

LAMPIRAN A

```
INCLUDE    "8051.H"

PENGATUR1  .EQU  P1.6      ;INPUT SWITCH
PENGATUR2  .EQU  P1.7
KONTROL1   .EQU  P3.6      ;MOTOR
KONTROL2   .EQU  P3.7      ;SEGMENT
DATABUS    .EQU  P0
SW1        .EQU  P1.3
SW2        .EQU  P1.4
SW3        .EQU  P1.5
SLANTAI1   .EQU  P1.0
SLANTAI2   .EQU  P1.1
SLANTAI3   .EQU  P1.2
SPINTUBUKA .EQU  P1.6
SPINTUTUTUP .EQU  P1.7
CTULIS     .EQU  10100000B ;WRITE DATA COMMAND Note 3
CBACA      .EQU  10100001B ;READ DATA COMMAND Note 3
CTEEP      .EQU  10100000B ;ALAMAT EEP
CBEEP      .EQU  10100001B ;
CTRTC      .EQU  11010000B ;WRITE DATA COMMAND Note 3
CBRTC      .EQU  11010001B ;READ DATA COMMAND Note 3
ADDRH      .EQU  0
ADDRL      .EQU  0
BYTCNT     .EQU  8
SDA        .EQU  P3.3
SCK        .EQU  P3.2

SWITCH     .EQU  P2
Y1         .EQU  P2.0
Y2         .EQU  P2.1
Y3         .EQU  P2.2
Y4         .EQU  P2.3
```

```
X1      .EQU  P2.4
X2      .EQU  P2.5
X3      .EQU  P2.6
```

```
        .ORG  $30
MENITLAMA  .BLOCK 1
DETIKLAMA  .BLOCK 1
DETIK      .BLOCK 1
MENIT      .BLOCK 1
JAM        .BLOCK 1
HARI       .BLOCK 1
TANGGAL    .BLOCK 1
BULAN      .BLOCK 1
TAHUN      .BLOCK 1
ADDR_WI2C  .BLOCK 1
ADDR_RI2C  .BLOCK 1
DATAKEY    .BLOCK 1
POSISI     .BLOCK 1
STATUSSER  .BLOCK 1
JUM_LANTAI1 .BLOCK 1
JUM_LANTAI2 .BLOCK 1
JUM_LANTAI3 .BLOCK 1
JUM_MENIT  .BLOCK 1
BUFFER     .BLOCK 8
PRIORITAS  .BLOCK 1
STLTPOS    .BLOCK 1
```

```
        .ORG  $0
        LJMP  MULAI
```

```
        .ORG  $100
MULAI:  MOV   SP,#$20
        MOV  PSW,#$00
        CLR  KONTROL1
        CLR  KONTROL2
        LCALL MOTOROFFSEMUA
```

```
LCALL DELAYSW
MOV  STATUSSER,#0
MOV  POSISI,#0
LCALL MOTOROFFSEMUA
LCALL SEGMENTMIN
```

```
LCALL INIT_SERIAL
MOV  A,#'A'
LCALL SENDCHR
MOV  A,#'B'
LCALL SENDCHR
MOV  A,#'C'
LCALL SENDCHR
MOV  A,#'D'
LCALL SENDCHR
MOV  A,#$0A
LCALL SENDCHR
MOV  A,#$0D
LCALL SENDCHR
```

```
LCALL MOTORNAIK
LCALL DELAY10X
LCALL POSISIRESET
LCALL SEGMENT1
MOV  POSISI,#1
MOV  A,#'1'
LCALL SENDCHR
MOV  A,#$0D
LCALL SENDCHR
LCALL DELAYSW
LCALL PINTUBUKATUTUP
```

```
MOV  JUM_LANTAI1,#0
MOV  JUM_LANTAI2,#0
MOV  JUM_LANTAI3,#0
MOV  JUM_MENIT,#0
```

```
MOV  PRIORITAS,#0
MOV  STLTPOS,#1
MOV  ADDR_WI2C,#CTRTC
MOV  ADDR_RI2C,#CBRTC
```

LOOP0:

```
LOOP:  MOV  SP,#$20
```

EXEC1:

```
JB  SW1,EXEC2
MOV  SP,#$20
MOV  A,JUM_LANTAI1
ADD  A,#1
DA  A
MOV  JUM_LANTAI1,A
LJMP SW1AKTIF
```

EXEC2:

```
JB  SW2,EXEC3
MOV  SP,#$20
MOV  A,JUM_LANTAI2
ADD  A,#1
DA  A
MOV  JUM_LANTAI2,A
LJMP SW2AKTIF
```

EXEC3:

```
JB  SW3,EXECPASS
MOV  SP,#$20
MOV  A,JUM_LANTAI3
ADD  A,#1
DA  A
MOV  JUM_LANTAI3,A
LJMP SW3AKTIF
```

EXECPASS:

;----PENGECEKAN JAM PRIORITAS-----

CEKJAMPAGI:

```
MOV  A,JAM
```

```

    CJNE A,#$07,CEKJAMPAGI1
    MOV  PRIORITAS,#'T'    ;TURUN KE LT.1
    LCALL PRIORITASTURUN
    LJMP CEKJAMPASS
CEKJAMPAGI1: CJNE A,#$08,CEKJAMSORE
    MOV  A,MENIT
    CJNE A,#$00,CEKJAMSORE
    MOV  PRIORITAS,#'T'    ;
    LCALL PRIORITASTURUN
    LJMP CEKJAMPASS
CEKJAMSORE: MOV  A,JAM
    CJNE A,#$16,CEKJAMSORE1
    MOV  PRIORITAS,#'A'
    LCALL PRIORITASNAIK
    LJMP CEKJAMPASS
CEKJAMSORE1: CJNE A,#$17,CEKJAMEND
    MOV  A,MENIT
    CJNE A,#$00,CEKJAMEND
    MOV  PRIORITAS,#'A'
    LCALL PRIORITASNAIK
    LJMP CEKJAMPASS
CEKJAMEND:  MOV  PRIORITAS,#'N'

CEKJAMPASS:

;-----PENGECEKAN PENEKANAN KEYPAD
    LCALL SCANNINGKEYPAD
    CJNE A,#$0C,BUKANSETJAM
    LCALL SEGMENTBLANK
    LCALL DELAYSW
    LCALL SEGMENTMIN
    LCALL DELAYSW
    LCALL SEGMENTBLANK
    LCALL DELAYSW
    LCALL SEGMENTMIN
    LCALL DELAYSW
    LCALL SEGMENTBLANK

```

```
LCALL DELAYSW
LCALL SEGMENTMIN
MOV SP,#$20
LJMP SET_JAMTGL
```

BUKANSETJAM:

```
LCALL KIRIMJAMKEPC
```

```
MOV A,JUM_MENIT
CJNE A,#$10,HITUNGJUMMENIT
MOV JUM_MENIT,#0
```

;---MEMBANDINGKAN LANTAI PEMAKAIAN

```
LCALL PERBANDINGANJUMLAH
MOV JUM_LANTAI1,#0
MOV JUM_LANTAI2,#0
MOV JUM_LANTAI3,#0
MOV A,PRIORITAS
CJNE A,#'N',HITUNGJUMMENIT
LCALL PRIORITASNORMAL
```

HITUNGJUMMENIT:

```
MOV A,MENIT
CJNE A,MENITLAMA,TAMBAHMENIT
LJMP LOOP0
```

TAMBAHMENIT: MOV MENITLAMA,MENIT

```
MOV A,JUM_MENIT
ADD A,#$1
DA A
MOV JUM_MENIT,A
LJMP LOOP0
```

;----- JAM PRIORITAS TURUN-----

PRIORITASTURUN:

```
MOV A,POSISI
CJNE A,#1,CEKPOSTRN2
RET
```

CEKPOSTRN2: CJNE A,#2,CEKPOSTRN3

```

        LCALL TURUNLANTAI2KE1
        RET
CEKPOSTRN3:  CJNE  A,#3,CEKPOSTRNPASS
        LCALL TURUNLANTAI3KE1
CEKPOSTRNPASS: RET

;----- JAM PRIORITAS NAIK-----
PRIORITASNAIK:
        MOV  A,POSISI
        CJNE A,#3,CEKPOSNAIK2
        RET
CEKPOSNAIK2:  CJNE  A,#2,CEKPOSNAIK1
        LCALL NAIKLANTAI2KE3
        RET
CEKPOSNAIK1:  CJNE  A,#1,CEKPOSNAIKPASS
        LCALL NAIKLANTAI1KE3
CEKPOSNAIKPASS: RET

;-----PRIORITAS YG TERBANYAK-----
PRIORITASNORMAL:
        POSLT1:  MOV  A,STLTPOS
                CJNE A,#1,POSLT2
                MOV  A,POSISI
                CJNE A,#1,POS12
                RET
        POS12:   CJNE  A,#2,POS13
                LCALL TURUNLANTAI2KE1
                RET
        POS13:   CJNE  A,#3,POS13PASS
                LCALL TURUNLANTAI3KE1
        POS13PASS:  RET

        POSLT2:
                CJNE A,#2,POSLT3
                MOV  A,POSISI
                CJNE A,#2,POS21
                RET

```

```

POS21:    CJNE  A,#1,POS23
          LCALL NAIKLANTAI1KE2
          RET
POS23:    CJNE  A,#3,POS23PASS
          LCALL TURUNLANTAI3KE2
POS23PASS:  RET

```

```

POST3:
          CJNE  A,#3,POS32PASS
          MOV   A,POSISI
          CJNE  A,#3,POS31
          RET
POS31:    CJNE  A,#1,POS32
          LCALL NAIKLANTAI1KE3
          RET
POS32:    CJNE  A,#2,POS32PASS
          LCALL NAIKLANTAI2KE3
POS32PASS:  RET

```

```

;-----
PERBANDINGANJUMLAH:
          MOV   A,JUM_LANTAI1
          CJNE  A,#$00,MULAIBANDING
          MOV   A,JUM_LANTAI2
          CJNE  A,#$00,MULAIBANDING
          MOV   A,JUM_LANTAI3
          CJNE  A,#$00,MULAIBANDING
          MOV   STLTPOS,#1
          RET

```

```

MULAIBANDING:
          MOV   A,JUM_LANTAI1
          MOV   B,JUM_LANTAI2
          CJNE  A,B,CEKLBKC
          LJMP  BD1DAN3
CEKLBKC:  JC    JUM1LBKC

```


;LT1 > LT2 MAKA BANDINGKAN LT1 DGN LT3

BD1DAN3:

```
MOV  A,JUM_LANTAI1
MOV  B,JUM_LANTAI3
CJNE A,B,CEKLT1LBKC
MOV  STLTPPOS,#1
RET
```

```
CEKLT1LBKC: JC  LT1LBKC
MOV  STLTPPOS,#1
RET
```

```
LT1LBKC:  MOV  STLTPPOS,#3
RET
```

JUM1LBKC:

;LT1 < LT2 MAKA BANDINGKAN LT2 DGN LT3

BD2DAN3:

```
MOV  A,JUM_LANTAI2
MOV  B,JUM_LANTAI3
CJNE A,B,CEKLT2LBKC
MOV  STLTPPOS,#2
RET
```

```
CEKLT2LBKC: JC  LT2LBKC
MOV  STLTPPOS,#2
RET
```

```
LT2LBKC:  MOV  STLTPPOS,#3
RET
```

;-----
; PENGIRIMAN PAKET DATA KEKOMPUTER
;-----

KIRIMJAMKEPC:

```
LCALL BACA_WAKTU
MOV  A,#'J'
LCALL SENDCHR
MOV  A,JAM
LCALL ANDF0PC
MOV  A,JAM
```

```

LCALL AND0FPC
MOV A,MENIT
LCALL ANDF0PC
MOV A,MENIT
LCALL AND0FPC
MOV A,DETIK
LCALL ANDF0PC
MOV A,DETIK
LCALL AND0FPC
MOV A,#'- '
LCALL SENDCHR
MOV A,JUM_LANTAI1
LCALL ANDF0PC
MOV A,JUM_LANTAI1
LCALL AND0FPC
MOV A,JUM_LANTAI2
LCALL ANDF0PC
MOV A,JUM_LANTAI2
LCALL AND0FPC
MOV A,JUM_LANTAI3
LCALL ANDF0PC
MOV A,JUM_LANTAI3
LCALL AND0FPC
MOV A,#'- '
LCALL SENDCHR
MOV A,PRIORITAS
LCALL SENDCHR
MOV A,#'- '
LCALL SENDCHR
MOV A,JUM_MENIT
LCALL ANDF0PC
MOV A,JUM_MENIT
LCALL AND0FPC
MOV A,#'*'
LCALL SENDCHR
MOV A,STLTPOS
ADD A,#$30

```

```
LCALL SENDCHR
MOV A,#'*'
LCALL SENDCHR
MOV A,#$0D
LCALL SENDCHR
RET
```

```
ANDF0PC: SWAP A
AND0FPC: ANL A,#$0F
ADD A,#$30
LCALL SENDCHR
RET
```

```
TURUNLANTAI2KE1:
LCALL LANTAI2KE1
MOV POSISI,#1
LCALL SEGMENT1
MOV A,#'1'
LCALL SENDCHR
MOV A,#$0D
LCALL SENDCHR
RET
```

```
TURUNLANTAI3KE1:
LCALL LANTAI3KE1
MOV POSISI,#1
LCALL SEGMENT1
MOV A,#'1'
LCALL SENDCHR
MOV A,#$0D
LCALL SENDCHR
RET
```

```
TURUNLANTAI3KE2:
LCALL LANTAI3KE2
```

```
MOV  POSISI,#2
LCALL SEGMENT2
MOV  A,#'2'
LCALL SENDCHR
MOV  A,#$0D
LCALL SENDCHR
RET
```

NAIKLANTAI1KE2:

```
LCALL LANTAI1KE2
MOV  POSISI,#2
LCALL SEGMENT2
MOV  A,#'2'
LCALL SENDCHR
MOV  A,#$0D
LCALL SENDCHR
RET
```

NAIKLANTAI1KE3:

```
LCALL LANTAI1KE3
MOV  POSISI,#3
LCALL SEGMENT3
MOV  A,#'3'
LCALL SENDCHR
MOV  A,#$0D
LCALL SENDCHR
RET
```

NAIKLANTAI2KE3:

```
LCALL LANTAI2KE3
MOV  POSISI,#3
LCALL SEGMENT3
MOV  A,#'3'
LCALL SENDCHR
MOV  A,#$0D
LCALL SENDCHR
RET
```

```

ADAKEYPAD1:  LJMPP ADAKEYPAD
SW1AKTIF:
    MOV  A,POSISI
CPOS1_1  CJNE  A,#$1,CPOS1_2
    LCALL PINTUBUKATUTUP
LSW1_1:   LCALL SCANNINGKEYPAD
    CJNE  A,#0,ADAKEYPAD1
    LJMP  LSW1_1
CPOS1_2  CJNE  A,#$2,CPOS1_3
    LCALL LANTAI2KE1
    MOV  POSISI,#1
    LCALL SEGMENT1
    MOV  A,#'1'
    LCALL SENDCHR
    MOV  A,#$0D
    LCALL SENDCHR
    LCALL PINTUBUKATUTUP
LSW1_2:   LCALL SCANNINGKEYPAD
    CJNE  A,#0,ADAKEYPAD1
    LJMP  LSW1_2

CPOS1_3  CJNE  A,#$3,CPOS1_3
    LCALL LANTAI3KE1
    MOV  POSISI,#1
    LCALL SEGMENT1
    MOV  A,#'1'
    LCALL SENDCHR
    MOV  A,#$0D
    LCALL SENDCHR
    LCALL PINTUBUKATUTUP
LSW1_3:   LCALL SCANNINGKEYPAD
    CJNE  A,#0,ADAKEYPAD1
    LJMP  LSW1_3
ADAKEYPAD2:  LJMPP ADAKEYPAD

```

```

SW2AKTIF:
    MOV  A,POSISI
CPOS2_1  CJNE A,#$1,CPOS2_2
    LCALL LANTAI1KE2
    MOV  POSISI,#2
    LCALL SEGMENT2
    MOV  A,#'2'
    LCALL SENDCHR
    MOV  A,#$0D
    LCALL SENDCHR
    LCALL PINTUBUKATUTUP
LSW2_1:  LCALL SCANNINGKEYPAD
    CJNE A,#0,ADAKEYPAD2
    LJMP LSW2_1
CPOS2_2  CJNE A,#$2,CPOS2_3
    LCALL PINTUBUKATUTUP
LSW2_2:  LCALL SCANNINGKEYPAD
    CJNE A,#0,ADAKEYPAD2
    LJMP LSW2_2
CPOS2_3  CJNE A,#$3,CPOS2_3
    LCALL LANTAI3KE2
    MOV  POSISI,#2
    LCALL SEGMENT2
    MOV  A,#'2'
    LCALL SENDCHR
    MOV  A,#$0D
    LCALL SENDCHR
    LCALL PINTUBUKATUTUP
LSW2_3:  LCALL SCANNINGKEYPAD
    CJNE A,#0,ADAKEYPAD2
    LJMP LSW2_3
ADAKEYPAD3:  LJMP ADAKEYPAD

SW3AKTIF:
    MOV  A,POSISI
CPOS3_1  CJNE A,#$1,CPOS3_2
    LCALL LANTAI1KE3

```

```

MOV  POSISI,#3
LCALL SEGMENT3
MOV  A,#'3'
LCALL SENDCHR
MOV  A,#$0D
LCALL SENDCHR
LCALL PINTUBUKATUTUP
LSW3_1:  LCALL SCANNINGKEYPAD
        CJNE A,#0,ADAKEYPAD3
        LJMP LSW3_1
CPOS3_2  CJNE A,#$2,CPOS3_3
        LCALL LANTAI2KE3
        MOV  POSISI,#3
        LCALL SEGMENT3
        MOV  A,#'3'
        LCALL SENDCHR
        MOV  A,#$0D
        LCALL SENDCHR
        LCALL PINTUBUKATUTUP
LSW3_2:  LCALL SCANNINGKEYPAD
        CJNE A,#0,ADAKEYPAD3
        LJMP LSW3_2
CPOS3_3  CJNE A,#$3,CPOS3_3
        LCALL PINTUBUKATUTUP
LSW3_3:  LCALL SCANNINGKEYPAD
        CJNE A,#0,ADAKEYPAD3
        LJMP LSW3_3

ADAKEYPAD:
        MOV  A,DATAKEY
        CJNE A,#$0B,CEKLAIN
        LJMP LOOP0
CEKLAIN:
        CJNE A,#1,CEKANGKA2
        MOV  A,POSISI
        CJNE A,#$2,CEKPOS3KE1

```

```

        LCALL LANTAI2KE1
        MOV  POSISI,#1
        LCALL SEGMENT1
        MOV  A,#'1'
        LCALL SENDCHR
        MOV  A,#$0D
        LCALL SENDCHR
        LCALL PINTUBUKATUTUP
        LJMP LOOP0
CEKPOSPASS1: LJMP CEKPOSPASS
CEKPOS3KE1:  CJNE A,#$3,CEKPOSPASS1
        LCALL LANTAI3KE1
        MOV  POSISI,#1
        LCALL SEGMENT1
        MOV  A,#'1'
        LCALL SENDCHR
        MOV  A,#$0D
        LCALL SENDCHR
        LCALL PINTUBUKATUTUP
        LJMP LOOP0

CEKANGKA2:  CJNE A,#2,CEKANGKA3
        MOV  A,POSISI
        CJNE A,#1,CEKPOS3KE2
        LCALL LANTAI1KE2
        MOV  POSISI,#2
        LCALL SEGMENT2
        MOV  A,#'2'
        LCALL SENDCHR
        MOV  A,#$0D
        LCALL SENDCHR
        LCALL PINTUBUKATUTUP
        LJMP LOOP0
CEKPOS3KE2: CJNE A,#3,CEKPOSPASS
        LCALL LANTAI3KE2
        MOV  POSISI,#2
        LCALL SEGMENT2

```



```

MOV A,#'2'
LCALL SENDCHR
MOV A,#$0D
LCALL SENDCHR
LCALL PINTUBUKATUTUP
LJMP LOOP0
CEKANGKA3: CJNE A,#3,CEKPOSPASS
MOV A,POSISI
CJNE A,#1,CEKPOS2KE3
LCALL LANTAI1KE3
MOV POSISI,#3
LCALL SEGMENT3
MOV A,#'3'
LCALL SENDCHR
MOV A,#$0D
LCALL SENDCHR
LCALL PINTUBUKATUTUP
LJMP LOOP0
CEKPOS2KE3: CJNE A,#2,CEKPOSPASS
LCALL LANTAI2KE3
MOV POSISI,#3
LCALL SEGMENT3
MOV A,#'3'
LCALL SENDCHR
MOV A,#$0D
LCALL SENDCHR
LCALL PINTUBUKATUTUP
CEKPOSPASS: LJMP LOOP0

;-----
LANTAI1KE2:
LCALL MOTORNAIK
TLANTAI2:
JB SLANTAI2,TLANTAI2
LCALL MOTOROFFSEMUA
RET
LANTAI2KE3:

```

```

        LCALL MOTORNAIK
TLANTAI3:
        JB  SLANTAI3,TLANTAI3
        LCALL MOTOROFFSEMUA
        RET

LANTAI1KE3:
        LCALL MOTORNAIK
TLANTAI1KE2:
        JB  SLANTAI2,TLANTAI1KE2
        LCALL SEGMENT2
        MOV  A,#'2'
        LCALL SENDCHR
        MOV  A,#$0D
        LCALL SENDCHR
TLANTAI2KE3:
        JB  SLANTAI3,TLANTAI2KE3
        LCALL MOTOROFFSEMUA
        RET

LANTAI3KE2:
        LCALL MOTORTURUN
TLANTAI3KE2:
        JB  SLANTAI2,TLANTAI3KE2
        LCALL MOTOROFFSEMUA
        RET
LANTAI2KE1:
        LCALL MOTORTURUN
POSISIRESET21:
        JB  SLANTAI1,POSISIRESET21
        LCALL MOTOROFFSEMUA
        RET

LANTAI3KE1:
        LCALL MOTORTURUN
TLANTAI3KE21:
        JB  SLANTAI2,TLANTAI3KE21

```

```

        LCALL SEGMENT2
        MOV  A,#'2'
        LCALL SENDCHR
        MOV  A,#$0D
        LCALL SENDCHR
POSISIRESET31:
        JB  SLANTAI1,POSISIRESET31
        LCALL MOTOROFFSEMUA
        RET
POSISIRESETPINTU:
        RET

DELAYLANTAI: MOV  R2,#8
DELAYLANTAI1:
        LCALL DELAYSW
        LCALL DELAYSW
        LCALL DELAYSW
        LCALL DELAYSW
        LCALL DELAYSW
        LCALL DELAYSW
        LCALL DELAYSW
        LCALL DELAYSW
        LCALL DELAYSW
        DJNZ R2,DELAYLANTAI1
        RET

PINTUBUKA:
        JB  SPINTUBUKA,PINTUBUKA1
        LCALL MOTOROFFSEMUA
        LCALL DELAYSW
        RET

PINTUBUKA1: LCALL MOTORBUKA
            LJMP PINTUBUKA

PINTUTUTUP:
        JB  SPINTUTUTUP,PINTUTUTUP1
        LCALL MOTOROFFSEMUA

```

```
        LCALL DELAYSW
        RET
PINTUTUTUP1: LCALL MOTORTUTUP
            LJMPL PINTUTUTUP
```

```
PINTUBUKATUTUP: LCALL DELAYSW
                LCALL DELAYSW
                LCALL DELAYSW
                LCALL DELAYSW
                LCALL PINTUBUKA
                LCALL DELAY10X
                LCALL PINTUTUTUP
                RET
```

```
POSISIRESET:
        LCALL MOTORTURUN
POSISIRESET1:
        JB  SLANTAI1,POSISIRESET1
        LCALL MOTOROFFSEMUA
        RET
```

```
MOTOROFFSEMUA:
        MOV  DATABUS,#$00
        LCALL KONTROLM
        RET
```

```
MOTORNAIK:
        MOV  DATABUS,#$01
        LCALL KONTROLM
        RET
```

```
MOTORTURUN:
        MOV  DATABUS,#$03
        LCALL KONTROLM
        RET
```

```
MOTORBUKA:
```

```

        MOV  DATABUS,#$04
        LCALL KONTROLM
        RET
MOTORTUTUP:
        MOV  DATABUS,#$0C
        LCALL KONTROLM
        RET

KONTROLM:  SETB  KONTROL1
           LCALL DELAY_K
           CLR  KONTROL1
           LCALL DELAY_K
           RET

SEGMENTBLANK:
           MOV  DATABUS,#$FF
           LCALL KONTROLSEG
           RET
SEGMENTMIN:
           MOV  DATABUS,#$3F
           LCALL KONTROLSEG
           RET
SEGMENT1:
           MOV  DATABUS,#$79
           LCALL KONTROLSEG
           RET
SEGMENT2:
           MOV  DATABUS,#$24
           LCALL KONTROLSEG
           RET
SEGMENT3:
           MOV  DATABUS,#$30
           LCALL KONTROLSEG
           RET

KONTROLSEG:  SETB  KONTROL2

```

```
LCALL DELAY_K
CLR KONTROL2
LCALL DELAY_K
RET
```

SET_JAMTGL:

```
MOV R1,#BUFFER
LCALL CEKKEYPAD
MOV @R1,A ;J
INC R1
LCALL CEKKEYPAD
MOV @R1,A ;J
INC R1
LCALL CEKKEYPAD
MOV @R1,A ;M
INC R1
LCALL CEKKEYPAD
MOV @R1,A ;M
INC R1
LCALL CEKKEYPAD
MOV @R1,A ;D
INC R1
LCALL CEKKEYPAD
MOV @R1,A ;D
INC R1
```

ISIJAM:

```
MOV A,BUFFER
; ANL A,#$0F
SWAP A
MOV JAM,A
```

JAMPASS:

```
MOV A,BUFFER+1
; ANL A,#$0F
ORL A,JAM
MOV JAM,A
```

```

        MOV    A,BUFFER+2
;       ANL    A,#$0F
        SWAP   A
        MOV    MENIT,A

        MOV    A,BUFFER+3
;       ANL    A,#$0F
        ORL    A,MENIT
        MOV    MENIT,A

        MOV    A,BUFFER+4
;       ANL    A,#$0F
        SWAP   A
        MOV    DETIK,A
        MOV    A,BUFFER+5
;       ANL    A,#$0F
        ORL    A,DETIK
        MOV    DETIK,A

        MOV    R4,#00H
        MOV    R6,#3
        MOV    R1,#DETIK
        LCALL  TULISI2C
        LCALL  DLYTULIS

        MOV    A,POSISI
        CJNE   A,#1,CTAMP2
        LCALL  SEGMENT1
        LJMP   LOOP0
CTAMP2:  CJNE   A,#2,CTAMP3
        LCALL  SEGMENT2
        LJMP   LOOP0
CTAMP3:  CJNE   A,#3,CPASS
        LCALL  SEGMENT3
CPASS:   LJMP   LOOP0

```

PASSJAM:

;SET TGL

```
    MOV  R1,#BUFFER
    LCALL CEKKEYPAD
    MOV  @R1,A
    INC  R1
    LCALL CEKKEYPAD
    MOV  @R1,A    ;
    INC  R1
    LCALL CEKKEYPAD
    MOV  @R1,A
    INC  R1
    LCALL CEKKEYPAD
    MOV  @R1,A    ;
    INC  R1
    LCALL CEKKEYPAD
    MOV  @R1,A
    INC  R1
    LCALL CEKKEYPAD
    MOV  @R1,A    ;
    INC  R1
    LCALL CEKKEYPAD
    MOV  @R1,A
    INC  R1
    LCALL CEKKEYPAD
    MOV  @R1,A
    INC  R1

    MOV  A,BUFFER
    ANL  A,#$0F
    SWAP A
    MOV  HARI,A
    MOV  A,BUFFER+1
    ANL  A,#$0F
    ORL  A,HARI
    MOV  HARI,A
```



```

MOV  A,BUFFER+2
ANL  A,#$0F
SWAP A
MOV  TANGGAL,A
MOV  A,BUFFER+3
ANL  A,#$0F
ORL  A,TANGGAL
MOV  TANGGAL,A
MOV  A,BUFFER+4
ANL  A,#$0F
SWAP A
MOV  BULAN,A
MOV  A,BUFFER+5
ANL  A,#$0F
ORL  A,BULAN
MOV  BULAN,A
MOV  A,BUFFER+6
ANL  A,#$0F
SWAP A
MOV  TAHUN,A
MOV  A,BUFFER+7
ANL  A,#$0F
ORL  A,TAHUN
MOV  TAHUN,A

```

ISITGL:

```

MOV  R3,#00H
MOV  R4,#03H
MOV  R6,#4
MOV  R1,#HARI
LCALL TULISI2C
LCALL DLYTULIS
LJMP LOOP

```

BACA_WAKTU:

```

MOV R3,#$00 ;high address
MOV R4,#$00 ;low address
MOV R6,#7 ;jumlah counter
MOV R1,#DETIK
LCALL BACAI2C
LCALL DLYTULIS
;----- Tulis Waktu Jika Jam Mati
MOV A,DETIK
ANL A,#80H
CJNE A,#80H,WAKTU_RET
ANL DETIK,#7FH
LCALL DLYTULIS
MOV R3,#$00 ;high address
MOV R4,#$00 ;low address
MOV R6,#1 ;jumlah counter
MOV R1,#DETIK
LCALL TULISI2C
LCALL DLYTULIS
WAKTU_RET
RET

CEKKEYPAD: LCALL SCANNINGKEYPAD
CJNE A,#$0,CEKKEYPASS
LJMP CEKKEYPAD
CEKKEYPASS: CJNE A,#$0B,CEKKEYPASS1
LCALL DELAYSW
LJMP LOOP0
CEKKEYPASS1: CJNE A,#$0A,BIASA
MOV DATAKEY,#0
BIASA: MOV A,DATAKEY
RET

SCANNINGKEYPAD:
SCANKEYPAD:
KOLOM1: MOV DATAKEY,#0
CLR X1
SETB X2

```

```

    SETB  X3
    MOV   A,SWITCH
    ANL   A,#0FH
CTOMBOL1: CJNE  A,#0EH,CTOMBOL4
    MOV   DATAKEY,#01H
    LJMP  TOLCD
CTOMBOL4: CJNE  A,#0DH,CTOMBOL7
    MOV   DATAKEY,#04H
    LJMP  TOLCD
CTOMBOL7: CJNE  A,#0BH,CTOMBOLB
    MOV   DATAKEY,#07H
    LJMP  TOLCD
CTOMBOLB: CJNE  A,#07H,KOLOM2
    MOV   DATAKEY,#0BH
    LJMP  TOLCD

KOLOM2:  SETB  X1
    CLR   X2
    SETB  X3
    MOV   A,SWITCH
    ANL   A,#0FH
CTOMBOL2: CJNE  A,#0EH,CTOMBOL5
    MOV   DATAKEY,#02H
    LJMP  TOLCD
CTOMBOL5: CJNE  A,#0DH,CTOMBOL8
    MOV   DATAKEY,#05H
    LJMP  TOLCD
CTOMBOL8: CJNE  A,#0BH,CTOMBOL0
    MOV   DATAKEY,#08H
    LJMP  TOLCD
CTOMBOL0: CJNE  A,#07H,KOLOM3
    MOV   DATAKEY,#0AH
    LJMP  TOLCD
KOLOM3:  SETB  X1
    SETB  X2
    CLR   X3
    MOV   A,SWITCH

```

```

        ANL  A,#0FH
CTOMBOL3: CJNE  A,#0EH,CTOMBOL6
        MOV  DATAKEY,#03H
        LJMP TOLCD
CTOMBOL6: CJNE  A,#0DH,CTOMBOL9
        MOV  DATAKEY,#06H
        LJMP TOLCD
CTOMBOL9: CJNE  A,#0BH,CTOMBOLP
        MOV  DATAKEY,#09H
        LJMP TOLCD
CTOMBOLP: CJNE  A,#07H,KEYPADRET
        MOV  DATAKEY,#0CH
TOLCD:
        LCALL DELAYSW
        MOV  A,DATAKEY
KEYPADRET:
        MOV  A,DATAKEY
        RET

;-----
; PROC RUTIN INTERRUPT SERIAL
; TERIMA DATA DARI PRINTER MODUL
;-----
SERINT:
        JBC  RI,GETDATASERIAL
        RETI

GETDATASERIAL: PUSH  ACC
        MOV  A,SBUF
        CLR  RI
CEKSER1:  CJNE  A,#'1',CEKSER2
        MOV  STATUSSER,#1
        POP  ACC
        RETI
CEKSER2:  CJNE  A,#'2',CEKSER3
        MOV  STATUSSER,#2
        POP  ACC

```

```

        RETI
CEKSER3:  CJNE  A,#'3',CEKSER4
          MOV   STATUSSER,#3
          POP   ACC
          RETI
CEKSER4:  CJNE  A,#'4',CEKSER5
          MOV   STATUSSER,#4
          POP   ACC
          RETI
CEKSER5:  CJNE  A,#'5',PASSRET
          MOV   STATUSSER,#5
          POP   ACC
          RETI
PASSRET:  MOV   STATUSSER,#0
          POP   ACC
          RETI

```

```

;-----KIRIM DATA SERIAL-----

```

```

SENDCHR:
        CLR  TI
        MOV  SBUF,A
TXLOOP:
        JNB  TI,TXLOOP
        RET

```

```

;-----AMBIL DATA SERIAL-----

```

```

GETCHR:  JNB  RI,GETCHR
        MOV  A,SBUF
        ANL  A,#$7F
        CLR  RI
        RET

```

```

INIT_SERIAL:
        MOV  TMOD, #20
        MOV  TCON, #41
        MOV  TH1, #fd
        MOV  SCON, #50

```

```
SETB TR1
RET
```

```
TULISI2C1:
```

```
BYTEWL:
```

```
MOV A,ADDR_WI2C ;LOAD WRITE COMMAND
LCALL OUTS ;SEND IT
MOV A,R3 ;GET HI BYTE ADDRESS
LCALL OUT ;SEND IT
MOV A,R4 ;GET LOW BYTE ADDRESS
LCALL OUT ;SEND IT
```

```
BTLP11:
```

```
CLR A
MOV A,@R1
LCALL OUT
INC R1
DJNZ R6,BTLP11
LCALL STOP
```

```
; MOV A,R1 ;GET DATA
; LCALL OUT ;SEND IT
; LCALL STOP ;SEND STOP CONDITION
RET
```

```
;-----
; Procedure Baca Data Dari I2C Bus
;-----
```

```
TULISI2C:
```

```
MOV A,ADDR_WI2C
LCALL OUTS
MOV A,R4
LCALL OUT
```

```
BTLP1: CLR A
MOV A,@R1
LCALL OUT
INC R1
DJNZ R6,BTLP1
LCALL STOP
```

RET

BACAI2C1:

BLKRDL: MOV A,ADDR_WI2C ;LOAD WRITE COMMAND TO SEND
ADDRESS

LCALL OUTS ;SEND IT
MOV A,R3 ;GET HI BYTE ADDRESS
LCALL OUT ;SEND IT
MOV A,R4 ;GET LOW BYTE ADDRESS
LCALL OUT ;SEND IT
MOV A,ADDR_RI2C ;LOAD READ COMMAND
LCALL OUTS ;SEND IT
SJMP BRDLP ;CONTINUE WITH DATA READ

BACAI2C:

MOV A,ADDR_WI2C ;LOAD WRITE COMMAND TO SEND
ADDRESS

LCALL OUTS ;SEND IT
MOV A,R4 ;GET LOW BYTE ADDRESS
LCALL OUT ;SEND IT
MOV A,ADDR_RI2C ;LOAD READ COMMAND
LCALL OUTS ;SEND IT

BRDLP: LCALL IN ;READ DATA
MOV @R1,A ;STORE DATA
INC R1 ;INCREMENT DATA POINTER
DJNZ R6,AKLP ;DECREMENT LOOP COUNTER
LCALL STOP ;IF DONE, ISSUE STOP CONDITION
RET ;DONE, EXIT ROUTINE

AKLP: CLR SDA ;NOT DONE, ISSUE ACK

SETB SCK
MOV R7,#3
LCALL DELAYCK
CLR SCK
SJMP BRDLP ;CONTINUE WITH READS

```

OUTS:  MOV  R2,#8
        SETB SDA
        SETB SCK
        MOV  R7,#2
        LCALL DELAYCK
        CLR  SDA
        MOV  R7,#2
        LCALL DELAYCK
        CLR  SCK
OTSLP:  RLC  A
        JNC  BITLS
        SETB SDA
        SJMP OTSL1
BITLS:  CLR  SDA
OTSL1:

```

```

        SETB SCK
        MOV  R7,#3
        LCALL DELAYCK
        CLR  SCK
        DJNZ R2,OTSLP
        SETB SDA
        MOV  R7,#2
        LCALL DELAYCK
        SETB SCK
        MOV  R7,#3
        LCALL DELAYCK
        CLR  SCK
        RET

```

```

OUT:    MOV  R2,#8
OTLP:   RLC  A
        JNC  BITL
        SETB SDA

```



```

        SJMP  OTL1
BITL:   CLR   SDA
OTL1:
        SETB  SCK
        MOV   R7,#3
        LCALL DELAYCK
        CLR   SCK
        DJNZ  R2,OTLP
        SETB  SDA
        MOV   R7,#2
        LCALL DELAYCK
        SETB  SCK
        MOV   R7,#3
        LCALL DELAYCK
        CLR   SCK
        RET

IN:     MOV   R2,#8
        SETB  SDA
INLP:   CLR   SCK
        MOV   R7,#3
        LCALL DELAYCK
        SETB  SCK
        CLR   C
        JNB  SDA,INL1
        CPL   C
INL1:   RLC   A
        DJNZ  R2,INLP
        CLR   SCK
        RET

STOP:   CLR   SDA      ;STOP CONDITION SET DATA LOW
        MOV   R7,#2
        LCALL DELAYCK
        SETB  SCK      ;SET CLOCK HI
        MOV   R7,#2

```

```
LCALL DELAYCK
SETB SDA ;SET DATA HIGH
```

```
MOV R7,#8
LCALL DELAYCK
RET
```

```
DELAYCK: DJNZ R7,DELAYCK
RET
```

```
DELAY_K:
MOV R7,#10
D_LK1: DJNZ R7,D_LK1
RET
```

```
DLYTULIS:
DELAYTL0: MOV R6,#50
DELAYTL1: MOV R7,#100
DELAYTL2: DJNZ R7,DELAYTL2
DJNZ R6,DELAYTL1
RET
```

```
DELAY10X: LCALL DELAYSW
LCALL DELAYSW
LCALL DELAYSW
LCALL DELAYSW
LCALL DELAYSW
LCALL DELAYSW
LCALL DELAYSW
LCALL DELAYSW
LCALL DELAYSW
LCALL DELAYSW
RET
```

```
DELAYSW:
MOV R5,#02H
```

```
DELAYSW0:  MOV  R6,#0FFH
DELAYSW1:  MOV  R7,#0FFH
DELAYSW2:  DJNZ R7,DELAYSW2
           DJNZ R6,DELAYSW1
           DJNZ R5,DELAYSW0
           RET
```

```
DELAYBT:   MOV  R5,#1
DELAYBT0:  MOV  R6,#$FF
DELAYBT1:  MOV  R7,#$FF
DELAYBT2:  DJNZ R7,DELAYBT2
           DJNZ R6,DELAYBT1
           DJNZ R5,DELAYBT0
           RET
```

```
KEY:      .BYTE      " 1234567890*#"
```

```
.END
```

LAMPIRAN B

```
unit Unit1;

interface
j5
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, ExtCtrls, QCCom32, StdCtrls, Buttons;

type
    TForm1 = class(TForm)
        COM: TQCCom32;
        Timer1: TTimer;
        Panel1: TPanel;
        BitBtn1: TBitBtn;
        Shape1: TShape;
        lift: TShape;
        Label1: TLabel;
        Label2: TLabel;
        Label3: TLabel;
        Label4: TLabel;
        Label5: TLabel;
        p3: TPanel;
        p2: TPanel;
        p1: TPanel;
        Bevel1: TBevel;
        Label6: TLabel;
        Pkoneksi: TPanel;
        Timer2: TTimer;
        procedure Timer1Timer(Sender: TObject);
        procedure FormCreate(Sender: TObject);
        procedure Timer2Timer(Sender: TObject);
    private
        { Private declarations }
    end;
end;
```

```

public
  { Public declarations }
end;

var
  Form1: TForm1;
  datars232:string;
  Hitung:integer;

implementation

{$R *.dfm}

procedure TForm1.Timer1Timer(Sender: TObject);
begin
  datars232:=com.read;
  if length(datars232)>0 then
  begin
    if copy(datars232,1,1)='J' then
    begin

panel1.caption:=copy(datars232,2,2)+' '+copy(datars232,4,2)+' '+copy(datars232,6,2)
;
      p1.caption:=copy(datars232,9,2);
      p2.caption:=copy(datars232,11,2);
      p3.caption:=copy(datars232,13,2);
      hitung:=0;
      pkoneksi.color:=clgreen;
    end else
    begin
      if copy(datars232,1,1)='1' then
      begin
        lift.top:=192;
        lift.left:=40;
        hitung:=0;
        pkoneksi.color:=clgreen;
      end;
    end;
  end;
end;

```

```

if copy(datars232,1,1)='2' then
begin
lift.top:=128;
lift.left:=40;
hitung:=0;
pkoneksi.color:=clgreen;
end;
if copy(datars232,1,1)='3' then
begin
lift.top:=64;
lift.left:=40;
hitung:=0;
pkoneksi.color:=clgreen;
end;
end;
end;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
com.Pick;
hitung:=0;
end;

procedure TForm1.Timer2Timer(Sender: TObject);
begin
hitung:=hitung+1;

if hitung>=60 then
begin
pkoneksi.color:=clred;
hitung:=0;
end;

end;

end.

```