Lampiran

Source Code Program

```
;------keypad's vars-------
kunci equ 20h
keypad equ 40h


;------LCD's vars--------
 DB0      EQU 0B0h ;P3.0
 DB1      EQU 0B1h ;P3.1
 DB2      EQU 0B2h ;P3.2
 DB3      EQU 0B3h ;P3.3
 DB4      EQU 0B4h ;P3.4
 DB5      EQU 0B5h ;P3.5
 DB6      EQU 0B6h ;P3.6
 DB7      EQU 0B7h ;P3.7
 LCD_EN   EQU 0A7h ;P2.7
 LCD_RW   EQU 0A6h ;P2.6
 LCD_RS   EQU 0A5h ;P2.5
 DATA EQU 0B0h

;----------LED-------------
LED_red        EQU p2.0
LED_green      EQU p2.1
LED_yellow     EQU p2.2
BUZZER         EQU p2.4
OUTPUT         EQU P2

;---log invalid code
timesInv  EQU 50h
limitInv  EQU 51h

org 0

;###############Main Program ############################
;#                                                      #
;########################################################

begin:
mov timesInv,#0          ;banyaknya salah kode awal = 0
mov limitInv,#3          ;batas salah kode =  3
mov kunci,#0B7h          ;digit ke-1=> 1
mov kunci+1,#0E7h        ;digit ke-2=> 3
mov kunci+2,#0DEh        ;digit ke-3=> 0
mov kunci+3,#0BDh        ;digit ke-4=> 7
mov kunci+4,#0DDh        ;digit ke-5=> 8
mov kunci+5,#0BBh        ;digit ke-6=> 4
LCALL INITIALIZE         ;inisisalisasi LCD
```

```
mulai:
LCALL CLEAR_SCREEN
mov OUTPUT, #0FFh
mov p0, #0FFh
MOV A,#80H
LCALL ADDRESS
mov dptr,#kal1      ;Kunci Digital v1' @1st row
lcall TRANSFER
mov A,#0C0H
LCALL ADDRESS       ;By : Akeda Bagus' @2nd row
mov dptr,#kal2
lcall TRANSFER
lcall delay_1s      ;tahan 2 detik
lcall delay_1s
LCALL CLEAR_SCREEN  ;layar bersih
MOV A,#80H
LCALL ADDRESS
mov dptr,#kal3      ;--------------- @1st row
lcall TRANSFER
mov A,#0C0H
LCALL ADDRESS       ; Please press # @2nd row
mov dptr,#kal4
lcall TRANSFER

requestIn:          ;minta ditekan '#' dulu..
 lcall ambilData
 cjne A,#0EEh,requestIn
modeRequest:
 lcall inKode       ;input 6 Kode

;############ CHECK EACH DIGIT ########################
;# Setiap digit yang disimpan di RAM (symbol directive
:# keypad),
;# akan dicek. Apabila digit ke-1 sudah salah, maka
;# langsung gagal,
;# bila benar lanjut ke digit ke-2. Bila digit ke-2
;# salah, langsung
;# gagal. Demikian seterusnya sampai 6 digit
;#####################################################

cekPasswd1:
 mov A,keypad
 cjne A,kunci,gagal
 cekPasswd2:
 mov A,keypad+1
 cjne A,kunci+1,gagal
cekPasswd3:
```

```
 mov A,keypad+2
 cjne A,kunci+2,gagal
cekPasswd4:
 mov A,keypad+3
 cjne A,kunci+3,gagal
cekPasswd5:
 mov A,keypad+4
 cjne A,kunci+4,gagal
cekPasswd6:
 mov A,keypad+5
 cjne A,kunci+5,gagal
;#########################################################

 sjmp modeValid     ;ke-6 digit valid, masuk mode valid
gagal:
 ljmp gagal_        ;digit invalid


;############ MODE VALID ################################
;#                                                      #
;########################################################
modeValid:
 clr LED_yellow
 lcall CLEAR_SCREEN
 lcall CURSOR_OFF
 mov A,#80h
 lcall ADDRESS
 mov dptr,#kal6
 lcall TRANSFER
 lcall delay_1s
 mov A,#0C0h
 lcall ADDRESS
 mov dptr,#kal7
 lcall TRANSFER
 lcall delay_1s
 lcall CLEAR_SCREEN
 mov A,#80h
 lcall ADDRESS
 mov dptr,#kal8          ;1. Buka kunci
 lcall TRANSFER
 mov A,#0C0h
 lcall ADDRESS
 mov dptr,#kal9          ;2. Ganti kode
 lcall TRANSFER
pilih1or2:
 lcall ambilData
 cjne A,#0B7h,apa2       ;klo bukan '1', cek tombol '2'
```

```
 sjmp pilih1             ;berarti pilih '1'
apa2:
 cjne A,#0D7h,pilih1or2  ;bukan '1'or '2'
                         ;cek truz
 sjmp pilih2             ;klo '2' loncat ke pilih2

;############# 1. Kunci Terbuka ########################
pilih1:
 lcall CLEAR_SCREEN
 mov A,#80h
 lcall ADDRESS
 mov dptr, #kal10   ;kunci terbuka!!
 lcall TRANSFER
 mov A,#0C0h
 lcall ADDRESS
 mov dptr, #kal11   ;Closed after xs
 lcall TRANSFER
 mov r7, #09h       ;9 detik redirect
 lcall COUNT_DOWN
 ljmp mulai

;############ 2. request new code ####################
pilih2:
 lcall inKode
 mov kunci,keypad
 mov kunci+1,keypad+1
 mov kunci+2,keypad+2
 mov kunci+3,keypad+3
 mov kunci+4,keypad+4
 mov kunci+5,keypad+5
 lcall CLEAR_SCREEN
 mov A,#80h
 lcall ADDRESS
 mov dptr, #kal14
 lcall TRANSFER
 lcall delay_1s
 ljmp mulai
;#########################################################


;############# MODE INVALID ############################
;#                                                      #
;########################################################
gagal_:
 inc timesInv
 mov A, timesInv
 cjne A, limitInv, mshBisa
```

```
 sjmp gagal_total
mshBisa:
 clr BUZZER
 lcall CLEAR_SCREEN
 lcall CURSOR_OFF
 mov A,#80h
 lcall ADDRESS
 mov dptr,#kal12    ;Kode salah..!!
 lcall TRANSFER
 mov A,#0C0h
 lcall ADDRESS
 mov dptr, #kal13   ;beBack after xs
 lcall TRANSFER
 mov r7, #05h       ;9 detik redirect
 lcall COUNT_DOWN2
 ljmp mulai

;###################### INVALID 3x !! ################
;# scrolling effect display hehe...
;# and wait for around 1 minuete
;#####################################################

gagal_total:
 clr timesInv       ;apus log invalid
 clr BUZZER
 lcall CLEAR_SCREEN
 lcall CURSOR_OFF
 mov A,#0C0h
 lcall ADDRESS
 mov dptr, #kal15   ;eLo dah salah xX 2nd row
 lcall TRANSFER
 mov A, #0CEh
 lcall ADDRESS
 mov A, limitInv
 ADD A,#30h
 lcall WRITE_ON
 lcall delay_1s
 lcall delay_1s
 lcall CLEAR_SCREEN
 mov A,#80h
 lcall ADDRESS
 mov dptr, #kal15   ;eLo dah salah xX 1st row
 lcall TRANSFER
 mov A, #8Eh
 lcall ADDRESS
 mov A, limitInv
 ADD A,#30h
```

```
 lcall WRITE_ON
 mov A,#0C0h
 lcall ADDRESS
 mov dptr, #kal16   ;Lupa or Maling? 2nd row
 lcall TRANSFER
 lcall delay_1s
 lcall delay_1s
 lcall CLEAR_SCREEN
 mov A,#80h
 lcall ADDRESS
 mov dptr, #kal16   ;Lupa or Maling? 1st row
 lcall TRANSFER
 mov A,#0C0h
 lcall ADDRESS
 mov dptr, #kal17   ;Tunggu 1 menit  2nd row
 lcall TRANSFER
wait1men:
 mov r7, #60        ;60 x 1s
wait1men_:
 mov r6, #10        ;100ms x 10 = 1s
wait1men2:
 mov A, #100        ;100ms
 lcall delay_Xms
 djnz r6, wait1men2
 djnz r7, wait1men_
 ljmp mulai

;#########################################################

;################ END OF MAIN PROGRAM ################
;#                                                       #
;#########################################################

;%%%%%%%%%%%%%%%%%% KUMPULAN RUTIN %%%%%%%%%%%%%%%%%%
;%                                                     %
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;rutin LCD
INITIALIZE:
SETB LCD_EN
CLR LCD_RW
CLR LCD_RS
MOV DATA,#00110000B
CLR LCD_EN
LCALL DELAY1
SETB LCD_EN
CLR LCD_RW
CLR LCD_RS
```

```
MOV DATA,#00110000B
CLR LCD_EN
LCALL DELAY1
SETB LCD_EN
CLR LCD_RW
CLR LCD_RS
MOV DATA,#00110000B
CLR LCD_EN
LCALL DELAY1
SETB LCD_EN
CLR LCD_RW
CLR LCD_RS
MOV DATA,#00111000B
CLR LCD_EN
LCALL DELAY1
SETB LCD_EN
CLR LCD_RW
CLR LCD_RS
MOV DATA,#00001100B
CLR LCD_EN
LCALL DELAY1
SETB LCD_EN
CLR LCD_RW
CLR LCD_RS
MOV DATA,#00000110B
CLR LCD_EN
LCALL DELAY1
RET

CURSOR_BLINK:
MOV A,#0FH
LCALL COMMAND
RET

CURSOR_OFF:
MOV A,#0CH
LCALL COMMAND
RET

CURSOR_CUSTOM:
MOV A,#0C0H
LCALL COMMAND
RET

SHIFT_LEFT_SCREEN:
MOV A,#18H
LCALL COMMAND
```

```
RET

SHIFT_RIGHT_SCREEN:
MOV A,#1CH
LCALL COMMAND
RET

COMMAND:
MOV DATA,A
SETB LCD_EN
CLR LCD_RW
CLR LCD_RS
LCALL DELAY0
CLR LCD_EN
LCALL DELAY0
RET

CLEAR_SCREEN:
SETB LCD_EN
CLR LCD_RS
CLR LCD_RW
MOV DATA,#00000001B
LCALL DELAY0
CLR LCD_EN
LCALL DELAY0
RET


ADDRESS:
SETB LCD_EN
CLR LCD_RW
CLR LCD_RS
MOV DATA,A
LCALL DELAY0
CLR LCD_EN
LCALL DELAY0
RET

WRITE_ON:
MOV DATA,A
SETB LCD_EN
CLR LCD_RW
SETB LCD_RS
LCALL DELAY0
CLR LCD_EN
LCALL DELAY0
LCALL DELAY0
```

```
RET

TRANSFER:
CLR A
MOVC A,@A+DPTR
INC DPTR
CJNE A,#0FFH,TRANS
LJMP EXIT3
TRANS:
MOV DATA,A
LCALL WRITE_ON
LJMP TRANSFER
EXIT3:
RET

DELAY0:
PUSH 7
MOV R7,#1
SUB_DELAY:
MOV TMOD,#00000001B
MOV TH0,#0FCH
MOV TL0,#00H
SETB TR0
TF0?:
JNB TF0,TF0?
CLR TR0
CLR TF0
DJNZ R7,SUB_DELAY
POP 7
RET


DELAY1:
PUSH 7
MOV R7,#1
SUB_DELAY1Z:
MOV TMOD,#00000001B
MOV TH0,#0A0H
MOV TL0,#00H
SETB TR0
TF0??:
JNB TF0,TF0??
CLR TR0
CLR TF0
DJNZ R7,SUB_DELAY1Z
POP 7
RET
```

```
COUNT_DOWN:
 CLR LED_green
 MOV A,#0CDh
 lcall ADDRESS
 MOV A,R7
 ADD A,#30h
 lcall WRITE_ON
 SETB LED_green
 lcall delay_1s
 djnz r7,COUNT_DOWN
RET

COUNT_DOWN2:
 CLR LED_red
 MOV A,#0CDh
 lcall ADDRESS
 MOV A,R7
 ADD A,#30h
 lcall WRITE_ON
 SETB LED_red
 lcall delay_1s
 djnz r7,COUNT_DOWN2
RET


;===keypad routine=======
inKode:
 lcall CLEAR_SCREEN ;bersihkan layar
 mov A,#80h
 lcall ADDRESS
 mov dptr,#kal5          ;Masukkan Password :
 lcall TRANSFER
 mov A,#0C0H
 lcall ADDRESS
 lcall delay_1s
 lcall CURSOR_BLINK      ;aktifkan kursor blink
 passwd1:                ;digit ke-1
 lcall ambilData
 cjne A, #0BEh, pass1
 clr BUZZER
 lcall DELAY0
 setb BUZZER
 sjmp passwd1
 pass1:                  ;digit ke-1 bukan '*'
 mov keypad, A
 mov A,#0C0h
```

```
lcall ADDRESS
mov A,#'*'
lcall WRITE_ON
mov A, #50
acall delay_Xms ;50ms
passwd2:                ;digit ke-2
lcall ambilData
cjne A, #0BEh, pass2
mov A,#0C0H
lcall ADDRESS
mov A,#' '
lcall WRITE_ON
mov A,#0C0H
lcall ADDRESS
sjmp passwd1
pass2:                  ;digit ke-2 bukan '*'
mov keypad+1,A
mov A,#'*'
lcall WRITE_ON
mov A, #50
acall delay_Xms         ;50ms
passwd3:                ;digit ke-3
lcall ambilData
cjne A, #0BEh, pass3
mov A,#0C1H
lcall ADDRESS
mov A,#' '
lcall WRITE_ON
mov A,#0C1H
lcall ADDRESS
sjmp passwd2
pass3:                  ;digit ke-3 bukan '*'
mov keypad+2,A
mov A,#'*'
lcall WRITE_ON
mov A, #50
acall delay_Xms ;50ms
passwd4:                ;digit ke-4
lcall ambilData
cjne A, #0BEh, pass4
mov A,#0C2H
lcall ADDRESS
mov A,#' '
lcall WRITE_ON
mov A,#0C2H
lcall ADDRESS
sjmp passwd3
```

```
 pass4:                  ;digit ke-4 bukan '*'
 mov keypad+3,A
 mov A,#'*'
 lcall WRITE_ON
 mov A, #50
 acall delay_Xms         ;50ms
 passwd5:                ;digit ke-5
 lcall ambilData
 cjne A, #0BEh, pass5
 mov A,#0C3H
 lcall ADDRESS
 mov A,#' '
 lcall WRITE_ON
 mov A,#0C3H
 lcall ADDRESS
 sjmp passwd4
 pass5:                  ;digit ke-5 bukan '*'
 mov keypad+4,A
 mov A,#'*'
 lcall WRITE_ON
 mov A, #50
 acall delay_Xms         ;50ms
 passwd6:                ;digit ke-6
 lcall ambilData
 cjne A, #0BEh, pass6
 mov A,#0C4H
 lcall ADDRESS
 mov A,#' '
 lcall WRITE_ON
 mov A,#0C4H
 lcall ADDRESS
 sjmp passwd5
 pass6:                  ;digit ke-6 bukan '*'
 mov keypad+5,A
 mov A,#'*'
 lcall WRITE_ON
RET

ambilData:
 mov p0, #0FFh
datapad:
 mov A, #50              ;50ms
 acall delay_Xms
 clr A
 mov A, p0
 push ACC
 mov A, #50
```

```
 acall delay_Xms         ;50ms
 pop ACC
 cjne A,#0FFh,ambil1
 ajmp datapad
ambil1:
 cjne A,#0B7h,ambil2
 sjmp ambil              ;'1' ditekan
ambil2:
 cjne A,#0D7h,ambil3
 sjmp ambil              ;'2' ditekan
ambil3:
 cjne A,#0E7h,ambil4     ;'3' ditekan
 sjmp ambil
ambil4:
 cjne A,#0BBh,ambil5     ;'4' ditekan
 sjmp ambil
 ambil5:
 cjne A,#0DBh,ambil6     ;'5' ditekan
 sjmp ambil
ambil6:
 cjne A,#0EBh,ambil7     ;'6' ditekan
 sjmp ambil
ambil7:
 cjne A,#0BDh,ambil8     ;'7' ditekan
 sjmp ambil
ambil8:
 cjne A,#0DDh,ambil9     ;'8' ditekan
 sjmp ambil
ambil9:
 cjne A,#0EDh,ambilStar  ;'9' ditekan
 sjmp ambil
ambilStar:
 cjne A,#0BEh,ambil0     ;'*' ditekan
 sjmp ambil
ambil0:
 cjne A,#0DEh,ambilSharp ;'0' ditekan
 sjmp ambil
ambilSharp:
 cjne A,#0EEh,ngacow     ;'#' ditekan
 sjmp ambil
ngacow:
 ajmp datapad
ambil:
RET

delay_1s:
          push 7
```

```
          mov r1,#5
loop1:    mov r2,#250
loop2:    mov r3,#250
loop3:    djnz r3, loop3
          djnz r2, loop2
          djnz r1, loop1
          pop 7
RET

delay_Xms:
          mov r1, A      ;A x 1000 = x us
          mov TMOD, #01  ;timer 0 - 16 bit
lagi:     mov TH0, #HIGH(-1000)
          MOV TL0, #LOW(-1000)
          SETB TR0
tunggu:
          JNB TF0, tunggu
          clr TF0
          clr TR0
          djnz r1,lagi
RET


kal1:
     db 'Kunci DigitaL v1',0ffh
kal2:
     db 'By : Akeda Bagus',0ffh
kal3:
     db '----------------',0ffh
kal4:
     db ' Please press # ',0ffh
kal5:
     db 'Masukkan Kode : ',0ffh
kal6:
     db 'OK, you're in...',0ffh
kal7:
     db ' Pilih 1 atau 2 ',0ffh
kal8:
     db '1. Buka kunci   ',0ffh
kal9:
     db '2. Ganti kode   ',0ffh
kal10:
     db 'Kunci terbuka!! ',0ffh
kal11:
     db 'Closed after   s',0ffh
kal12:
     db 'Kode Salah...!!!',0ffh
```

```
kal13:
     db 'beBack after   s',0ffh
kal14:
     db 'Kode telah ganti',0ffh
kal15:
     db 'eLo dah Salah  x',0ffh
kal16:
     db 'Lupa apa Maling?',0ffh
kal17:
     db 'Tunggu 1 Menit..',0ffh
end
```