jc      status1_matang                ; carry = matang
      ljmp    status1_tdk_matang                ; no carry = blm matang


status1_matang:
         mov    status1,#'M'
         ljmp   cekport3

status1_tdk_matang:
      mov   status1,#'B'
      ljmp  cekport3


cekport3:
          mov     A,P1
          cjne    A,#0ADh,cekport4      ; isi

; tampil _kosong
          mov    dptr,#kosong
          lcall   line2B
          lcall   delay_1_s
          ljmp   loop
          clr   c

cekport4:
         subb    A,#0A9h
         jc      status2_matang                ; carry = matang
      ljmp    status2_tdk_matang                ; no carry = blm matang

status2_matang:
         mov    status2,#'M'
         ljmp   cek_status
```

```
status2_tdk_matang:
      mov   status2,#'B'
      ljmp   cek_status

cek_status:
         mov   A,status1
         cjne  A,#'M',status1tdkmatang          ; periksa status2
      ljmp  statusmatang


status1tdkmatang:
      mov  A,status2
      cjne  A,#'M',tampilkan_blm_matang
      ljmp  tampilkan_setengah_matang

statusmatang:
      mov  A,status2
      cjne  A,#'M',tampilkan_setengah_matang
      ljmp  tampilkan_matang

;=====================================
;        Subrutin Matang
;=====================================

tampilkan_matang:
         mov    dptr,#matang
         lcall   line2B
         lcall   delay_1_s
         ljmp   loop

;=======================================
;        Subrutin Tampilkan Belum Matang
;=======================================

tampilkan_blm_matang:

         mov    dptr,#bmatang
         lcall   line2B
         lcall   delay_1_s
         ljmp   loop
;=========================================
;        Subrutin Tampilkan Setengah Matang
;=========================================
tampilkan_setengah_matang:
         mov    dptr,#smatang
         lcall   line2B
```

```
        lcall   delay_1_s
        ljmp   loop

;=====================================
;       Deteksi Buah Tomat
;=====================================

deteksi_buah_tomat:

      mov    dptr,#tomat
      lcall   line1B
                    ; buah tomat


backt:
         clr    WR_ADC1        ; WRITE (LOW)
         nop
         setb   WR_ADC1        ; WRITE (High)
heret:   jb     INTR_ADC1,heret     ; INTR

         mov    D_BIN,P1
         lcall  BIN2ASC
      mov    command,#080h
      lcall  send_command
      mov    character,ASCH
      lcall  send_character

      mov    character,ASCL
      lcall  send_character


         lcall   delay_1_s



; baca dulu ADC

back1t:    clr   WR_ADC2        ; WRITE (LOW)
         nop
         setb   WR_ADC2        ; WRITE (High)
here1t:   jb    INTR_ADC2,here1t     ; INTR

         mov    D_BIN,P3
         lcall  BIN2ASC

      mov    command,#0C0h
      lcall  send_command
```

```
        mov     character,ASCH
        lcall   send_character

        mov     character,ASCL
        lcall   send_character
          lcall   delay_1_s


          mov     A,P3
          cjne    A,#0AFh,deteksit        ; isi

; tampil _kosong
          mov    dptr,#kosong
          lcall   line2B
          lcall   delay_1_s
          ljmp   loop

          clr   c

deteksit:
         subb    A,#0A5h
         jnc      status1_matangt                ; carry = matang
      ljmp    status1_tdk_matangt                ; no carry = blm matang


status1_matangt:
         mov    status1,#'M'
         ljmp   cekport3t

status1_tdk_matangt:
      mov   status1,#'B'
      ljmp  cekport3t


cekport3t:
          mov     A,P1
          cjne    A,#0ADh,cekport4t      ; isi

; tampil _kosong
          mov    dptr,#kosong
          lcall   line2B
          lcall   delay_1_s
          ljmp   loop
          clr   c

cekport4t:
         subb    A,#0A9h
```

```
          jnc      status2_matangt                ; carry = matang
       ljmp    status2_tdk_matangt                ; no carry = blm matang

status2_matangt:
          mov    status2,#'M'
          ljmp   cek_statust

status2_tdk_matangt:
       mov   status2,#'B'
       ljmp   cek_statust

cek_statust:
          mov   A,status1
          cjne  A,#'M',status1tdkmatangt         ; periksa status2
       ljmp  statusmatangt

status1tdkmatangt:
       mov  A,status2
       cjne  A,#'M',tbm
       ljmp  tampilkan_setengah_matang

statusmatangt:
       mov  A,status2
       cjne  A,#'M',tsm
       ljmp  tampilkan_matang
tsm:
          ljmp   tampilkan_setengah_matang
tbm:    ljmp   tampilkan_blm_matang

org          500h

pesan1:    db    ' DETEKSI BUAH   '
pesan2:    db    ' MATANG / BLM   '
pesan3:    db    '  JOHAN HALIM   '
pesan4:    db    '  NRP : 0322103 '
blank:     db   '                '
jerruk:    db   'BUAH JERUK '
tomat:     db   'BUAH TOMAT '
matang:    db   'M A T A N G'
bmatang:   db   'BLM MATANG '
smatang:   db   '1/2 MATANG '
kosong:    db   'K O S O N G'
isi:       db   '  I S I    '
end
```