

**SISTEM OTOMATISASI PENGAMBILAN LEMBAR
HASIL STUDI DAN LEMBAR RENCANA STUDI SETIAP
SEMESTER BERBASIS MIKROKONTROLER AT89C51
DENGAN MENGGUNAKAN PC SEBAGAI BASIS DATA**

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh:

HENDRY KARTIKA
NIM. 0001063280-63

**DEPARTEMEN PENDIDIKAN NASIONAL
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
M A L A N G
2007**

**SISTEM OTOMATISASI PENGAMBILAN LEMBAR
HASIL STUDI DAN LEMBAR RENCANA STUDI SETIAP
SEMESTER BERBASIS MIKROKONTROLER AT89C51
DENGAN MENGGUNAKAN PC SEBAGAI BASIS DATA**

SKRIPSI

Diajukan Untuk Memenuhi Sebagian Persyaratan
Memperoleh Gelar Sarjana Teknik



Disusun oleh:

HENDRY KARTIKA
NIM. 0001063280-63

DOSEN PEMBIMBING :

R. Arief Setyawan ,ST,MT
NIP. 132 231 713

Suprpto ST, MT
NIP. 132 149 320

**SISTEM OTOMATISASI PENGAMBILAN LEMBAR
HASIL STUDI DAN LEMBAR RENCANA STUDI SETIAP
SEMESTER BERBASIS MIKROKONTROLER AT89C51
DENGAN MENGGUNAKAN PC SEBAGAI BASIS DATA**

Disusun oleh:

**HENDRY KARTIKA
NIM. 0001063280-63**

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 10 Agustus 2007

DOSEN PENGUJI

Ir. Ponco Siwindarto, MS.
NIP. 131 837 966

Ir. Bambang Siswojo
NIP. 131 759 588

Tibyani,ST,MT
NIP. 132 135 200

Ir. M. Julius S. ,MS
NIP. 131 124 655

Mengetahui
Ketua Jurusan Teknik Elektro

Ir. Heru Nurwarsito,M.kom
NIP 131 879 033


```

KBDLOCK      BIT      P2.1
KBDDATA      BIT      P2.0
TE           BIT      P3.3
KEY          EQU      P1
BUFFERKEYBOARD EQU    30H
BUFFERPASS   EQU      10H
BUFFERSEM    EQU      50H
;
;*****
;*****
LINE1        EQU      080H
LINE2        EQU      0C0H
ELCD         BIT      P2.6
RSLCD        BIT      P2.7
STROBE       EQU      P2.4
BUSY         EQU      P2.3
;
;*****
;*****
ORG          0000H
LJMP        START
;
ORG          23H
NOP
RETI
;
; $INCLUDE (LCD20.H51) ;
; $INCLUDE (KEYBOARD.H51) ;
; $INCLUDE (RS232.H51) ;
; $INCLUDE (KEYPAD.H51) ;
;
ORG          30H
;*****
;*****
WAIT41MS:    PUSH     02H          ; tunda 41 ms
             MOV      R2,#30H
             DJNZ    R2,$
             POP     02H
             RET
;*****
; memberi intruksi pada lcd
;*****
SETLCD:      CLR      RSLCD
             SETB   ELCD
             MOV    P0,A
             NOP
             CLR   ELCD
             NOP
             SETB  ELCD
             ACALL WAIT41MS
             RET
;*****
;*****
WAIT16MS:    PUSH     02H          ; tunda 160 ms
             PUSH   03H
             MOV    R2,#08H
WAIT16MS0:   MOV      R3,#0FAH
             DJNZ   R3,$
             DJNZ   R2,WAIT16MS0
             POP    03H
             POP    02H
             RET
;*****
;*****
DISPSMSG1:   MOV      A,#LINE1      ; display pada line 1
             LCALL  DISPSMSG
             RET
;*****
;*****

```



```

DISPMSG2:    MOV     A,#LINE2           ; display pada line 2
             LCALL  DISPMSG
             RET

;*****
;
;*****
INITLCD:     MOV     A,#38H           ; inisialisasi lcd
             ACALL  SETLCD
             MOV     A,#06H
             ACALL  SETLCD
             MOV     A,#0DH
             ACALL  SETLCD
CLEARLCD:    MOV     A,#01H
             ACALL  SETLCD
             LCALL  WAIT16MS
             MOV     A,#02H
             ACALL  SETLCD
             LCALL  WAIT16MS
             RET

;*****
;
;*****
BLINKON:     MOV     A,#0DH           ; blink di aktifkan
             ACALL  SETLCD
             RET

;*****
;
;*****
BLINKOFF:    MOV     A,#0CH           ; blink di non aktifkan
             ACALL  SETLCD
             RET

;*****
;
;*****
DISPPT:      SETB    RSLCD           ; kirim data pada lcd
             SETB    ELCD
             MOV     P0,A
             NOP
             CLR     ELCD
             NOP
             SETB    ELCD
             ACALL  WAIT41MS
             RET

;*****
;
;*****
DISPMSG:     ACALL  SETLCD           ;tampilkan teks dari rom
             MOV     B,#10H
LOOPDISP:    MOV     A,#00H
             MOVC   A,@A+DPTR
             ACALL  DISPPT
             INC     DPTR
             DJNZ   B,LOOPDISP
             RET

;*****
;
;*****
DISPMSGX:    PUSH    DPH             ; tampilkan teks dari internal ram
             PUSH    DPL
             ACALL  SETLCD
             MOV     B,#10H
LOOPDISPX:   MOV     A,@R0
             ACALL  DISPPT
             INC     R0
             DJNZ   B,LOOPDISPX
             POP     DPL
             POP     DPH
             RET

;*****
;
;*****
DISPMSGR:    PUSH    DPH             ; tampilkan teks dari eksternal ram
             PUSH    DPL

```

```

ACALL SETLCD
MOV B,#10H
LOOPDISPR: MOVX A,@DPTR
ACALL DISPTP
INC DPTR
DJNZ B,LOOPDISPR
POP DPL
POP DPH
RET

;*****
;
;*****
SHIFTCURSOR: PUSH ACC ; geser kursor
LCALL BLINKOFF
POP ACC
LCALL SETLCD
MOV A,B
CJNE A,#00H,SHIFTA
LCALL BLINKON
RET
SHIFTA: MOV A,#16H
LCALL SETLCD
DJNZ B,SHIFTA
LCALL BLINKON
RET
;
;*****
; PERNYATAAN PEMAKAIAN PORT
;*****

;*****
; MENGAMBIL DATA 1 BIT
;*****
AMBILBIT: JNB KBDCLOCK,$ ; SELAMA KBECLOCK='0' TUNGGU DI BARIS INI
JB KBDCLOCK,$ ; SELAMA KBDCLOCK='1' TUNGGU DI BARIS INI
MOV C,KBDDATA ; AMBIL DATA BIT DI KBDDATA KE CARRY BIT PSW
RET

;*****
; MENGAMBIL DATA 1 BIT
;*****
AMBILBIT1: PUSH B
MOV B,#10H
JNB KBDCLOCK,$ ; SELAMA KBECLOCK='0' TUNGGU DI BARIS INI
AMBILBIT11: JB KBDCLOCK,AB000 ; SELAMA KBDCLOCK='1' TUNGGU DI BARIS INI
MOV C,KBDDATA ; AMBIL DATA BIT DI KBDDATA KE CARRY BIT PSW
POP B
RET
AB000: DJNZ B,AMBILBIT11
POP B
AJMP AMBILBIT1

;*****
; MENGAMBIL DATA 1 BYTE DARI KEYBOARD
;*****
AMBILBYTE1: ;***** BAGIAN 'START BIT'
ACALL AMBILBIT1
JC AMBILBYTE1 ; SALAH, 'START BIT' HARUS ='0'
LJMP ABT000

;*****
; MENGAMBIL DATA 1 BYTE DARI KEYBOARD
;*****
AMBILBYTE: ;***** BAGIAN 'START BIT'
ACALL AMBILBIT
JC AMBILBYTE ; SALAH, 'START BIT' HARUS ='0'

;*****
;***** BAGIAN 8 BIT DATA
ABT000: MOV R7,#8
AMBILLAGI: ACALL AMBILBIT ; AMBIL 1 BIT
RRC A ; GESER MASUK KE A
DJNZ R7,AMBILLAGI ; SUDAH 8 KALI? ULANGI KALAU BELUM
MOV R7,A ; SIMPAN DULU HASILNYA DI R7

;*****
;***** BAGIAN BIT PARITI
MOV C,P ; 'PARITI GENAP' DARI ACC. A SIMPAN DULU KE C
CLR A ; A := 00000000
MOV ACC.0,C ; A.0 := PARITI YANG DISIMPAN DI C
ACALL AMBILBIT ; AMBIL 'PARITI GASAL' YANG DIKIRIM KEYBOARD

```




```

MOV ACC.1,C ; SIMPAN KE A.1
MOV C,P ; YANG BENAR ADALAH A.1<>A.0, ATAU P=1
JNC SALAH ; KALAU C=0, SALAH!

;***** BAGIAN 'STOP BIT'
MOV A,R7 ; KEMBALIKAN R7 KE A
ACALL AMBILBIT ; AMBIL 'STOP BIT', KALAU C=0, SALAH!
SALAH: RET

;*****
; MERUBAH SCANCODE MENJADI KODE ASCII
;*****
TABELSCANCODE: ; 00H,01H,02H,03H,04H,05H,06H,07H,08H,09H,0AH,0BH,0CH,0DH,0EH,0FH
DB 00H,00H,00H,07H,05H,03H,00H,00H,00H,0CH,00H,00H,00H,00H,'~',00H ;0
DB 00H,00H,00H,00H,00H,'Q','1',00H,00H,00H,'Z','S','A','W','2',00H ;1
DB 00H,'C','X','D','E','4','3',00H,00H,' ','V','F','T','R','5',00H ;2
DB 00H,'N','B','H','G','Y','6',00H,00H,'M','J','U','7','8',00H ;3
DB 00H,'<','K','I','O','0','9',00H,00H,'>','?','L',':','P','_',00H ;4
DB 00H,00H,'"',00H,'{','+',00H,00H,14H,00H,0AH,'}',00H,'|',00H,00H ;5
DB 00H,00H,00H,00H,00H,00H,01H,00H,00H,00H,02H,00H,00H,00H,00H,00H ;6
DB 00H,00H,13H,00H,10H,12H,11H,00H,00H,00H,00H,00H,00H,00H,00H,00H ;7
DB 00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H ;8
DB 00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H ;9
DB 00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H ;A
DB 00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H ;B
DB 00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H ;C
DB 00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H ;D
DB 00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H ;E
DB 00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H ;F

;*****
;
;*****
JADIKANASCII: PUSH DPH
PUSH DPL
MOV DPTR,#TABELSCANCODE
MOVC A,@A+DPTR
POP DPL
POP DPH
RET

;*****
; MENGIRIM DATA 1 BIT
;*****
KIRIMBIT: JNB KBDCLOCK,$ ; SELAMA KBDCLOCK='0' TUNGGU DI BARIS INI
MOV KBDDATA,C ; KIRIM DATA BIT DI CARRY BIT PSW KE KBDDATA
JB KBDCLOCK,$ ; SELAMA KBDCLOCK='1' TUNGGU DI BARIS INI
RET

;*****
; MENGIRIM DATA 1 BYTE KE KEYBOARD
;*****
KIRIMBYTE: MOV C,P ; SIMPAN DULU BIT PARITI
MOV F0,C ; KE BIT F0 DI PSW

;***** BAGIAN 'MINTA IJIN'
CLR KBDCLOCK ; KBDCLOCK DI-'NOL'-KAN
MOV R7,#60
DJNZ R7,$ ; MENUNGGU DISINI SELAMA 2X60 MIKRO-DETIK
SETB KBDCLOCK ; KBDCLOCK DI-'SATU'-KAN

;***** BAGIAN START BIT
CLR C
ACALL KIRIMBIT

;***** BAGIAN 8 BIT DATA
MOV R7,#8
KIRIMLAGI: RRC A ; PINDAHKAN A.0 KE C
ACALL KIRIMBIT ; KIRIMKAN
DJNZ R7,KIRIMLAGI ; SUDAH 8 KALI? ULANGI KALAU BELUM

;***** BAGIAN BIT PARITI
MOV C,F0
CPL C
ACALL KIRIMBIT ; KIRIM 'PARITI GASAL' KE KEYBOARD KEYBOARD

```



```

;***** BAGIAN STOP BIT
SETB    C
ACALL   KIRIMBIT

;***** BAGIAN MENGAMBIL SINYAL ACK
ACALL   AMBILBIT
RET

;*****
;
;*****
KEYBOARD:    PUSH    07H
KEYBOARD1:   LCALL   AMBILBYTE    ; AMBIL 1 BYTE DARI KEYBOARD
              JNC     KEYBOARD1    ; KALAU ADA KESALAHAN ULANGI LAGI KE MENU
              LCALL   JADIKANASCII ; KALAU TIDAK SALAH JADIKAN KODE ASCII
              AJMP    KEYB003
KEYB001:     AJMP    KEYB003
KEYB000:     CJNE    A,#00H,KEYB003
              ACALL   WAIT16MS
              AJMP    KEYBOARD1
KEYB003:     ;MOV    B,#01H
KEYB002:     PUSH    02H    ; tunda 160 ms
              PUSH    03H
              MOV     R2,#07H
WAIT16MS01:  MOV     R3,#0FAH
              DJNZ   R3,$
              DJNZ   R2,WAIT16MS01
              POP     03H
              POP     02H
              ;DJNZ  B,KEYB002
              POP     07H
              RET

;
;*****
; INISIALISASI SERIAL
;*****
INITSERIAL:  MOV     PCON,#80H    ; SET UNTUK DOUBLE BAUDRATE
              MOV     TMOD,#21H ; SET TIMER MODE AUTO RELOAD
              MOV     SCON,#50H ; SET UNTUK SERIAL MODE 1
              MOV     TL1,#0FDH
              MOV     TH1,#0FDH ; SET UNTUK BAUDRATE 57600
              SETB   TR1    ; JALANKAN TIMER 1
              SETB   EA
              RET           ; KEMBALI

;*****
; TRANSMIT DATA SERIAL
;*****
TRANSMIT:    SETB    ES

              SETB    TE
              CLR     TI    ; CLEAR INTERRUPT TRANSMIT
              MOV     SBUF,A ; KIRIM DATA KE SBUF
              JNB    TI,$   ; MENUNGGU SAMPAI DATA SELESAI DIKIRIM
              PUSH   00H
              MOV     R0,#00
              DJNZ   R0,$
              POP    00H
              CLR    TE
              CLR    ES
              RET           ; KEMBALI

;*****
; AMBIL DATA SERIAL
;*****
RECIEV:     SETB    ES
              CLR    RI
              CLR    TE
              JNB    RI,$   ; MENUNGGU ADANYA DATA
              MOV    A,SBUF ; AMBIL DATA DARI SERIAL BUFFER
              CLR    RI
              CLR    ES
              RET           ; KEMBALI

;
;*****
; KEYPAD

```



```

;*****
; tabel data keypad
; 00H,01H,02H,03H,04H,05H,06H,07H,08H,09H,0AH,0BH,0CH,0DH,0EH,0FH

DATAKEY: DB 31H,32H,33H,34H,35H,36H,37H,38H,39H,0AH,30H,0CH,0DH,00H,0EH,0FH

KEYPAD: PUSH 00H
        PUSH 02H
KP0011: MOV R0,#02H ; NILAI
        MOV R2,#10111111B ; SCAN
KP0021: MOV A,R2
        MOV KEY,A
        NOP
        MOV A,KEY
        ANL A,#0FH
        CJNE A,#0FH,KP0031
KP0001: DEC R0
        MOV A,R2
        RR A
        MOV R2,A
        CJNE A,#11110111B,KP0021
        AJMP KP0011
KP0031: RRC A
        JNC KP0041
        INC R0
        INC R0
        INC R0
        AJMP KP0031
KP0041: MOV A,R0
        MOV DPTR,#DATAKEY
        MOVC A,@A+DPTR
        PUSH ACC
        ACALL WAIT16MS
KP0061: MOV A,KEY
        ANL A,#0FH
        CJNE A,#0FH,KP0061
        POP ACC
        POP 02H
        POP 00H
        RET
;
;*****
;
;*****
;
;*****
INITHARD: CLR TE
          RET
;*****
;
;*****
TLOG01: DB ' MASUKKAN KARTU '

LOGO: LCALL CLEARLCD
      MOV DPTR,#TLOG01
      MOV A,#LINE1
      LCALL DISPMMSG
      RET
;*****
;
;*****
DELAY: PUSH 00H
      PUSH 01H
      MOV R0,#00H
      MOV R1,#00H
DL000: DJNZ R0,$
      DJNZ R1,DL000
      POP 01H
      POP 00H
      RET
;*****
;
;*****
BARCODE: MOV R0,#BUFFERKEYBOARD
        MOV B,#20H
BRD001: MOV @R0,#' '
        INC R0

```



```

DJNZ     B, BRD001
MOV      R0, #BUFFERKEYBOARD
BRD000:  LCALL  KEYBOARD
         MOV    @R0, A
         INC   R0
         CJNE  A, #0AH, BRD000
         DEC   R0
         MOV   @R0, #' '
         RET

;*****
;
;*****
DISPLBAR: MOV    R0, #BUFFERKEYBOARD
         MOV    A, #LINE2
         LCALL  DISPMMSGX
         RET

;*****
;
;*****
TPASS:   DB    ' MASUKKAN PASS '

PASSWORD: LCALL  CLEARLCD
         MOV   DPTR, #TPASS
         MOV   A, #LINE1
         LCALL  DISPMMSG
         MOV   R0, #BUFFERPASS
         MOV   B, #10H
PSW000:  MOV   @R0, #' '
         INC   R0
         DJNZ  B, PSW000
         MOV   R0, #BUFFERPASS
PSW003:  LCALL  KEYPAD
         CJNE  A, #0AH, PSW001
         RET
PSW001:  CJNE  A, #0CH, PSW002
         RET
PSW002:  MOV   @R0, A
         INC   R0
         PUSH  00H
         MOV   R0, #BUFFERPASS
         MOV   A, #LINE2
         LCALL  DISPMMSGX
         POP   00H
         LJMP  PSW003

;*****
;
;*****
TMES:    DB    ' SEMESTER '

SEMESTER: LCALL  CLEARLCD
         MOV   DPTR, #TMES
         MOV   A, #LINE1
         LCALL  DISPMMSG
         MOV   R0, #BUFFERSEM
         MOV   B, #10H
TSW000:  MOV   @R0, #' '
         INC   R0
         DJNZ  B, TSW000
         MOV   R0, #BUFFERSEM
TSW003:  LCALL  KEYPAD
         CJNE  A, #0AH, TSW001
         RET
TSW001:  CJNE  A, #0CH, TSW002
         RET
TSW002:  MOV   @R0, A
         INC   R0
         PUSH  00H
         MOV   R0, #BUFFERSEM
         MOV   A, #LINE2
         LCALL  DISPMMSGX
         POP   00H
         LJMP  TSW003

;*****
;
;*****
TNOT1:   DB    ' PASSWORD '
    
```




```

ST000:   LCALL  BARCODE
        LCALL  DISPLBAR
ST003:   LCALL  KEYPAD
        CJNE  A,#0CH,ST002
        LJMP  ST001
ST002:   CJNE  A,#0AH,ST003
;
        LCALL  DELAY
        MOV   A,#'N'
        LCALL  TRANSMIT
        MOV   A,#'I'
        LCALL  TRANSMIT
        MOV   A,#'M'
        LCALL  TRANSMIT
        MOV   R0,#BUFFERKEYBOARD
        MOV   B,#20H
KRM000:  MOV   A,@R0
        LCALL  TRANSMIT
        INC   R0
        DJNZ  B,KRM000
        MOV   A,#13
        LCALL  TRANSMIT
        LCALL  RECIEV          ;TUNGGU JAWABAN 'OK' DR KOMPUTER
        CJNE  A,#'O',NIMSALAH
        LCALL  RECIEV
        CJNE  A,#'K',NIMSALAH
        MOV   DPTR,#NIM2
        MOV   A,#LINE2
        LCALL  DISPMMSG
        MOV   R7,#05
STBAL2:  LCALL  DELAY
        DJNZ  R7,STBAL2
        LJMP  ST006
NIMSALAH:
        MOV   DPTR,#NIM1
        MOV   A,#LINE2
        LCALL  DISPMMSG
        MOV   R7,#05
STBAL3:  LCALL  DELAY
        DJNZ  R7,STBAL3
        LJMP  ST001
;
ST006:   LCALL  PASSWORD
        CJNE  A,#0AH,ST004
        LCALL  KIRIMDATA
;
        LCALL  RECIEV
        CJNE  A,#'O',PSWSALAH
        LCALL  RECIEV
        CJNE  A,#'K',PSWSALAH
        MOV   DPTR,#TNOT1
        MOV   A,#LINE1
        LCALL  DISPMMSG
        MOV   DPTR,#TNOT3
        MOV   A,#LINE2
        LCALL  DISPMMSG
        MOV   R7,#05
STBAL1:  LCALL  DELAY
        DJNZ  R7,STBAL1
        LCALL  NGEPRINT
        LJMP  ST001
PSWSALAH:
        MOV   DPTR,#TNOT1
        MOV   A,#LINE1
        LCALL  DISPMMSG
        MOV   DPTR,#TNOT2
        MOV   A,#LINE2
        LCALL  DISPMMSG
        MOV   R7,#05
STBAL4:  LCALL  DELAY
        DJNZ  R7,STBAL4
        LCALL  DELAY
        LJMP  ST001
;
;
        LCALL  SEMESTER
    
```



```

;
; CJNE A,#0AH,ST004
; LCALL KIRIMDATA
ST004: LJMP ST001
;
;
;*****
; NGEPRINT DATA KARAKTER PADA LX800
;*****
;
PRNOTE1: DB ' HARAP TUNGGU '
PrNOTE2: DB ' SEDANG CETAK '
;KODE 'F' AWAL,KARAKTER,'D'AKHIR KARAKTER,'G' AKHIR PROSES PRINT
NGEPRINT:
MOV DPTR,#PRNOTE1
MOV A,#LINE1
LCALL DISPMMSG
MOV DPTR,#PRNOTE2
MOV A,#LINE2
LCALL DISPMMSG
mov r0,#BUFFERPASS
;
NGEPBAL1:
LCALL RECIEV
CJNE A,#'F',NGEPBAL1
;
; MOV A,#'T'
; LCALL TRANSMIT
; LCALL RECIEV
; MOV @r0,A
; inc r0
;
; MOV A,#'T'
; LCALL TRANSMIT
; LCALL RECIEV
; CJNE A,#'D',NGEPTER1
; MOV A,@R0
; LCALL PRINTTULIS
; MOV A,#'T' ;JAWAB 'T' DATA TELAH DITERIMA dan di print
; LCALL TRANSMIT
; LJMP NGEPBAL1
NGEPTER1:
CJNE A,#'G',NGEPTER3
MOV A,#'T' ;JAWAB 'T' DATA TELAH DITERIMA
; LCALL TRANSMIT
; MOV A,#13
; LCALL PRINTTULIS
; MOV R7,#10
;NGEPBAL2:
;
; MOV A,#10
; LCALL PRINTTULIS
; DJNZ R7,NGEPBAL2
NGEPTER2:
RET
NGEPTER3:
MOV A,#'R' ;JAWAB 'R' ULANGI PENGIRIMAN DATA PRINT
LCALL TRANSMIT
LJMP NGEPBAL1
;
;*****
;
;*****
;
PRINTTULIS:
JB BUSY,$
MOV P0,A
CLR STROBE
JNB BUSY,$
SETB STROBE
JB BUSY,$
NOP
NOP
NOP
NOP
NOP
NOP
RET
;
;*****
;
;*****
;
;
END

```





PENGANTAR

Assalamualaikum Wr. Wb.

Alhamdulillah Robbil ‘Alammin, segala puji syukur kehadiran Allah SWT atas rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi ini pada waktunya. Skripsi berjudul “Sistem Otomatisasi Pengambilan Lembar Hasil Studi dan Lembar Rencana Studi Setiap Semester Berbasis Mikrokontroler AT89C51 dengan Menggunakan PC sebagai Basis Data” ini disusun sebagai salah satu syarat memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.

Pada kesempatan ini penulis ingin menyampaikan terima kasih yang tak terhingga kepada pihak-pihak yang baik secara langsung maupun secara tidak langsung membantu dan memberikan dukungan dalam menyelesaikan skripsi ini, khususnya kepada:

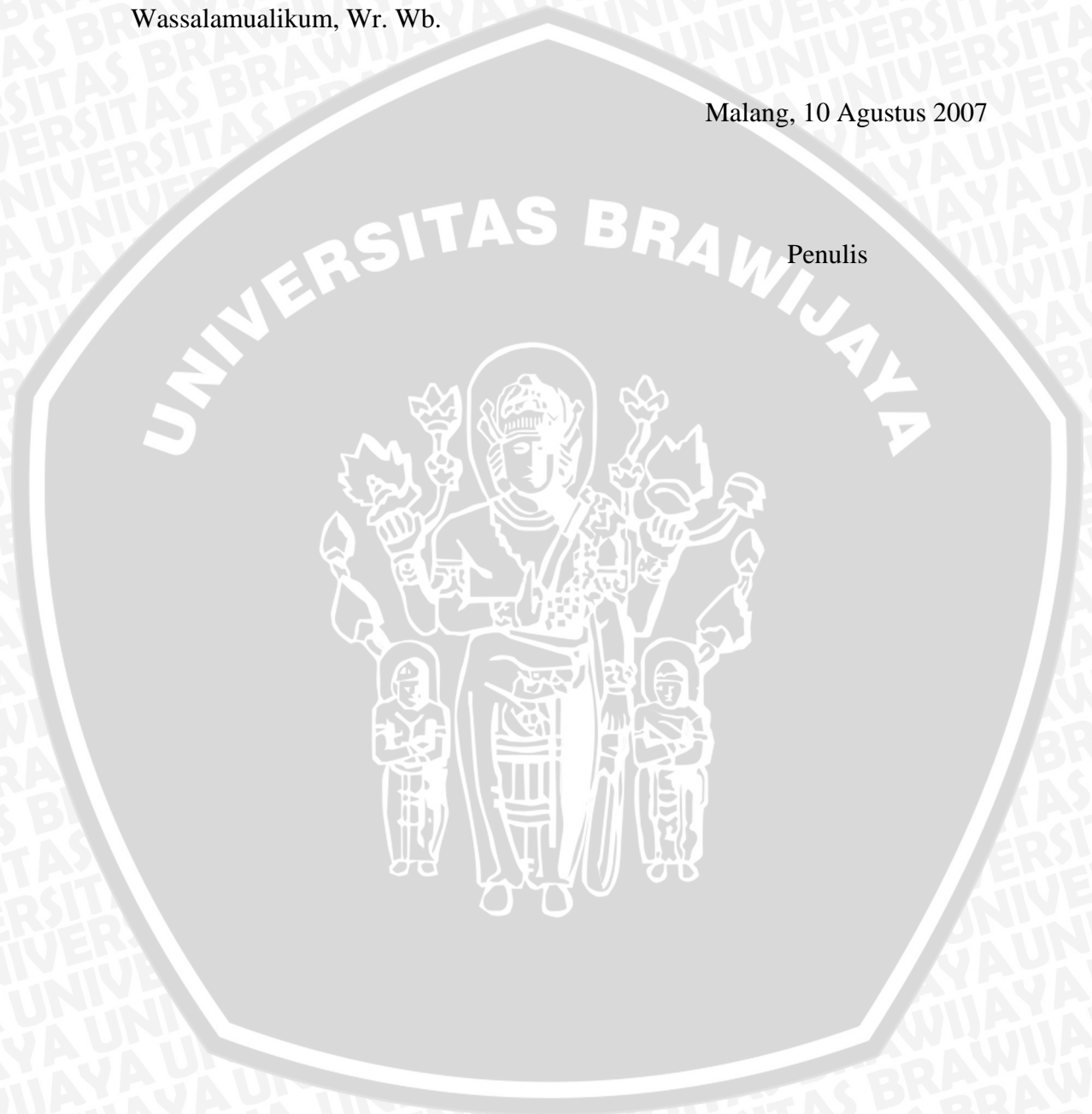
1. Bapak Ir. Heru Nurwarsito, M.Kom selaku Ketua Jurusan Teknik Elektro.
2. Bapak Rudy Yuwono, ST, MSc. selaku Sekretaris Jurusan Teknik Elektro.
3. Bapak Ir. Ponco Siwindarto, MS selaku Ketua Kelompok Pengajar Keahlian Elektronika.
4. Bapak R. Arief Setyawan, ST, MT selaku dosen pembimbing yang senantiasa memberikan koreksi, masukan serta bimbingan hingga Tugas Akhir ini terselesaikan.
5. Bapak Suprpto ST, MT selaku dosen pembimbing yang senantiasa memberikan koreksi, masukan serta bimbingan hingga Tugas Akhir ini terselesaikan.
6. Seluruh dosen dan karyawan Jurusan Teknik Elektro.
7. Kedua orang tuaku Drs. Sukarno dan Ibu Rahayu Rubayati, adikku Ayu Candra dan Septian Bagus serta seluruh keluarga besar yang dengan do’a, kasih sayang dan dukungannya penulis bisa menyelesaikan skripsi ini.
8. Ika Candra A, Dwi Rahayu, Hembing, Andy, Erna, Hahan, Tatang, Arif, Lia, Asih, Dian, Pintarka, Semanggi Crew, Hidayatullah Crew, CMS Crew serta semua teman-teman yang tidak dapat saya sebutkan satu persatu, atas kebersamaannya dan masukannya untuk melengkapi referensi dalam penulisan skripsi ini.
9. Rekan-rekan Elektro UB angkatan 1999, 2000, 2001 dan 2002 dan teman-teman dari ahli jenjang atas segala dukungan dan doanya.

Penulis menyadari ketidaksempurnaan dalam penyusunan skripsi ini, sehingga penulis sangat mengharapkan kritik, saran dan masukan dari pembaca untuk dapat memperbaiki segala kekurangan di dalam skripsi ini. Akhirnya, semoga skripsi ini dapat memberikan manfaat dan berguna terutama bagi mahasiswa Jurusan Teknik Elektro Universitas Brawijaya Malang.

Wassalamualikum, Wr. Wb.

Malang, 10 Agustus 2007

Penulis



DAFTAR ISI

	Halaman
PENGANTAR	i
DAFTAR ISI	iii
DAFTAR TABEL	vi
DAFTAR GAMBAR	vii
DAFTAR LAMPIRAN	ix
ABSTRAK	x
BAB I. PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah	2
1.4. Tujuan	2
1.5. Sistematika Penulisan	3
BAB II. TINJAUAN PUSTAKA	
2.1. Sistem Mikrokontroler AT89C51.....	4
2.1.1 Organisasi Memori.....	7
2.2.1.1 Memori Program.....	8
2.2.1.2 Memori Data.....	8
2.1.2 Pewaktuan <i>CPU</i>	10
2.1.3 Interupsi	11
2.1.3.1 <i>Interrupt Enable</i>	13
2.1.3.2 <i>Interrupt Priority</i>	13
2.1.4 <i>Special Function Register</i>	14
2.1.4 <i>Reset</i>	16
2.2 <i>PC (Personal Computer)</i>	17
2.3. Komunikasi serial antara <i>PC</i> dan <i>MCU</i>	18
2.4. <i>Keypad</i>	19
2.5. <i>LCD (Liquid Crystal Display) M1632</i>	21
2.6. <i>Barcode Reader</i>	23
2.7. Port Printer	26

BAB III. METODOLOGI PENELITIAN

3.1	Kajian Pustaka	30
3.2.	Perencanaan dan Pembuatan Alat.....	30
3.2.1	Perencanaan Alat	30
3.2.1.1	Perancangan piranti lunak.....	31
3.2.1.2	Perancangan piranti keras	31
3.2.1.3	Spesifikasi alat	31
3.3.	Pengujian Alat.....	32
3.5.	Pengambilan Kesimpulan dan Saran	32

BAB IV. PERANCANGAN ALAT

4.1.	Spesifikasi Alat	33
4.2.	Prinsip Kerja Alat	34
4.3.	Perancangan Perangkat Keras.....	35
4.3.1	Antarmuka Mikrokontroler AT89C51	35
4.3.2	Antarmuka LCD (<i>Liquid Crystal Display</i>).....	38
4.3.3	Perencanaan Rangkaian <i>Keypad</i>	40
4.3.4	Antarmuka <i>Printer</i>	41
4.3.5	Antaramuka <i>Barcode Reader</i>	42
4.3.6	Rangkaian Komunikasi Serial	42
4.4.	Perancangan Perangkat Lunak.....	43

BABV. PENGUJIAN ALAT

5.1.	Pengujian Mikrokontroler AT89C51 – LCD.....	47
5.2.	Pengujian <i>Keypad</i>	48
5.3.	Pengujian Barcode Reader.....	49
5.4.	Pengujian Komunikasi Serial <i>PC</i> dengan <i>MCU</i>	53
5.5.	Pengujian <i>MCU</i> - <i>Printer</i>	53
5.6.	Pengujian keseluruhan sistem.....	54

BAB VI.PENUTUP

6.1.	Kesimpulan	57
6.2.	Saran	57

DAFTAR PUSTAKA	58
-----------------------------	----

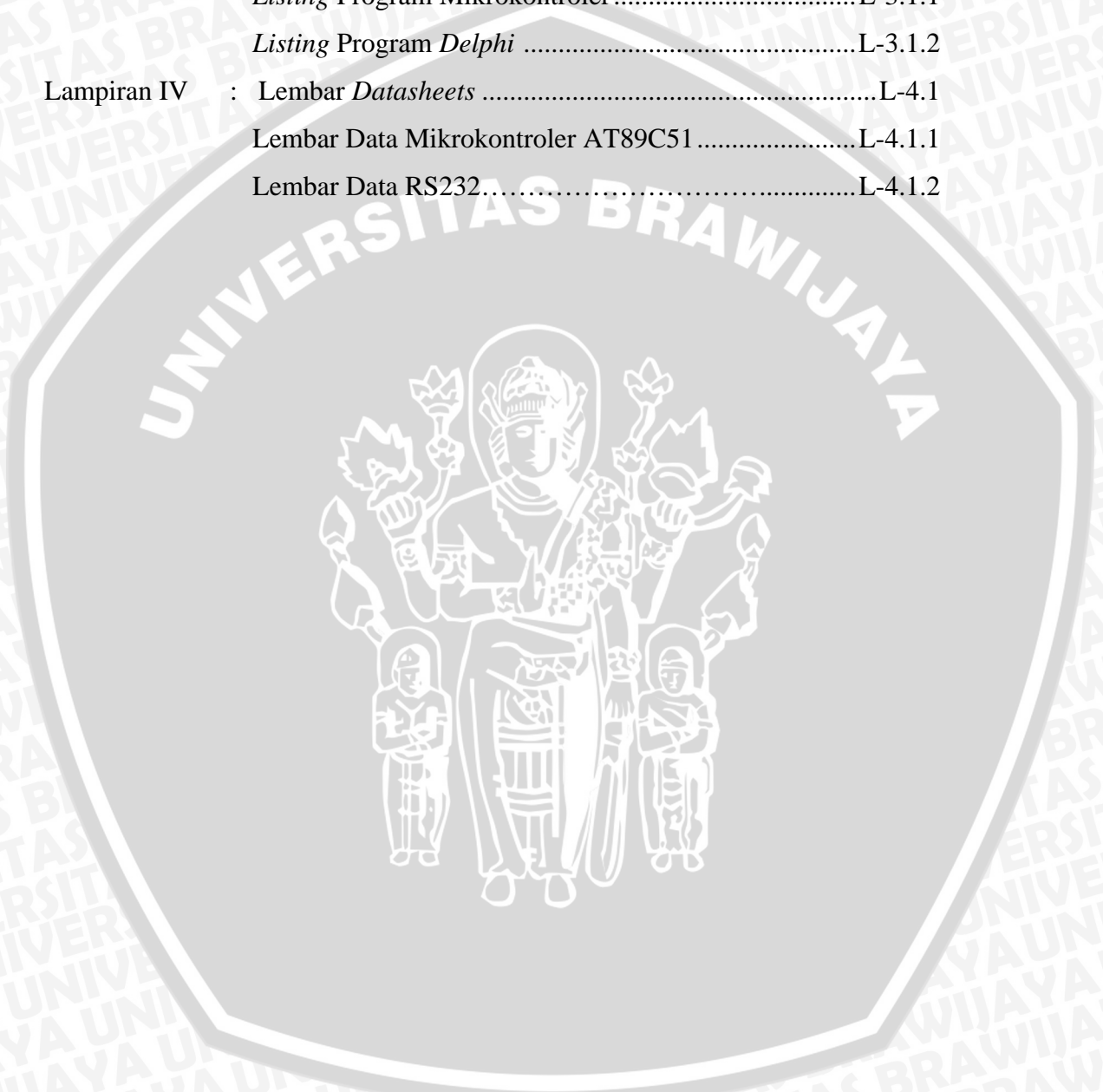
DAFTAR LAMPIRAN

Lampiran I : Rangkaian LengkapL-1.1

Lampiran II : Foto AlatL-2.1

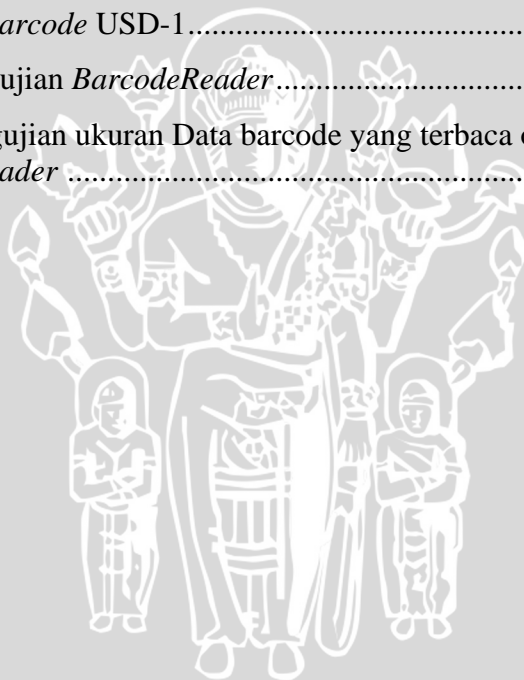
Lampiran III : *Listing* ProgramL-3.1
Listing Program MikrokontrolerL-3.1.1
Listing Program *Delphi*L-3.1.2

Lampiran IV : Lembar *Datasheets*L-4.1
Lembar Data Mikrokontroler AT89C51L-4.1.1
Lembar Data RS232.....L-4.1.2



DAFTAR TABEL

		Halaman
Tabel 2.1	Alamat layanan rutin interupsi.....	12
Tabel 2.2	<i>Special Function Register</i>	15
Tabel 2.3	Fungsi Pin RS-232 dalam DB9.....	19
Tabel 2.4	Spesifikasi RS-232.....	19
Tabel 2.5	Spesifikasi <i>Barcode</i> tipe SD 1000 Series	24
Tabel 2.6	Karakter <i>Barcode</i> USD-1.....	25
Tabel 2.7	Alamat Register port pada LPT1	27
Tabel 2.8	Tabel <i>Data Port, Status Port dan Control Port</i>	28
Tabel 5.1	Kombinasi Bit <i>Keypad</i>	48
Tabel 5.2	Karakter <i>Barcode</i> USD-1.....	49
Tabel 5.3	Hasil Pengujian <i>BarcodeReader</i>	51
Tabel 5.4	Tabel pengujian ukuran Data barcode yang terbaca oleh <i>barcode reader</i>	52



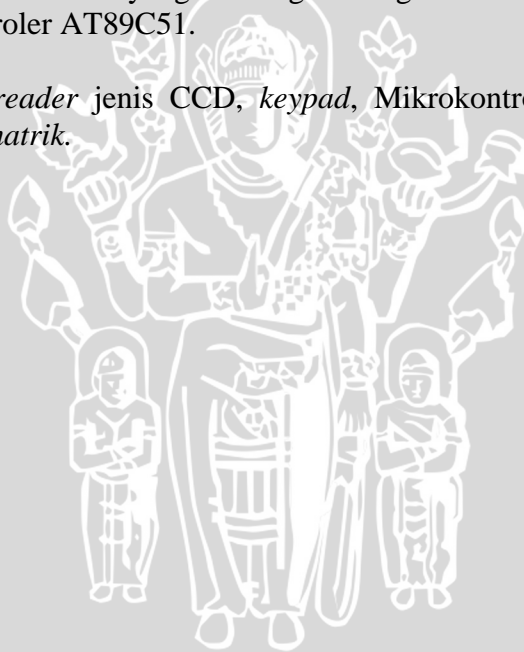
ABSTRAK

Hendry Kartika, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya, Agustus 2007, **Perancangan Sistem Otomatisasi Pengambilan Lembar Hasil Studi dan Lembar Rencana Studi Setiap Semester Berbasis Mikrokontroler AT89C51 dengan Menggunakan PC sebagai Basis Data** .
Dosen Pembimbing : R. Arief Setyawan ST, MT dan Suprpto ST, MT..

Sistem ini merupakan satu paket alat yang berfungsi untuk mengambil KHS atau KRS tiap semester yang bertujuan untuk meringankan tugas pelayanan akademik dan mendayagunakan Kartu Mahasiswa, bukan hanya untuk identitas saja.

Sistem ini memanfaatkan Kartu Mahasiswa sebagai masukan data melalui *barcode reader* jenis CCD 1000 series untuk mengakses NIM dan *keypad* untuk mengakses *password*. Dalam hal ini data yang diakses berupa *database* dengan program *borland Delphi 7* pada PC yang berupa data KHS dan KRS. Untuk keluaran sistem ini berupa LCD M1632 yang berfungsi untuk menampilkan NIM dan *password* serta *printer dot matrik* Epson LX-800 sebagai alat cetak KHS dan KRS yang diakses oleh mahasiswa. Sedangkan subsistem yang berfungsi mengendalikan sistem keseluruhan adalah berupa Mikrokontroler AT89C51.

Kata kunci : *barcode reader* jenis CCD, *keypad*, Mikrokontroler AT89C51, LCD M1632, PC, *Printer dot matrik*.



BAB I PENDAHULUAN

1.1 Latar Belakang

Ilmu pengetahuan dan teknologi terus berkembang pesat, khususnya di bidang elektronika. Demikian pula permasalahan yang dihadapi manusia semakin kompleks sejalan dengan berjalannya waktu. Kondisi demikian ini menuntut manusia untuk terus berpacu mencari alternatif penyelesaian dengan cepat, tepat serta seefektif dan seefisien mungkin. Sehingga otomatisasi di segala bidang terus dikembangkan, sedangkan cara-cara manual dan konvensional perlahan ditinggalkan karena dianggap sudah tidak sesuai dengan kebutuhan

Sebagai contoh, pada umumnya pengambilan Lembar Hasil Studi dan Lembar Rencana Studi Setiap Semester dilakukan oleh mahasiswa langsung melalui loket Pelayanan Urusan Kemahasiswaan. Pengambilan Lembar Hasil Studi dan Lembar Rencana Studi tersebut hanya dilakukan pada batas waktu tertentu saja. Pada saat pengambilan ini sering menimbulkan antrian sehingga terkadang menimbulkan kekecewaan bagi mahasiswa yang telah menunggu lama dalam antrian. Mereka menyangka bahwa pelayanan petugasnya kurang cekatan sehingga terkesan kurang memuaskan.

Ketika seorang mahasiswa ingin mengambil Lembar Hasil Studi dan Lembar Rencana Studi, ia harus menunjukkan kartu mahasiswa kepada petugas loket. Petugas loket akan mencari Lembar Hasil Studi dan Lembar Rencana Studi mahasiswa tersebut sesuai dengan nama dan NIM yang terdapat pada kartu mahasiswa. Proses demikian akan memakan waktu yang relatif lama dan melelahkan bagi petugas loket, sementara terjadi antrian yang kurang teratur di depan loket tersebut.

Dengan mengetahui kasus di atas maka sistem otomatisasi ini akan banyak membantu mempermudah Petugas loket pelayanan urusan kemahasiswaan (meningkatkan efisiensi tenaga), antrian akan lebih teratur dan adanya pendayagunaan kartu mahasiswa

1.2 Rumusan Masalah

Berdasarkan latar belakang ada beberapa rumusan yang akan menjadi bahasan dalam proposal perancangan ini, yaitu:

- (1) Bagaimana sistem kerja Mikrokontroler AT89C51 sehingga dapat digunakan untuk mengendalikan sistem otomatisasi pengambilan Lembar Hasil Studi dan Lembar Rencana Studi .
- (2) Pemanfaatan *Barcode Reader* yang berfungsi sebagai pembaca data barcode yang dimiliki oleh setiap mahasiswa pada kartu mahasiswanya.
- (3) Pemanfaatan *LCD* sebagai penampil informasi.
- (4) Pemanfaatan sistem *software* pada Mikrokontroler AT89C51 dan bahasa pemrograman *Borland Delphi* pada *PC*

1.3 Batasan Masalah

1. Tidak membahas rangkaian catu daya
2. Tidak membahas sistem software secara detail
3. Tidak membahas *PC* secara detail, hanya untuk basis penyimpanan data
4. Tidak membahas *Barcode Reader* secara detail, hanya untuk pembacaan *barcode*
5. Tidak membahas *LCD* secara detail, hanya untuk penampil informasi
6. Tidak membahas *Printer* secara detail, hanya untuk sarana pencetakan

1.4 Tujuan

Tujuan penulisan tugas akhir ini adalah merancang dan membuat sistem otomatisasi dalam pengambilan Lembar Hasil Studi dan Lembar Rencana Studi Setiap Semester, sehingga secara otomatis akan meningkatkan efisiensi tenaga dan waktu petugas loket pelayanan

1.5 Sistematika Penulisan

Sistematika penulisan Tugas Akhir ini adalah sebagai berikut:

BAB I : Memuat latar belakang, rumusan masalah, ruang lingkup masalah, tujuan penulisan, dan sistematika penulisan

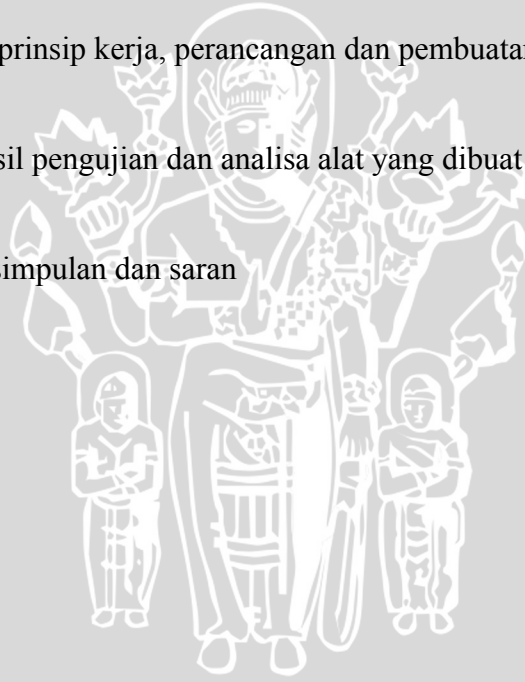
BAB II : Membahas teori-teori dasar penunjang perancangan dan pembuatan Alat

BAB III : Membahas metodologi yang digunakan

BAB IV : Membahas prinsip kerja, perancangan dan pembuatan alat

BAB V : Memuat hasil pengujian dan analisa alat yang dibuat

BAB VI : Memuat kesimpulan dan saran



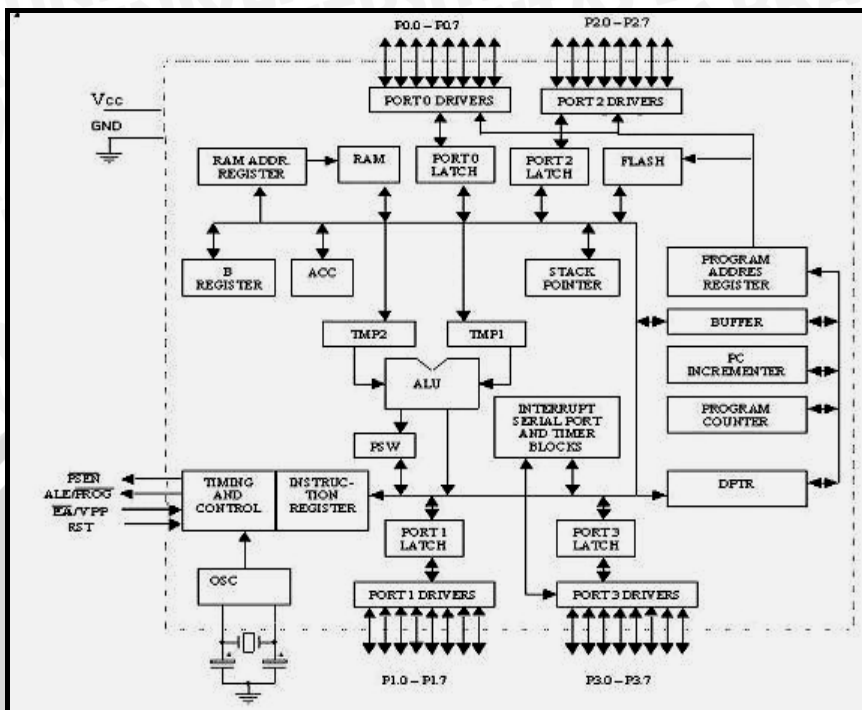
BAB II

TINJAUAN PUSTAKA

2.1 Sistem Mikrokontroler AT89C51

Arsitektur mikrokontroler AT89C51 ditunjukkan dalam Gambar 2.1 Mikrokontroler AT89C51 terdiri dari sebuah *Control Processing Unit* (CPU), dua jenis memori yaitu data (RAM) dan memori program (ROM), port *input* atau *output* dengan *programmable* pin secara *independent*, dan register-register mode, status, *timer* internal dan *counter*, *serial communication* dan logika random yang diperlukan oleh berbagai fungsi periperhal. Masing-masing bagian saling berhubungan satu dengan yang lain lewat kabel data bus 8 bit. Bus ini *dibuffer* melalui port I/O bila diperlukan perluasan memori atau sebagian perangkat I/O. MCU AT89C51 mempunyai arsitektur sebagai berikut :

1. 8 bit CPU dengan register A (*accumulator*) dan *match register* (register B).
2. 16 bit *Program Counter* (PC) dan *Data Pointer* (DPTR) register.
3. 8 bit *Program Status Word* (PSW) register.
4. 8 bit *Stack Pointer*.
5. Internal ROM dan EPROM dengan kapasitas 4kbyte.
6. Internal RAM dengan kapasitas 128 byte yang digunakan untuk 4 buah register bank, yang masing-masing terdiri dari 8 register, 16 *byte*, yang mana dapat dieksekusi pada masing-masing bit secara *independent* (*Bit Addesable*) dan sebagai memori variable 8 bit.
7. 32 input/output yang disusun pada 4 port (port 0 – port 3).
8. 2 buah 16 bit *timer/counter* :T0 dan T1.
9. *Full Duplex Serial Data Communication* :SBUF.
10. *Control Register* :TCON, TMOD, PCON, IP dan IE.
11. 2 eksternal *interrupt* dan 3 internal *interrupt*.
12. Oscilator dan *Clock Circuit*.



Gambar 2.1. Arsitektur AT89C51

Sumber: Atmel, 1997:2

Masing-masing kaki dalam mikrokontroler AT89C51 mempunyai fungsi tersendiri. Dengan mengetahui fungsi masing-masing kaki dari mikrokontroler AT89C51, perancangan aplikasi mikrokontroler AT89C51 akan lebih mudah untuk direalisasikan. AT89C51 mempunyai 40 pin, susunan masing-masing pin dapat dilihat dalam Gambar 2.2.

P1.0	1	40	VCC
P1.1	2	39	P0.0 (AD0)
P1.2	3	38	P0.1 (AD1)
P1.3	4	37	P0.2 (AD2)
P1.4	5	36	P0.3 (AD3)
P1.5	6	35	P0.4 (AD4)
P1.6	7	34	P0.5 (AD5)
P1.7	8	33	P0.6 (AD6)
RST	9	32	P0.7 (AD7)
(RXD) P3.0	10	31	EA/VPP
(TXD) P3.1	11	30	ALE/PROG
(INT0) P3.2	12	29	PSEN
(INT1) P3.3	13	28	P2.7 (A15)
(T0) P3.4	14	27	P2.6 (A14)
(T1) P3.5	15	26	P2.5 (A13)
(WR) P3.6	16	25	P2.4 (A12)
(RD) P3.7	17	24	P2.3 (A11)
XTAL2	18	23	P2.2 (A10)
XTAL1	19	22	P2.1 (A9)
GND	20	21	P2.0 (A8)

Gambar 2.2. Susunan Kaki MCS-51

Sumber: Atmel AT89C51 Data Sheets, 1997: 1

Fungsi kaki-kaki AT89C51 adalah:

- Port 1 (Pin 1...8), berfungsi sebagai port I/O biasa.
- Pin 9 (RST), pulsa transisi dari rendah ke tinggi yang diumpungkan ke pin RST akan mereset AT89C51. Pin ini dihubungkan dengan rangkaian *power on reset*.
- Port 3 (Pin 10...17), port paralel 8 bit dua arah yang memiliki fungsi pengganti. Fungsi pengganti meliputi TXD (*Transmit Data*) yang digunakan sebagai port serial *output*, RXD (*Receive Data*) yang digunakan sebagai port serial *input*, INT0 (*Interrupt 0*) yang digunakan sebagai port eksternal *interrupt 0*, INT1 (*Interrupt 1*) yang digunakan sebagai port port eksternal *interrupt 1*, T0 (*Timer 0*) yang digunakan sebagai port eksternal *timer 0 input*, T1 (*Timer 1*) yang digunakan sebagai port eksternal *timer 1 input*, WR (*Write*) yang digunakan sebagai eksternal data *memory write strobe*, dan RD (*Read*) yang digunakan sebagai eksternal data *memory read strobe*. Apabila fungsi pengganti tidak digunakan, pin-pin ini dapat digunakan sebagai port I/O biasa.
- Pin 18 (XTAL1), merupakan pin masukan ke rangkaian osilator internal. Osilator kristal dan sumber osilator luar dapat digunakan.
- Pin 19 (XTAL2), merupakan pin masukan ke rangkaian osilator internal. Pin ini dipakai apabila menggunakan osilator kristal.
- Pin 20 (*Ground*), dihubungkan ke VSS atau *ground*.
- Port 2 (Pin 21...28), port paralel 8 bit dua arah, dapat digunakan sebagai port I/O 8 bit biasa, dan digunakan untuk mengirim *byte* alamat bila digunakan untuk mengakses memori eksternal.
- Pin 29 (PSEN/*Program Store Enable*), merupakan pengontrol yang digunakan untuk mengakses program memori eksternal masuk ke dalam bus selama proses pemberian/pengambilan instruksi.
- Pin 30 (ALE/*Address Latch Enable*), digunakan untuk menahan alamat memori eksternal selama pelaksanaan instruksi.
- Port 31 (EA/*External Access*), bila pin yang diberikan logika tinggi, maka mikrokontroler akan melaksanakan instruksi dari memori program luar.

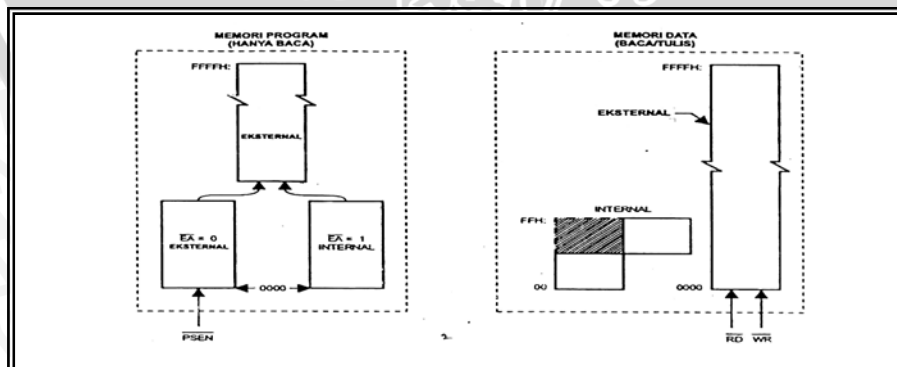
- Port 0 (Pin 32...39), merupakan port paralel 8 bit *open drain* dua arah. Port 0 dapat digunakan sebagai port I/O biasa dan dapat juga digunakan untuk memultipleks alamat dengan data pada waktu mengakses memori eksternal.
- Pin 40 (VCC), dihubungkan dengan VCC (+5 volt).

2.1.1 Organisasi Memori

Semua mikrokontroler dalam keluarga MCS-51 memiliki pembagian ruang alamat (*address space*) untuk program dan data seperti yang ditunjukkan dalam Gambar 2.3. Pemisahan memori program dan memori data memungkinkan memori data untuk dapat di akses oleh alamat 8 bit. Meskipun demikian, alamat data memori 16 bit dapat dihasilkan melalui register DPTR (*Data Pointer Register*)

Memori program hanya dapat dibaca, tidak dapat ditulisi (karena disimpan dalam EPROM (*Erasable and Programmable Read Only Memory*)). Memori program sebesar 64k dapat dimasukkan dalam EPROM (*Erasable and Programmable Read Only Memory*) eksternal.

Sinyal yang memungkinkan pembacaan dari memori program eksternal adalah dari pin \overline{PSEN} (*Program Store Enable*). Memori data terletak pada ruang alamat terpisah dari memori program. RAM (*Random Acces Memory*) eksternal 64k dapat dialamati dalam ruang memori data eksternal. CPU (*Central Processing Unit*) menghasilkan sinyal *read* atau *write* selama menghubungi memori data eksternal.



Gambar 2.3. Struktur Memori Mikrokontroler MCS-51

Sumber: Agfianto Eko Putra, 2002 :3

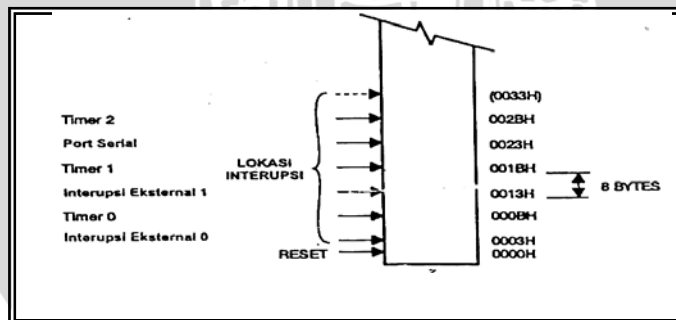
Mikrokontroler AT89C51 memiliki 5 buah ruang alamat, yaitu:

1. Ruang alamat kode (*code address space*) sebanyak 64k, yang seluruhnya merupakan ruang alamat kode eksternal (*off-chip*).
2. Ruang alamat data internal yang dapat dialamati secara langsung, yang terdiri atas:
 - RAM (*Random Access Memory*) sebanyak 128 byte.
 - *Hardware register* sebanyak 128 byte.
 - EEPROM sebanyak 4 kilobyte.
3. Ruang alamat data internal yang dialamati secara tidak langsung sebanyak 128 byte, seluruhnya diakses dengan pengalamatan tidak langsung.
4. Ruang alamat data eksternal sebanyak 64 kilobyte (*off-chip*) yang dapat ditambahkan oleh pemakai.
5. Ruang alamat bit dapat diakses dengan pengalamatan langsung.

2.1.1.1 Memori Program

Gambar 2.4 memperlihatkan bagian bawah dari memori program. Setelah *reset* CPU memulai eksekusi dari lokasi 0000H.

Setiap interupsi mempunyai lokasi tetap dalam memori program. Interupsi menyebabkan CPU melompat ke lokasi tersebut dan pada lokasi tersebut terdapat sub rutin yang harus dilaksanakan.



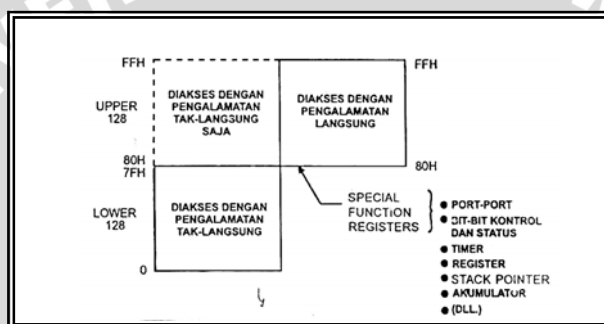
Gambar 2.4. Memori Program
 Sumber: Agfianto Eko Putra, 2002 : 4

2.1.1.2 Memori Data

Memori data internal dipetakan seperti dalam Gambar 2.5. Ruang memorinya dibagi menjadi tiga blok, yaitu sebagai *lower 128*, *upper 128*, dan ruang SFR (*Special Function Register*).

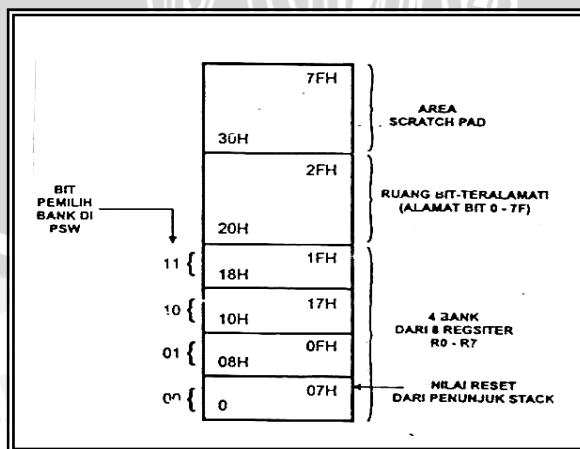
Bagian bawah dari 128 byte RAM dipetakan seperti dalam Gambar 2.6. Berdasarkan Gambar 2.6 ditunjukkan bahwa 32 byte paling bawah dikelompokkan dalam 4 bank (masing-masing terdiri atas 8 register, yaitu R0 sampai R7), 2 bit dalam PSW (*Program Status Word*) digunakan memilih register bank yang digunakan. Hal ini memungkinkan penggunaan yang lebih efisien dari ruang alamat kode, karena instruksi yang menggunakan register lebih singkat dari pada instruksi yang menggunakan pengalamatan secara langsung.

Sedangkan 16 byte di atas bank register (alamat 20h sampai 2Fh) membentuk blok ruang memori untuk pengalamatan bit. Alamat bit pada daerah ini adalah mulai 00h sampai 7Fh.

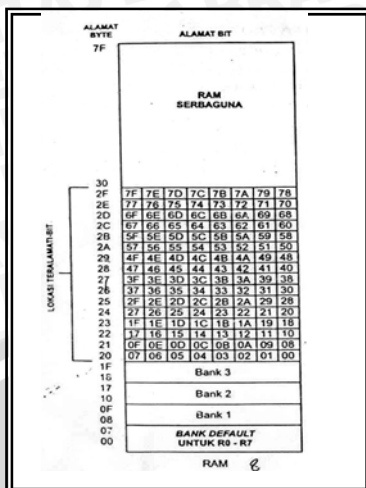


Gambar 2.5 Memori Data Internal
 Sumber: Agfianto Eko Putra, 2002 : 6

Semua byte pada bagian bawah 128 byte RAM internal dapat diakses oleh pengalamatan langsung. Sedangkan bagian atas 128 byte RAM internal hanya dapat diakses dengan pengalamatan tak langsung. Gambar 2.7 menunjukkan bagian atas 128 byte RAM internal.

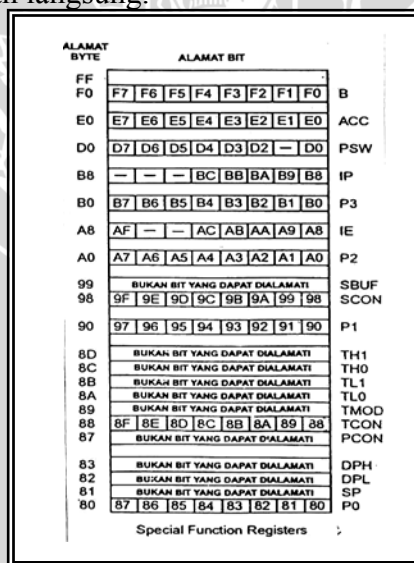


Gambar 2.6 Bagian Bawah 128 Byte RAM Internal
 Sumber: Agfianto Eko Putra, 2002 : 7



Gambar 2.7 Bagian Atas 128 Byte RAM Internal
 Sumber: Agfianto Eko Putra, 2002 : 8

Gambar 2.8 menunjukkan ruang SFR. SFR berisi penahan port (*port latch*), pewaktu (*timer*), pengontrol peripheral, dan lain-lain. Register ini hanya dapat diakses oleh pengalamatan langsung.

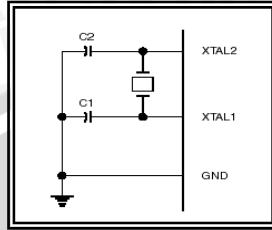


Gambar 2.8. Ruang *Special Function Register*
 Sumber: Agfianto Eko Putra, 2002 : 8

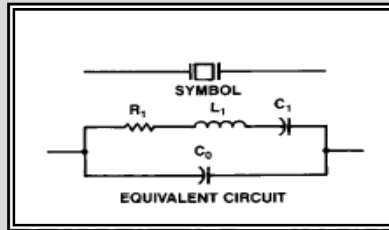
2.1.2 Pewaktuan CPU

Mikrokontroler AT89C51 memiliki osilator internal (*on chip oscillator*) yang dapat digunakan sebagai sumber *clock* bagi CPU. Untuk menggunakan osilator internal diperlukan sebuah kristal. Kristal pada dasarnya merupakan penyetelan rangkaian osilator yang menggunakan *piezo-electric* sebagai rangkaian resonansi. Cara menggunakan osilator internal dipasangkan kristal antara pin XTAL1 dan

XTAL2 serta dua buah kapasitor ke *ground* seperti terlihat dalam Gambar 2.9. Untuk kristalnya dapat digunakan frekuensi dari 6 sampai 12MHz, sedangkan untuk kapasitor dapat bernilai antara 30pF sampai 33pF. Gambar simbol dan rangkaian ekivalen dari kristal dapat kita lihat dalam Gambar 2.10



Gambar 2.9. Rangkaian Osilator Internal
Sumber: Agfianto Eko Putra, 2002 : 22



Gambar 2.10. Simbol dan rangkaian ekivalen Kristal
Sumber: *Oscillators for Microcontrollers Intel*, 2003 : 3

2.1.3 Interupsi

Apabila CPU (*Central Processing Unit*) dalam mikrokontroler AT89C51 sedang melaksanakan suatu program, pelaksanaan program tersebut dapat dihentikan sementara dengan adanya permintaan layanan interupsi. Apabila CPU (*Central Processing Unit*) mendapat permintaan layanan interupsi, *program counter* (PC) akan diisi alamat vektor interupsi. CPU (*Central Processing Unit*) kemudian melaksanakan rutin pelayanan interupsi mulai alamat tersebut. Bila rutin pelayanan interupsi selesai dilaksanakan, CPU(*Central Processing Unit*) AT89C51 kembali ke pelaksanaan program utama yang ditinggalkan.

Dalam mikrokontroler AT89C51 terdapat beberapa saluran interupsi. Interupsi AT89C51 dibedakan dalam 2 jenis, yaitu:

1. Interupsi yang tak dapat dihalangi oleh perangkat lunak (*non maskable interrupt*), misalnya *reset*.

2. Interupsi yang dapat dihalangi oleh perangkat lunak (*maskable interrupt*). Contoh interupsi jenis ini adalah \overline{INT}_0 dan \overline{INT}_1 (eksternal) serta *Timer/counter 0*, *Timer/counter 1*, dan interupsi port serial (internal).

Instruksi RETI (*return from interrupt routine*) harus digunakan untuk kembali dari layanan rutin interupsi. Instruksi ini dipakai agar saluran interupsi kembali dapat dipakai. Alamat awal layanan rutin interupsi setiap sumber interupsi diperlihatkan dalam Tabel 2.1.

Tabel 2.1 Alamat layanan rutin interupsi

Nama	Lokasi	Alat interupsi
Reset	00H	<i>Power on reset</i>
\overline{INT}_0	03H	\overline{INT}_0
<i>Timer 0</i>	0BH	<i>Timer 0</i>
\overline{INT}_1	13H	\overline{INT}_1
<i>Timer 1</i>	1BH	<i>Timer 1</i>
Sint	23H	Port I/O serial

Sumber: Agfianto Eko Putra, 2002 : 157

Mikrokontroler AT89C51 menyediakan 5 sumber interupsi: 2 interupsi eksternal, 2 interupsi *timer*, dan satu interupsi port serial. Interupsi eksternal \overline{INT}_0 dan \overline{INT}_1 masing-masing dapat diaktifkan berdasarkan level atau transisi, tergantung bit IT0 dan IT1 dalam TCON. Flag yang menghasilkan interupsi ini adalah bit dalam IE0 dan IE1 dalam TCON.

Interupsi *Timer 0* dan *Timer 1* dihasilkan oleh TF0 dan TF1. Interupsi port serial dihasilkan oleh logika OR dari R1 dan T1.

Ada dua buah register yang mengontrol interupsi, yaitu IE (*interrupt enable*) dan IP (*interrupt priority*). Mikrokontroler AT89C51 tidak akan menanggapi permintaan interupsi jika suatu instruksi belum dilaksanakan secara lengkap.

2.1.3.1 Interrupt Enable

Setiap sumber interupsi dapat diaktifkan maupun dilumpuhkan secara individual dengan mengatur satu bit di SFR yang bernama IE (*interrupt enable*). Bit-bit IE didefinisikan sebagai berikut:

MSB				LSB			
EA	—	—	ES	ET1	EX1	ET0	EX0

Simbol	Posisi	Fungsi
EA	IE.7	Melumpuhkan semua interupsi. Jika EA=0 tidak ada interupsi yang akan dilayani. Jika EA=1 setiap sumber interupsi dapat dijalankan atau dilumpuhkan secara individual.
—	IE.6	Kosong
—	IE.5	Kosong
ES	IE.4	Bit pembuat <i>enable</i> port serial
ET1	IE.3	Bit pembuat <i>enable timer</i> 1
EX1	IE.2	Bit pembuat <i>enable</i> /INT 1
ET0	IE.1	Bit pembuat <i>enable timer</i> 0
EX0	IE.0	Bit pembuat <i>enable</i> /INT 0

Misalnya jika akan mengaktifkan interupsi 0 ($\overline{\text{INT}}_0$), logika 1 diberikan pada EA dan EX0, sehingga keduanya menjadi aktif dan mengaktifkan $\overline{\text{INT}}_0$.

2.1.3.2 Interrupt Priority

Setiap sumber interupsi dapat diprogram secara individual (sendiri-sendiri) menjadi satu atau dua tingkat prioritas dengan mengatur bit pada SFR (*Special Function Register*) yang bernama IP (*interrupt priority*). Interupsi dengan prioritas rendah (*low priority*) dapat diinterupsi oleh interupsi yang memiliki prioritas lebih tinggi (*high priority*), tetapi tidak dapat diinterupsi oleh interupsi yang memiliki prioritas lebih rendah. Interupsi yang memiliki prioritas tertinggi tidak dapat diinterupsi oleh sumber interupsi yang lainnya.

Jika dua permintaan interupsi dengan tingkat prioritas yang berbeda diterima secara bersamaan, permintaan interupsi dengan prioritas tertinggi yang akan

dilayani. Jika permintaan interupsi dengan prioritas yang sama diterima secara bersamaan, akan dilakukan *polling* untuk menentukan mana yang akan dilayani.

Bit-bit pada IP adalah sebagai berikut:



Priority bit = 1 menandakan prioritas tinggi

Priority bit = 0 menandakan prioritas rendah

Simbol	Posisi	Fungsi
—	IP.7	Kosong
—	IP.6	Kosong
—	IP.5	Kosong
PS	IP.4	Bit prioritas interupsi port serial
PT1	IP.3	Bit prioritas interupsi <i>timer</i> 1
PX1	IP.2	Bit prioritas interupsi /INT 1
PT0	IP.1	Bit prioritas interupsi <i>timer</i> 0
PX0	IP.0	Bit prioritas interupsi /INT 0

2.1.4 Special Function Register

SFR (*Special Function Register*) berisi register-register dengan fungsi tertentu. Masing-masing register ditunjukkan dalam Tabel 2.2 yang meliputi simbol, nama, dan alamat, serta keadaannya dalam nilai biner pada saat terjadi *power-on reset*.

Beberapa fungsi SFR yang penting yaitu:

- *Accumulator* (ACC) merupakan register yang penting dalam operasi penambahan dan pengurangan,
- Register B merupakan register khusus yang berfungsi melayani operasi perkalian dan operasi pembagian.
- *Program Status Word* (PSW) berisi beberapa bit status yang menggambarkan keadaan CPU pada saat itu. PSW terdiri dari bit *carry*, *auxiliary carry*, dua bit pemilih bank, bendera *overflow*, bit paritas dan dua bendera yang dapat didefinisikan sendiri oleh pemakai.

- *Stack Pointer* (SP) merupakan register 8 bit. Register SP digunakan dalam proses simpan dan ambil dari atau ke *stack*, dan dapat diletakkan pada alamat manapun pada RAM internal. Isi register ini ditambah sebelum data disimpan, menggunakan instruksi *PUSH* dan *CALL*. Walaupun *stack* bisa menempati lokasi dimana saja dalam RAM (*Random Acces Memory*), namun setelah adanya *reset*, register SP akan selalu diinisialisasikan pada alamat 07h, sehingga *stack* akan dimulai pada lokasi 08h. Sedangkan untuk pengurangan isi register SP menggunakan instruksi *POP*, yaitu untuk mentransfer data dari areal *stack* yang ditunjukkan oleh *Stack Pointer* ke operand tujuan sehingga menyebabkan *Stack Pointer* berkurang. Selanjutnya instruksi *RET* akan mengembalikan proses pelaksanaan program ke program utama dengan cara mengambil isi *stack* yang disimpan pada saat instruksi *CALL*, dan selanjutnya dimasukkan ke dalam PC (*Program Counter*).

Tabel 2.2 *Special Function Register*

Simbol	Nama	Alamat	Power-on Reset
ACC	Akumulator	E0h	0000 0000
B	Register B	F0h	0000 0000
PSW	Program Status Word	D0h	0000 0000
SP	Stack Pointer	81h	0000 0111
DPTR	Data Pointer 16 bit		
DPL	Data Pointer byte rendah	82h	0000 0000
DPH	Data Pointer byte tinggi	83h	0000 0000
P0	Port 0	80h	1111 1111
P1	Port 1	90h	1111 1111
P2	Port 2	A0h	1111 1111
P3	Port 3	B0h	1111 1111
IP	Interrupt Priority Control	B8h	xxx0 0000
IE	Interrupt Enable Control	A8h	0xx0 0000
TMOD	Timer/counter Mode Control	89h	0000 0000
TCON	Timer/counter Control	88h	0000 0000

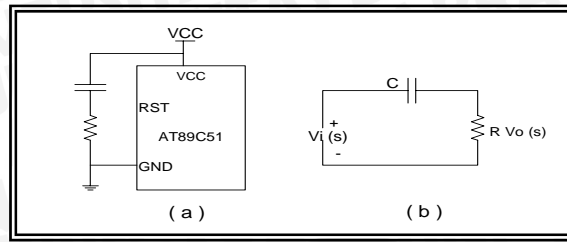
TH0	Timer/counter 0 high byte	8Ch	0000 0000
TL0	Timer/counter 0 low byte	8Ah	0000 0000
TH1	Timer/counter 1 high byte	8Dh	0000 0000
TL1	Timer/counter 0 low byte	8Bh	0000 0000
SCON	Serial Control	98h	0000 0000
SBUF	Serial Data Buffer	99h	xxxx xxxx
PCON	Power Control	87h	0xxx xxxx

Sumber: Agfianto Eko Putra, 2002 : 189

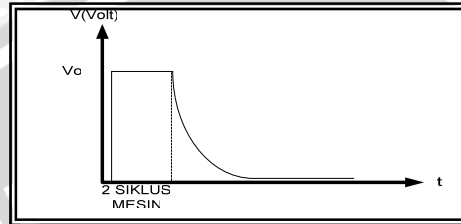
- *Data pointer* (DTPR) terdiri dari dua register, yaitu untuk *byte* rendah (*Data Pointer Low/DPL*) dan untuk *byte* tinggi (*Data Pointer High/DPH*). Fungsinya adalah untuk menahan alamat 16 bit. DPTR dapat dimanipulasi sebagai register 16 bit atau sebagai dua buah register 8 bit.
- Port 0 sampai port 3 merupakan register yang berfungsi untuk membaca dan mengeluarkan data pada port 0, 1, 2, dan 3. Masing-masing register ini dapat dialamti secara per-bit maupun per-*byte*.
- *Control register*, terdiri atas register yang mempunyai fungsi kontrol untuk mengontrol sistem interupsi, terdapat dua register khusus, yaitu register IP (*Interrupt Priority*) dan register IE (*Interrupt Enable*). Untuk mengontrol pelayanan *timer/control* terdapat register khusus yaitu register TMOD (*Timer/Counter Mode Control*) dan register TCON (*Timer/counter Control*), serta untuk pelayanan port serial menggunakan register SCON (*serial port control*).

2.1.5 Reset

Rangkaian *power on reset* diperlukan untuk mereset mikrokontroler secara otomatis setiap catu daya *on*. Gambar 2.11 menunjukkan rangkaian *power on reset*. Ketika catu daya diaktifkan, rangkaian *reset* menahan logika tinggi pin RST dengan jangka waktu yang ditentukan oleh besarnya pengisian muatan C. Sinyal yang dihasilkan dapat kita lihat dalam Gambar 2.12



Gambar 2.11 (a) Rangkaian *Power-on Reset* (b) Rangkaian Ekuivalen
 Sumber: Agfianto Eko Putra, 2002 : 22



Gambar 2.12 Sinyal Keluaran *Power-on Reset*
 Sumber: Anymous, 2004

Dari rangkaian ekivalen seperti ditunjukkan dalam Gambar 2.11 (b) maka didapat persamaan sebagai berikut:

$$V_o = \frac{R}{R + 1/Cs} \times V_i(s) = \frac{RCs}{RCs + 1} \times V_i(s)$$

dengan tegangan V_i adalah tegangan V_{cc} yaitu 5V, dalam fungsi Laplace adalah $5/s$ sehingga :

$$V_o = \frac{RCs}{RCs + 1} \times \frac{5}{s} = \frac{RC}{RCs + 1} \times 5 = 5 \times \left[\frac{1}{s + 1/RC} \right] = 5 \cdot e^{-t/RC}$$

$$\frac{5}{V_o(t)} = e^{-t/RC} \text{ sehingga } \ln \frac{5}{V_o} = \frac{t}{RC}$$

$$\Rightarrow t = R \times C \left(\ln \frac{5}{V_o} \right)$$

Untuk memastikan keabsahan reset, logika tinggi harus ditahan lebih dari 2 siklus mesin (24 periode osilator).

2.2 PC (Personal Computer)

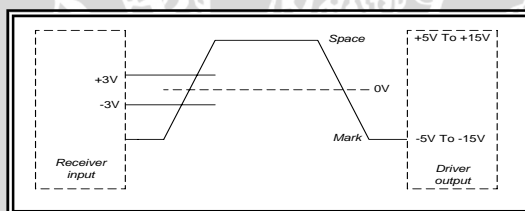
PC (*Personal Computer*) berfungsi sebagai pusat database Lembar Hasil Studi dan Lembar Rencana Studi mahasiswa. Di samping itu komputer berfungsi

sebagai tempat menyimpan data mahasiswa yang telah mengambil Lembar Hasil Studi dan Lembar Rencana Studi. Dengan demikian data tersebut hanya dapat dibaca 1 kali. Dalam perancangan di atas komputer tersebut akan melayani 1 buah terminal yang dikontrol oleh Mikrokontroler AT89C51

2.3 Komunikasi serial antara PC dan MCU.

Komunikasi serial dari sistem ini menggunakan RS-232 yang merupakan seperangkat alat yang berfungsi sebagai *interface* dalam proses transfer data secara serial. Metode pengiriman secara serial RS-232 adalah asinkron. Pengiriman asinkron berarti tidak membutuhkan pewaktu sebagai sinkronisasi. Dalam pengiriman serial *asinkron*, *clock* tidak dikirimkan, tetapi dikondisikan oleh *timing start bit* yang merupakan isyarat dari sumber ke tujuan untuk mengkodekan adanya pengiriman karakter sudah selesai dikirim.

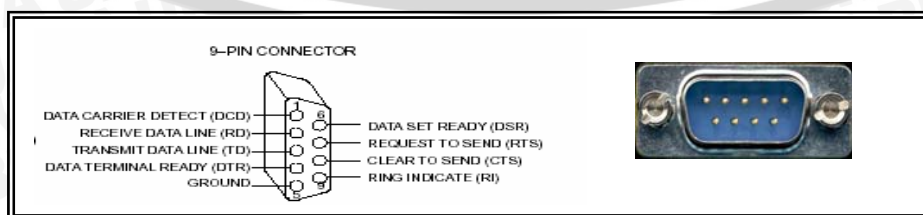
Karakteristik elektrik dari sistem RS-232 adalah mempunyai tegangan keluaran antara -15 Volt sampai dengan +15 Volt. Tegangan +5 sampai +15 volt untuk mewakili level rendah (logika '0' / *spacing*) dan tegangan -5 sampai -15 volt untuk mewakili level tinggi (logika '1' / *marking*). Hal tersebut seperti ditunjukkan dalam Gambar 2.13.



Gambar 2.13 Level Logika Standar RS-232

Sumber : Dallas Semiconductor, 1998: 2.

Di dalam komputer terdapat fasilitas komunikasi serial yang menggunakan standar RS-232, yaitu terletak pada COM1 dan COM2. Kedua fasilitas ini menggunakan konektor DB9 atau DB25 sebagai penghubung dengan piranti luar. Gambar konektor DB9 seperti terdapat dalam Gambar 2.14.



Gambar 2.14 Konfigurasi Pin Konektor RS-232

Sumber : Dallas Semiconductor, 1998: 3-9

Fungsi masing-masing pin seperti terdapat dalam Tabel 2.3.

Tabel 2.3. Fungsi Pin RS-232 dalam DB9

Pin	Nama	Fungsi
1	DCD (<i>Data Carrier Detect</i>)	Mendeteksi sinyal <i>carrier</i> dari modem lain
2	RD (<i>Receive Data Line</i>) / (R_xD)	Pengiriman data serial dari DCE ke DTE
3	TD (<i>Transmit Data Line</i>) / (T_xD)	Pengiriman data serial dari DTE ke DCE
4	DTR (<i>Data Terminal Ready</i>)	Memberitahu DCE bahwa DTE telah aktif dan siap untuk bekerja
5	<i>Ground</i>	Referensi semua tegangan antarmuka
6	DSR (<i>Data Set Ready</i>)	Memberitahu DTE bahwa DCE telah aktif dan siap untuk bekerja
7	RTS (<i>Request To Send</i>)	Memberitahu DCE bahwa DTE akan mengirim data
8	CTS (<i>Clear To Send</i>)	Memberitahu DTE bahwa DCE siap menerima data
9	RI (<i>Ring Indikator</i>)	Aktif jika modem menerima sinyal ring pada jalur telepon

Sumber : Ganiadi Gunawan, 1991: 4.

Spesifikasi RS-232 dapat dilihat dalam Tabel 2.4.

Tabel 2.4 Spesifikasi RS-232

Keistimewaan	Karakteristik
Jenis operasi	<i>Single ended</i> (tak seimbang)
Jenis penggerak dan Penerima per jalur	1 <i>driver</i> 1 <i>receiver</i>
Data rate maksimum	20 kbps
Panjang saluran maksimum	50 ft (15 m)
Tegangan keluaran penggerak	$\pm 5 - \pm 15$ volt
Sensitivitas penerima	± 3 volt

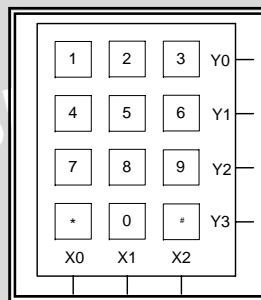
Sumber : Dallas Semiconductor, 1998

2.4 Keypad

Tombol masukan biasa disebut juga dengan *keypad*. Pada alat ini *keypad* tersebut difungsikan untuk memberikan masukan data melalui tombol-tombol yang terdapat pada papan *keypad* tersebut. *Keypad* ini berfungsi untuk menerjemahkan penekanan pada salah satu tombol ke dalam bentuk biner. Sehingga jumlah jalur yang akan masuk ke rangkaian berikutnya dapat diperkecil. *Keypad* matrik ini bekerja dengan menggunakan prinsip *scanning* pada baris dan kolom. Empat bit sebagai *input* dan empat bit yang lain sebagai *output*. Jika terdeteksi adanya persambungan antara baris dan kolom yang *valid*,

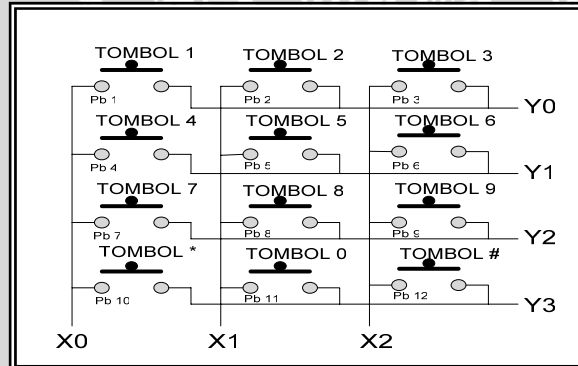
software akan mendekodekan baris dan kolom mana yang menyambung menjadi data biner.

Pada kondisi tidak terjadi penekanan tombol *keypad*, kondisi logika pada port adalah logika 1 pada setiap bitnya. Saat salah satu tombol *keypad* ditekan, baris dan kolom yang berhubungan akan terhubung, sehingga kondisi baris dan kolom tersebut akan berlogika 0. Konfigurasi pin *keypad* ditunjukkan pada Gambar 2.15 Dan rangkaian ekuivalen *keypad* ditunjukkan pada Gambar 2.16.



Gambar 2.15. Konfigurasi Pin *Keypad*

Sumber: Paulus Andi Nalwan, 1998: 132



Gambar 2.16. Rangkaian Ekuivalen *Keypad*

Sumber: Paulus Andi Nalwan, 1998: 132

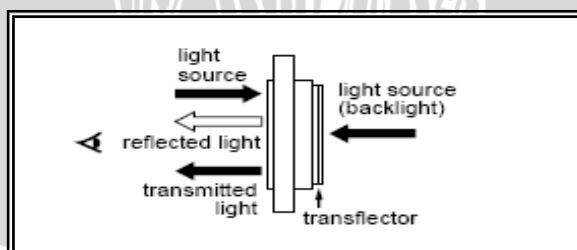
Sebagai contoh aplikasinya, seluruh pin *keypad* dapat langsung dihubungkan pada port 1 MCU (*MicroController Unit*) AT89C51, yang berfungsi sebagai unit *input* atau *output* biasa dan memiliki *pull up* internal. Catu daya *keypad* berasal dari logika “1” pada port 1. Sebagian dari port 1 difungsikan sebagai penerima input data dari *keypad*, untuk itu port 1 di-*setting* berlogika 1. Dan sebagian yang lain difungsikan untuk proses *scanning*.

Misalnya, Y0-Y3 dihubungkan ke P1.0-P1.3 (berlogika 1), sedangkan X0-X2 dihubungkan ke P1.4-P1.6 dengan X0 berlogika 0, X1 dan X2 berlogika 1. Berdasarkan program pada MCU (*Microcontroller Unit*) akan dilakukan proses *scanning* pada pin X0-X2, dengan cara melakukan penggeseran logika 0 secara bergantian pada pin X0, X1, dan X3, hingga di deteksi adanya penekanan pada salah satu tombol *keypad*.

Apabila salah satu tombol ditekan, kolom dan baris yang berhubungan akan terhubung. Misalnya, tombol angka 1 ditekan. Dengan proses *scanning* akan terdeteksi adanya penekanan tombol pada X0, sehingga X0 berlogika 0. Baris yang terhubung dengan kolom X0 adalah Y0, sehingga Y0 menjadi berlogika 0. Kombinasi logika 0 pada X0 dan Y0 mengisyaratkan bahwa tombol angka 1 yang telah ditekan.

2.5 LCD (*Liquid Crystal Display*) M1632

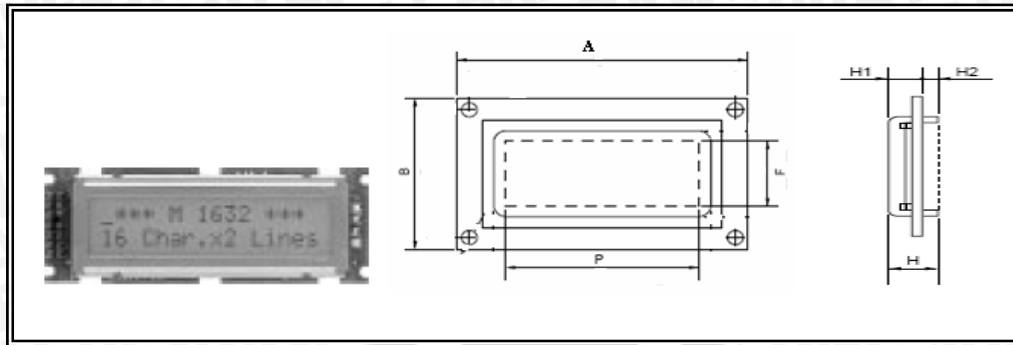
LCD (*Liquid Crystal Display*) adalah salah satu jenis penampil yang digunakan untuk menampilkan angka, karakter atau bahkan angka dan karakter. LCD terdiri atas tumpukan tipis atau sel dari dua lembar kaca dengan pinggiran yang tertutup rapat. Di antara dua lembar kaca tersebut diberi bahan kristal cair (*liquid crystal*) yang tembus cahaya dimana akan beremulsi apabila diberi tegangan. Transfektor dilekatkan pada bagian belakang sebagai polarisator refleksi cahaya dari depan, seolah-olah cahaya datang dari belakang. Prosesnya dapat kita lihat dalam Gambar 2.17.



Gambar 2.17. Proses tampilan LCD

Sumber: *Module LCD M1632*, 2002

Permukaan luar dari masing-masing keping kaca mempunyai lapisan penghantar tembus cahaya. Sel mempunyai ketebalan sekitar 1×10^{-5} meter dan di isi dengan kristal cair. Dimensi LCD dapat kita lihat dalam Gambar 2.18



Gambar 2.18 Bentuk dan Dimensi LCD M1632

Sumber: *Module LCD M1632, 2002*

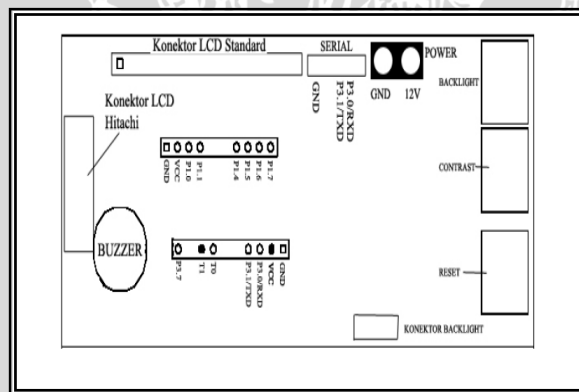
Ukuran LCD : $A \times B \times H = 85 \times 30 \times 10,1 \text{ mm}^3$

Tampilan Area : $P \times F = 62 \times 16 \text{ mm}^2$

Ukuran area tiap karakter : $2,76 \times 4,27 \text{ mm}^2$

Ukuran Dot : $0,5 \times 0,55 \text{ mm}^2$

Penampil kristal cair berupa dot matrik 5 x 7, tersusun sebanyak dua baris dan masing-masing baris terdiri atas 16 karakter. Untuk lebih jelas dapat kita lihat Konfigurasi pin pada LCD (*Liquid Crystal Display*) ditunjukkan dalam Gambar 2.19



Gambar 2.19. LCD (*Liquid Crystal Display*) dan Konfigurasi Pinnya

Sumber: M1632 User Manual, 1987:7

Fungsi dari masing-masing penyemat LCD (*Liquid Crystal Display*) M1632 ditunjukkan sebagai berikut:

- **D0....D7:** Merupakan saluran data, berisi perintah dan data yang akan ditampilkan pada LCD.
- **Enable (E):** Sinyal operasi awal. Sinyal ini akan mengaktifkan data tulis atau baca.



- **R/W** : Sinyal seleksi tulis dan baca: 0 = tulis 1 = baca
- **RS** :Sinyal pemilih register internal (0 = instruksi register (tulis); 1 = data register (tulis dan baca)
- **V_{EE}**: Untuk mengendalikan kecerahan LCD dengan mengubah-ubah nilai resistor variabel yang dihubungkan padanya.
- **V_{CC}**: Catu daya +5 volt.
- **V_{SS}**: Terminal *ground*.

2.6 Barcode Reader

Barcode Reader berfungsi sebagai pembaca data *barcode* yang dimiliki oleh setiap mahasiswa dalam kartu mahasiswa dan mengirimnya ke sistem mikrokontroler

Barcode adalah suatu metode dalam menggantikan karakter memakai susunan balok – balok *vertical* yang berdasar system bilangan biner, sehingga kode bar dapat secara langsung diakses.

Barcode ini terdiri atas dua unsur, yaitu bar dan spasi. *Bar* adalah suatu garis hitam *vertical*, sedangkan spasi merupakan jalur putih *vertical* antara dua bar. Bentuk kode bar ini mempunyai tebal yang berbeda – beda. Variasi tebal tipis kode bar inilah yang memuat informasi yang akan dibaca.

Kata "*reader*" dan "*scanner*" seringkali digunakan untuk menjelaskan peralatan yang menggunakan pembacaan *barcode*. Pada Umumnya, *barcode scanner* hanya didesain dari bagian peralatan optoelektronik yang mentransformasikan *optical image* dari *barcode* ke dalam sinyal elektrik.. *Barcode Reader* terdiri atas *barcode scanner* dan dekoder yang mentransformasikan sinyal elektrik dari *scanner* ke dalam kode ASCII (*American Standard Code for Information Interchange*).

Barcode Reader membaca dengan menyapu semua garis tegak (bar) yang tercetak (berupa simbol *barcode*) yang dilewatinya Pembacaan dimulai dari area putih sebelum bar pertama dan berlanjut pada bar berikutnya, dan berakhir pada area putih setelah bar terakhir. Karena sebuah *bar code* tidak dapat terbaca jika penyapuan melewati daerah diluar area simbol *barcode*, maka pemilihan ketinggian bar diperlukan untuk mempermudah penyapuan pada *barcode* area. *Barcode* yang digunakan, barcode dengan tipe CCD, SD 1000 Series.

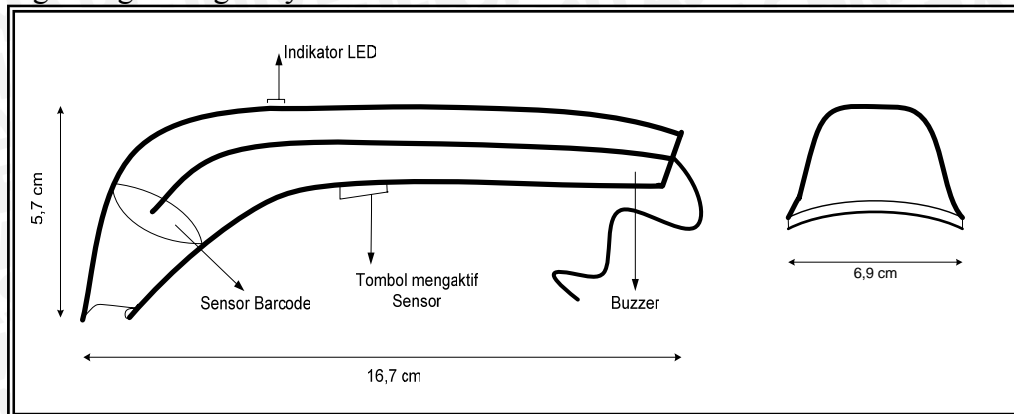
Tabel 2.5. Spesifikasi Barcode tipe SD 1000 Series

PERFORMA PEMBACAAN	
Kode Pembacaan:	<ul style="list-style-type: none"> • Code 39 (Standard / Full ASCII) • Italy Pharmacode • French Pharmacode • Industrial 25 • Interleave 25 • Matrix 25 • Codabar (NW-7) • UPCA (with or without Addon) • UPCE (with or without Addon) • EAN8 (with or without Addon) • EAN13 (with or without Addon) • Code 93 • Code 128 • EAN 128 • MSI • Plessey
OPTIK	
Resolusi:	0.125mm
Nilai PCS:	≥ 0.45
Lebar Jendela:	67mm
Lebar Scanning:	Maks. 85mm (PCS= 0.9)
Tebal :	0-29mm (PCS= 0.9)
Pencahayaannya:	$\lambda=660\text{nm}$ Red LED
Photo sensor:	2048 pixel CCD image sensor
Scan Rate:	100 Scans/second
Indikator Pembacaan:	Red LED and adjustable volume buzzer
INTERFACE	
Sinyal Interface:	Keyboard Wedge(PS2) / RS232/ Other available on request
Tipe Konektor:	Din 5P / Mini Din-6P / Dsub-9P / Dsub-25P/ Other available on request.
KELISTRIKAN	
Sumber Tegangan:	DC 5V+/-10%
Arus Kerja maksimal	85 mA
Arus kerja Typical	75 mA
EMI:	FCC, CE, Class A approved
PENYIMPANAN	
Temperatur Kerja:	Temperature 0°C~50°C Humidity 20%-90%
Temperatur Penyimpanan:	Temperature 20°C~60°C Humidity 10%~95%
BENTUK FISIK	
Berat:	165g (tanpa kabel)
Kabel:	1.75m Straight/Coiled replaceable cable

Sumber : Manual Book CCD SD 1000 Series, 2005



Dalam Gambar 2.20 dapat kita lihat ukuran barcode reader CCD SD 1000 series dengan bagian-bagiannya



Gambar 2.20 Ukuran dan Bagian-bagian Barcode Reader CCD SD 1000 series

Sumber : *Manual Book CCD SD 1000 Series*, 2005

Cara penulisan barcode menggunakan Barcode *Uniform Symbol Description-1* (USD-1) yang hanya dikodekan untuk angka, yang mewakili 10 digit (0 sampai 9) dan 2 kode untuk kode start dan kode stop. Tabel 2.6 memperlihatkan pengkodean angka ke dalam bentuk kode USD-1.

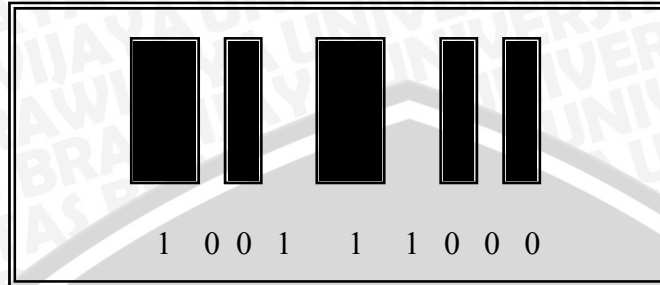
Tabel 2.6 Karakter Barcode USD-1

Data	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5
0	0	0	1	1	0
1	1	0	0	0	1
2	0	1	0	0	1
3	1	1	0	0	0
4	0	0	1	0	1
5	1	0	1	0	0
6	0	1	1	0	0
7	0	0	0	1	1
8	1	0	0	1	0
9	0	1	0	1	0

Sumber : Solomon, 1994:244

Karakter USD-1 disusun oleh 5 bit yang terdiri dari 2 bit tinggi (1) dan 3 bit rendah (0). Suatu elemen bentuk tebal baik bar maupun spasi akan selalu

diartikan sebagai logika tinggi (bit tinggi), sedangkan bentuk tipis baik bar atau spasi akan selalu diartikan logika rendah (bit rendah). Gambar 2.21 memperlihatkan cara pembacaan kode USD-1.



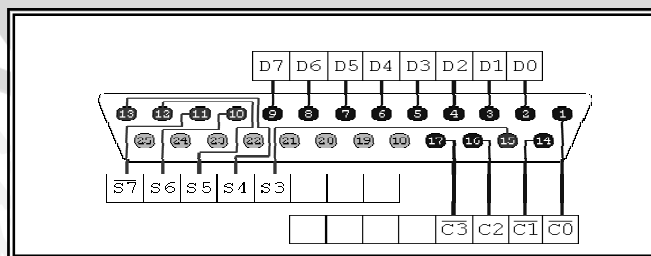
Gambar 2.21 Pembacaan Kode USD-1

Sumber: Solomon, 1994:245

2.7 Port Printer

LPT1 atau *port printer* sebenarnya terdiri dari tiga bagian besar yang masing-masing diberi nama sesuai dengan tugasnya dalam melaksanakan tugas pencetakan. Tiga bagian tersebut adalah DP (*Data Port*), PC (*Port Control*), dan PS (*Port Status*). *Data Port* (DP) digunakan oleh komputer dalam hal ini CPU (*Central Processing Unit*) untuk mengirimkan data biner dari karakter (data standar ASCII) yang harus dicetak oleh printer. PC (Personal Computer) mempunyai tugas mengirimkan kode-kode pengontrolan dari CPU ke printer.

DP, PC dan PS sebenarnya adalah port-port 8 bit namun hanya DP yang benar-benar port 8 bit. Untuk PC dan PS karena hanya beberapa bit saja yang dibutuhkan dalam proses pencetakan, maka bit-bit yang lain tidak digunakan yang berarti hanya beberapa bit saja dari port-port ini yang dapat kita manfaatkan untuk keperluan pengontrolan nanti. DP dan PC adalah port baca/tulis (*read/write*) sedangkan PS adalah port baca saja (*read only*). Konfigurasi dari slot untuk printer atau slot DB-25 dapat dilihat dalam Gambar 2.22

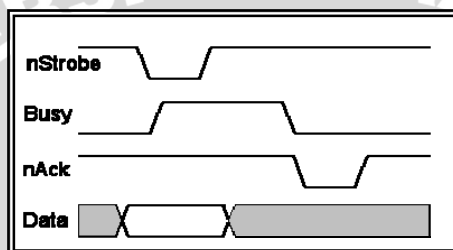


Gambar 2.22 Konektor DB 25 port Printer

Sumber: <http://www.beyondlogic.org/spp/parallel.htm>

Sedangkan proses jabat-tangan antara komputer dengan *printer* (Gambar 2.23) dapat digambarkan sebagai berikut :

Sebelum komputer mengirimkan datanya ke printer, ia akan melihat kondisi printer, sibuk atau tidak. Bila sibuk ditandai dengan tingginya sinyal *BUSY*. Jika tidak sedang sibuk, setiap pengiriman data 8 bit akan disertai dengan rendahnya sinyal *STROBE*. Segera setelah menerima data, printer mengirimkan data berikutnya, sampai data tersebut selesai disimpan pada buffer memorinya. Jika data tersebut sudah dipahami dan disimpan pada buffer memori printer akan mengirimkan sinyal *ACK* (acknowledge), yang berarti printer siap menerima data berikutnya.



Gambar 2.23 Proses Jabat-tangan (*handshaking*)

Sumber: <http://www.beyondlogic.org/spp/parallel.htm>

Port printer dapat digunakan untuk pengontrolan peralatan dalam hal ini mengirim dan menerima data. Untuk hal tersebut yang perlu diketahui adalah dimana alamat register dari port tersebut. Alamat dari register yang biasanya digunakan LPT1 dapat dilihat dalam Tabel 2.7.

Tabel 2.7 Alamat Register port pada LPT1

LPT1	Alamat Register
DP (Data Port)	378
PS (Port Status)	379
PC (Port Control)	37A

Sumber : <http://www.beyondlogic.org/spp/parallel.htm>

Alamat ini sangat penting karena bahasa program nantinya akan mengirim dan membaca data ke dan dari port dengan statemen yang menyebutkan alamat register. Statemen berikut ini merupakan contoh statemen yang digunakan untuk mengirim data ke suatu port dengan bahasa pemrograman Delphi :



Port [alamat register] := data yang dikirim;

Sedangkan yang berikut ini adalah contoh statemen yang digunakan untuk membaca data dari suatu *port* :

Data yang diterima = *Port* [alamat register];

Letak dari *Data Port*, *Port Status* dan *Port Control* dapat dilihat dalam Tabel 2.8.

Tabel 2.8 Tabel *Data Port*, *Status Port* dan *Control Port*

Pin	Sinyal SPP	Arah I/O	Register
1	$\overline{\text{Strobe}}$	<i>In/Out</i>	<i>Control</i>
2	Data 0	<i>Out</i>	Data
3	Data 1	<i>Out</i>	Data
4	Data 2	<i>Out</i>	Data
5	Data 3	<i>Out</i>	Data
6	Data 4	<i>Out</i>	Data
7	Data 5	<i>Out</i>	Data
8	Data 6	<i>Out</i>	Data
9	Data 7	<i>Out</i>	Data
10	$\overline{\text{Ack}}$	<i>In</i>	Status
11	<i>Busy</i>	<i>In</i>	Status
12	<i>Paper-Out / Paper-End</i>	<i>In</i>	Status
13	<i>Select</i>	<i>In</i>	Status
14	$\overline{\text{Auto - LineFeed}}$	<i>In/Out</i>	<i>Control</i>
15	$\overline{\text{Error/Foult}}$	<i>In</i>	Status
16	$\overline{\text{Initialize}}$	<i>In/Out</i>	<i>Control</i>
17	$\overline{\text{Select - Printer/In}}$	<i>In/Out</i>	<i>Control</i>
18-25	<i>Ground</i>	Gnd	

Sumber: <http://www.beyondlogic.org/spp/parallel.htm>

Penjelasan pin-pinnya adalah sebagai berikut:

- $\overline{\text{Strobe}}$, sinyal ini digunakan untuk memberitahu printer bahwa karakter sudah dikirimkan. Sinyal ini aktif rendah. Lebar pulsa harus lebih dari 0,5 μs pada terminal penerima.
- Data_n , sinyal-sinyal ini merupakan sinyal informasi selebar 8 bit (1 karakter). Tiap-tiap sinyal akan bernilai tinggi untuk data logika '1' dan bernilai rendah untuk logika '0'.
- $\overline{\text{Ack}}$, sinyal ini diberikan printer sebagai jawaban sinyal $\overline{\text{Strobe}}$. Logika rendah menunjukkan bahwa data yang dikirimkan telah diterima printer. Lebar pulsa harus lebih dari 0,5 μs .
- *Busy*, sinyal ini berisi informasi kondisi printer yang tidak bisa menerima data. Kondisi ini disebabkan karena printer sedang menerima data, dalam proses mencetak dan dalam kondisi *offline*.
- *Paper-Out / Paper-In*, kondisi tinggi menunjukkan printer kehabisan kertas.
- *Select*, kondisi tinggi menunjukkan printer sedang dalam proses pemilihan (*selected state*).
- $\overline{\text{Auto-LineFeed}}$, kondisi rendah akan memberitahu printer untuk menambah 1 baris lagi setelah pencetakan selesai.
- $\overline{\text{Error/Fault}}$, sinyal ini akan berlogika rendah jika printer pada kondisi *Paper-Out / Paper-In*, *offline* dan *error*.
- *Initialize*, logika rendah akan memberi perintah mereset printer.
- $\overline{\text{Select-Printer/In}}$, pengiriman data ke printer hanya dapat terjadi kalau sinyal ini berlogika rendah.

BAB III METODOLOGI

Pelaksanaan skripsi ini dilaksanakan dalam beberapa tahapan pembahasan yang tersusun atas beberapa bab. Tahapan tersebut tersusun dalam suatu metodologi penelitian, yaitu sebagai berikut:

3.1 Kajian Pustaka

Melaksanakan kajian terhadap pustaka atau literatur mengenai hal-hal yang diperlukan dalam penyelesaian penyusunan skripsi ini dan digunakan untuk mempelajari cara kerja komponen-komponen penyusun rangkaian secara keseluruhan dan juga teori-teori pendukung sesuai dengan alat yang dirancang sehingga dapat bekerja dengan baik. Hal – hal yang dipelajari tersebut adalah ;

1. Sistem Mikrokontroler AT89C51
2. PC (*Personal Computer*)
3. Komunikasi serial antara PC dan MCU.
4. *Keypad*
5. LCD (*Liquid Crystal Display*) M1632
6. *Barcode Reader*
7. *Printer Dot Matrik Epson LX-800*

3.2 Perencanaan dan Pembuatan Alat

Dalam pembahasan tahap ini terdapat dua bagian yaitu tahap perencanaan alat dan tahap penyelesaian alat dimana dua bagian tersebut sangat berkaitan erat dalam spesifikasi dan pembuatannya.

3.2.1 Perencanaan Alat

Perencanaan alat ini meliputi penentuan spesifikasi alat yang akan dibuat yang mengacu pada perencanaan blok diagram sistem pada piranti keras dan perancangan perangkat lunak mikrokontroler menggunakan bahasa *assembly* dan PC menggunakan bahasa pemrograman.

3.2.1.1 Perancangan piranti lunak terdiri atas 2 bagian yaitu:

- Program pada sistem Mikrokontroler AT89C51 terdiri dari beberapa prosedur program. Prosedur yang diperlukan dalam perancangan sistem adalah:
 - Prosedur inisialisasi sistem dengan menyalakan masing-masing subsistem, apakah siap untuk dijalankan.
 - Prosedur baca data barcode yang ada pada KTM untuk mengakses NIM
 - Prosedur kirim data barcode ke PC melalui komunikasi serial
 - Prosedur terima data dan kode akses dari PC, apakah data yang diakses pernah diambil sebelumnya.
 - Prosedur baca data *keypad* untuk kode akses
 - Prosedur cek kode akses, bila sudah 3 x salah akan mengulang ke awal yaitu masukan kartu
 - Prosedur cetak data ke *printer*.
- Program PC ditulis dengan bahasa pemrograman Borland Delphi dalam pengolahan data

3.2.1.2 Perancangan piranti keras yaitu:

Pemasangan piranti keras sesuai dengan blok diagram system

3.2.1.3 Spesifikasi alat

- Menggunakan sistem mikrokontroler AT89C51
- Tegangan catu: DC +5V sistem terminal
- Menggunakan tampilan berupa LCD M1632 16 karakter x 2 baris
- Menggunakan komputer PC untuk pengolah database dan komunikasi serial RS 232
- Masukan sistem untuk NIM menggunakan pembacaan barcode *reader* CCD 1000 Series dan masukan sistem untuk kode akses menggunakan tombol *keypad*
- Keluaran sistem menggunakan *printer dot matrik* Epson LX-800

3.3 Pengujian Alat

Sebelum direalisasikan dalam bentuk PCB, rangkaian yang telah dirancang perlu dicoba dahulu dengan menggunakan *protoboard*. Hasil pengujian dapat diketahui dengan menggunakan multimeter atau menggunakan *logic probe* lewat *point-point* (titik-titik) keluaran yang dapat dideteksi. Sehingga perubahan jenis dan nilai komponen dapat dilakukan dengan mudah.

Pengujian tahap awal tersebut dapat dilakukan dengan menguji tiap blok dari sistem yang dibuat. Dari pengujian tersebut didapatkan spesifikasi dari tiap blok sehingga memenuhi spesifikasi dari sistem yang direncanakan.

Selanjutnya adalah menguji keseluruhan dari sistem untuk mendapatkan kondisi kerja alat, sehingga dapat diketahui ketepatan alat serta unjuk kerja alat yang dibuat.

Setelah pengujian pada *protoboard* selesai, maka keseluruhan komponen siap dipindahkan ke dalam PCB. Perakitan perangkat keras ini dilakukan tepat seperti rancangan yang telah dibuat dan telah diset dengan *protoboard*.

Setelah rangkaian tiap blok system selesai direalisasikan pada PCB, selanjutnya adalah menggabungkan sub-sub system yang telah dibuat menjadi satu kesatuan system sehingga membentuk kinerja sebuah alat secara menyeluruh, tepat seperti yang direncanakan. Selanjutnya data yang telah terkumpul dari kegiatan pengujian alat akan dicatat untuk kemudian dilakukan proses analisa data, dengan tujuan mengetahui apakah sistem memenuhi perencanaan.

3.4 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dan saran dilakukan setelah didapatkan data hasil pengujian alat. Data tersebut menentukan unjuk kerja alat baik dari segi kualitas, akurasi, fungsi alat serta mekanisme operasional alat yang sebenarnya sehingga diperoleh masukan untuk kesimpulan dan saran.

BAB IV

PERANCANGAN ALAT

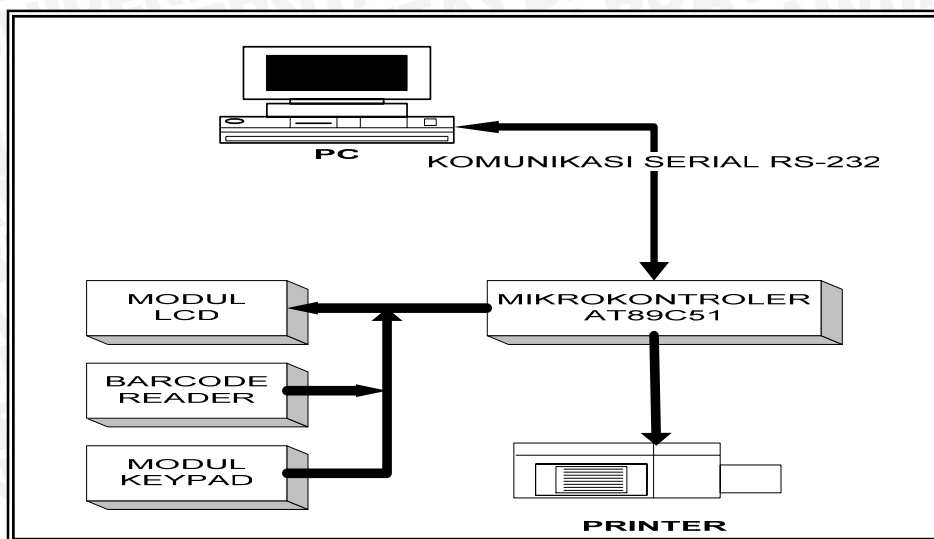
Dalam perancangan alat ini dilakukan bertahap blok demi blok untuk memudahkan penganalisaan sistem tiap bagian maupun sistem secara keseluruhan. Perancangan dan pembuatan sistem ini terdiri dari dua bagian yaitu perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*). Beberapa aspek lain yang perlu dijelaskan dalam pembahasan bab ini adalah penentuan spesifikasi dari sistem yang dirancang, blok diagram sistem serta prinsip kerja sistem.

4.1 Spesifikasi Alat

Sistem Otomatisasi Pengambilan Lembar Hasil Studi Dan Lembar Rencana Studi Setiap Semester berbasis Mikrokontroler AT89C51 Dengan menggunakan PC sebagai Basis Data yang akan dirancang mempunyai spesifikasi teknis sebagai berikut:

- ✚ Menggunakan sistem mikrokontroler AT89C51
- ✚ Tegangan catu: DC +5V sistem terminal
- ✚ Menggunakan tampilan berupa LCD 16 karakter x 2 baris monitor PC
- ✚ Menggunakan komputer PC untuk pengolah database dan komunikasi serial RS-232
- ✚ Masukan sistem untuk NIM menggunakan pembacaan *barcode reader CCD 1000 series* dan masukan sistem untuk kode akses menggunakan tombol *keypad M1632*
- ✚ Keluaran sistem menggunakan printer dot matrik Epson LX-800
- ✚ Perangkat lunak berupa bahasa Mikrokontroler AT89C51 dan Delphi

Blok diagram Sistem Otomatisasi Pengambilan Lembar Hasil Studi Dan Lembar Rencana Studi Setiap Semester berbasis Mikrokontroler AT89C51 Dengan menggunakan PC sebagai Basis Data dapat dilihat dalam Gambar 4.1



Gambar 4.1 Blok diagram sistem otomatisasi pengambilan Lembar Hasil dan Rencana Studi

4.2 Prinsip Kerja Alat

Prinsip kerja sistem otomatisasi pengambilan Lembar Hasil dan Rencana Studi meliputi beberapa tahapan yaitu, menyiapkan sistem, menyiapkan database.

Menyiapkan Sistem

1. Hubungkan terminal RS 232 pada konektor serial PC melalui kabel serial
2. Hubungkan printer pada konektor printer terminal lalu hidupkan printer.
3. Siapkan kertas khusus pada printer untuk pencetakan modus yang diinginkan.
4. Hidupkan PC dan jalankan program aplikasi “Sistem Otomatisasi Pengambilan Lembar Hasil dan Rencana Studi

Menyiapkan database

Masukan file database sumber dari pusat data ke PC dan konversi. Mengaktifkan Modus Pengambilan. Setelah PC melakukan inisialisasi, terminal menunggu pemasukan kode pada *Barcode*. Terminal yang telah siap menerima masukan dapat dipakai mahasiswa untuk mengambil Lembar Hasil atau Rencana Studi secara terpisah tanpa melalui petugas loket.

Cara pemakaian terminal tersebut adalah sebagai berikut:

Mahasiswa yang ingin mengambil Lembar Hasil dan Rencana Studi memasukan NIM-nya dengan bantuan Kartu mahasiswa yang diberi barcode yang digesekan melalui pembaca barcode pada terminal. Kemudian melihat pada LCD untuk keterangan status NIM nya. Jika NIM benar dan ia belum pernah melakukan pengambilan sebelumnya maka ia diminta untuk memasukan kode

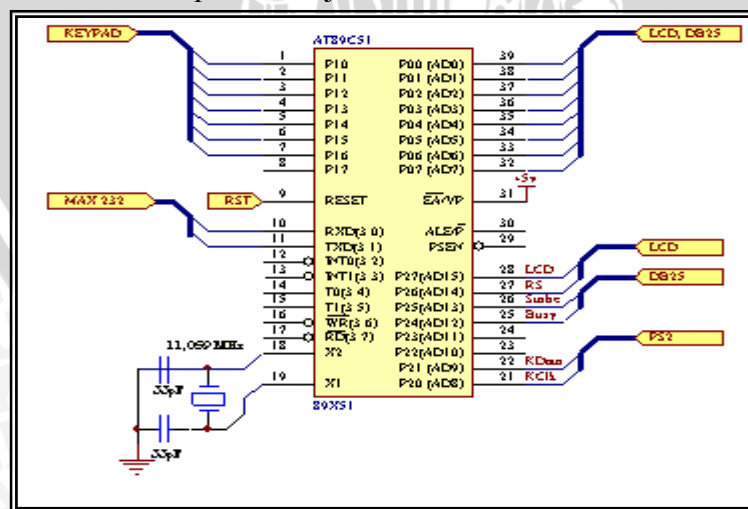
akses melalui keypad. Sedangkan jika NIM tidak benar atau ia telah melakukan pengambilan sebelumnya maka ia akan ditolak dengan keterangan pada LCD bahwa pengambilan tidak sah. Jika pada saat memasukan kode akses terjadi kesalahan maka ia dapat mengulanginya dengan menekan tombol bintang (*). Pengulangan masukan kode akses maksimal adalah 3 kali, setelah itu sistem akan meminta untuk memasukan NIM kembali. Setelah memasukan kode akses diterima, mahasiswa dapat mengambil Lembar Hasil atau Rencana Studi langsung dari alat pencetak (printer) yang telah disediakan.

4.3 Perancangan Perangkat Keras

Perancangan perangkat keras merupakan penjabaran dari masing-masing blok diagram sistem. Perancangan perangkat keras terdiri dari perancangan rangkaian kontrol menggunakan AT89C51, perancangan rangkaian LCD (*Liquid Crystal Display*), serta perancangan rangkaian keypad, pemasangan barcode dan pemasangan printer.

4.3.1 Antarmuka Mikrokontroler AT89C51

Pada rangkaian kontrol ini komponen utamanya menggunakan mikrokontroler AT89C51. Sebagai tempat dari pengolahan data dan pengontrolan alat, pin-pin AT89C51 dihubungkan pada rangkaian pendukung membentuk suatu minimum sistem seperti ditunjukkan dalam Gambar 4.2.



Gambar 4.2. Rangkaian Antarmuka Mikrokontroler AT89C51

Pin-pin mikrokontroler yang digunakan yaitu:

1. Port 0

P0.0-P0.7 dihubungkan secara paralel dengan *DB25* (pin AD0-AD7) dan *Liquid Crystal Display* (pin Q0-Q7).

2. Port 1

P1.0-P1.6 dihubungkan pada *keypad*

3. Port 2

- P2.0-P2.1 dihubungkan ke PS 2 pin K Clock dan K data
- P2.4-P2.5 dihubungkan ke DB 25 pin Strobe dan Busy
- P2.6-P2.7 dihubungkan ke LCD pin RS dan Enable.

4. Port 3

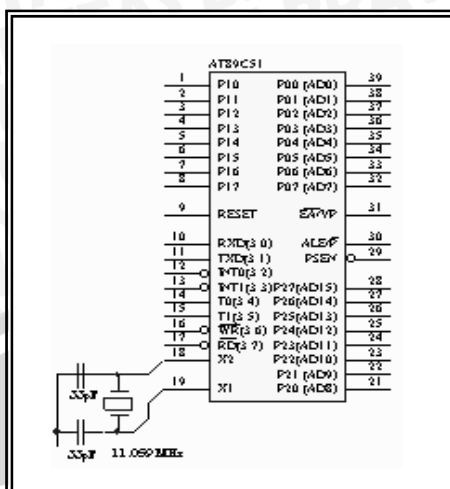
- P3.0-P3.1 dihubungkan ke MAX 232 secara serial di pin R1out dan T1in

5. Pin 18 dan pin 19 digunakan sebagai *input* dari rangkaian osilator kristal, yang terdiri atas osilator kristal dengan frekuensi 11,059 Mhz, kapasitor kristal C1 dan C2 yang masing-masing sebesar 33pF (sesuai *data sheet*, 30 pF±10 pF untuk kapasitor kristal). Rangkaian ini akan membangkitkan pulsa *clock* yang akan menjadi penggerak bagi seluruh operasi internal mikrokontroler.

6. Perencanaan Clock

Kecepatan proses yang dilakukan oleh mikrokontroler ditentukan oleh sumber *clock* yang mengendalikan mikrokontroler tersebut. Sistem yang dirancang ini menggunakan osilator *internal* yang telah tersedia dalam *chip* AT89C51. Untuk menentukan frekuensi osilatornya cukup dengan menghubungkan kristal dalam pin 19 (X₁) dan pin 18 (X₂) serta dua buah kapasitor ke *ground*.

Besarnya kapasitansi disesuaikan dengan spesifikasi dalam lembar data AT89C51 yaitu 30±10% pF. Besarnya kapasitansinya disesuaikan dengan spesifikasi dalam lembar data AT89C51 yaitu 33 pF. Kristal yang digunakan adalah 11,059MHz. Dalam Gambar 4.3 memperlihatkan rangkaian *clock* yang direncanakan.



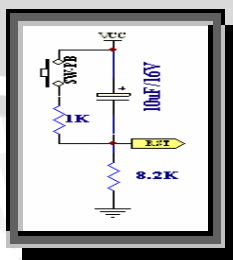
Gambar4.3 Rangkaian Clock

Baudrate dibangkitkan menggunakan Timer 1 dalam mode 2. Dengan kristal 11,059 MHz, dengan mengisi TH1 dengan FDH (253 *d*) dan SCON dengan 52*h*. sehingga diperoleh nilai baudrate, yaitu :

$$\begin{aligned}
 f_{\text{baud}} &= \frac{2^{\text{SMOD}}}{32d} \times \frac{\text{frekuensi osilator}}{12d \times [256d - (\text{TH1})]} \\
 &= \frac{2^1}{32d} \times \frac{11,059 \times 10^6}{12d \times [256d - 253d]} \\
 &= 19200 \text{ bps}
 \end{aligned}$$

7. Rangkaian Reset

Untuk mereset mikrokontroler AT89C51, maka pin RST diberi logika tinggi selama sekurangnya dua siklus mesin (24 periode osilator). Untuk membangkitkan sinyal reset kapasitor dihubungkan dengan VCC dan sebuah resistor yang dihubungkan ke ground. Rangkaian reset ditunjukkan dalam Gambar 4.4.



Gambar 4.4 Rangkaian Reset MCU

Karena kristal yang digunakan mempunyai frekuensi sebesar 11,059 MHz, maka satu perioda membutuhkan waktu sebesar :

$$T = \frac{1}{f_{XTAL}} = \frac{1}{11,059 \text{ MHz}} \text{ s} = 9,042 \times 10^{-8} \text{ s}$$

Dengan demikian waktu minimal logika tinggi yang dibutuhkan untuk mereset mikrokontroler adalah :

$$\begin{aligned} t_{\text{reset(min)}} &= T \times \text{periode yang dibutuhkan} \\ &= 9,042 \times 10^{-8} \times 24 = 2,170 \mu\text{s} \approx 2 \mu\text{s} \end{aligned}$$

Jadi mikrokontroler membutuhkan waktu minimal 2 μs untuk mereset. Waktu minimal inilah yang dijadikan pedoman untuk menentukan nilai R dan C. Dengan menentukan nilai R= 8,2 k Ω , dan C = 10 μF serta V_o adalah tegangan logika nominal yang diijinkan oleh pin RST.

$V_o = 0,7 \times V_{cc} = 0,7 \times 5 \text{ Volt} = 3,5 \text{ Volt}$, sehingga dari persamaan (2.1):

$$\begin{aligned} t &= R \times C \left[\ln \frac{5}{V_o} \right] \\ \Rightarrow t &= 8,2 \text{ k}\Omega \times 10 \mu\text{F} \left[\ln \frac{5}{3,5 \text{ Volt}} \right] \\ \Rightarrow t &= 29,25 \text{ ms} \end{aligned}$$

Jadi dengan nilai komponen R= 8,2 k Ω , dan C = 10 μF dapat memenuhi syarat minimal untuk waktu yang dibutuhkan oleh mikrokontroler.

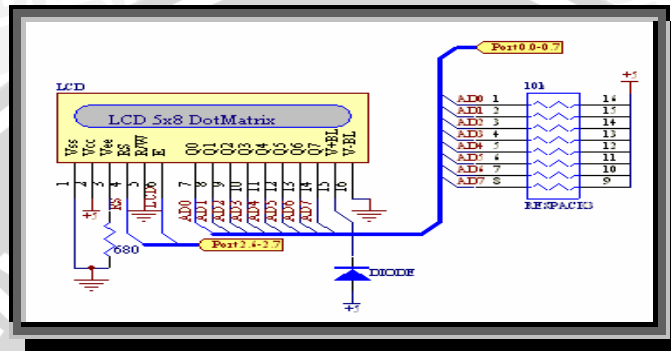
4.3.2 Antarmuka LCD (*Liquid Crystal Display*)

LCD yang digunakan dalam perancangan ini adalah tipe M1632, berfungsi untuk menampilkan informasi NIM yang berasal dari pembacaan barcode , password yang berasal dari penekanan tombol keypad serta sebagai tampilan informasi hasil konfirmasi PC apakah data yang dimasukkan benar atau salah

LCD dalam perancangan ini dikhususkan untuk melakukan proses tulis, sehingga hanya dapat digunakan untuk menulis perintah atau data, untuk itu pin R/W dihubungkan ke *ground* sehingga memberikan logika 0 yang mana sinyal ini mengisyaratkan bahwa LCD hanya melakukan proses tulis saja. Sedangkan sinyal pada pin RS akan mengisyaratkan apakah data yang diberikan pada LCD

merupakan sebuah instruksi atau benar-benar merupakan sebuah data. Apabila pin RS berlogika rendah, berarti data dianggap sebuah instruksi, dan apabila pin RS berlogika tinggi berarti data yang diberikan benar-benar adalah sebuah data.

Gambar antarmuka LCD (*Liquid Crystal Display*) ditunjukkan dalam Gambar 4.5. Dioda pada rangkaian LCD digunakan sebagai pengaman dari tegangan terbalik yang mungkin terjadi.



Gambar 4.5 Antarmuka LCD

Pin-pin yang digunakan adalah:

- D0-D7**

Pin ini dihubungkan pada port 0 mikrokontroler. Pin ini merupakan saluran data, berisi perintah dan data yang akan ditampilkan pada LCD (*Liquid Crystal Display*)
- Enable (E)**

Pin ini adalah sinyal operasi awal yang akan mengaktifkan data tulis, E dihubungkan pada mikrokontroler pin 2.7.
- R/ \bar{W}**

Pin ini adalah sinyal seleksi tulis (logika 0) dan baca (logika 1). Pin dihubungkan ke *ground*, karena dalam perancangan ini LCD (*Liquid Crystal Display*) dikhususkan untuk melakukan proses tulis saja.
- RS**

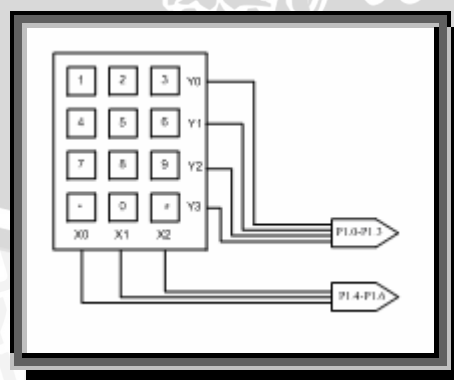
Pin ini adalah sinyal pemilih register internal. Logika 0 berarti instruksi register (tulis), dan logika 1 berarti data register (tulis dan baca). Pin RS ini dihubungkan pada mikrokontroler pin 2.6.

- V_{EE}
Pin ini untuk mengendalikan kecerahan LCD (*Liquid Crystal Display*) dengan memberikan nilai resistor yang dihubungkan padanya. Pada perancangan ini resistor yang dihubungkan pada VEE sebesar 680Ω .
- V_{CC}
Pin ini dihubungkan dengan catu daya sebesar +5 volt.
- V_{SS}
Pin ini merupakan terminal *ground*, dihubungkan ke *ground*.

4.3.3 Perencanaan Rangkaian Keypad

Keypad yang digunakan adalah adalah *keypad* matrik 3x4. Pin Y0-Y3 dihubungkan ke P1.0-P1.3 (berlogika 1) mikrokontroler, sedangkan X0-X2 dihubungkan ke P1.4-P1.6 mikrokontroler dengan X0 berlogika 0, X1 dan X2 berlogika 1. Berdasarkan program pada MCU (*Microcontroller Unit*) akan dilakukan proses *scanning* pada pin X0-X2, dengan cara melakukan penggeseran logika 0 secara bergantian pada pin X0, X1, dan X3, hingga dideteksi adanya penekanan pada salah satu tombol *keypad*.

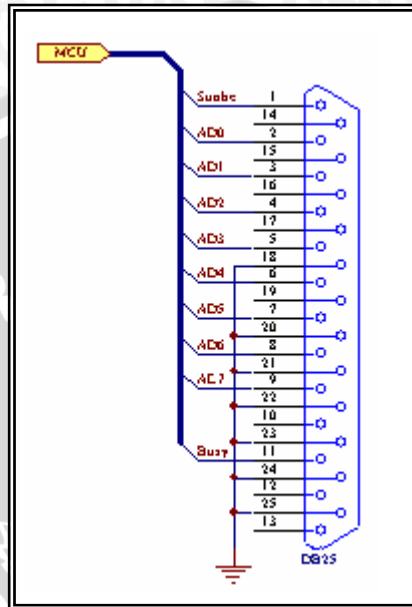
Apabila salah satu tombol ditekan, kolom dan baris yang berhubungan akan terhubung. Misalnya, tombol angka 1 ditekan. Dengan proses *scanning* akan terdeteksi adanya penekanan tombol pada X0, sehingga X0 berlogika 0. Baris yang terhubung dengan kolom X0 adalah Y0, sehingga Y0 menjadi berlogika 0. Kombinasi logika 0 pada X0 dan Y0 mengisyaratkan bahwa tombol angka 1 yang telah ditekan. Gambar rangkaian *keypad* ditunjukkan dalam Gambar 4.6



Gambar 4.6 Rangkaian Keypad

4.3.4 Antarmuka Printer

Port printer dapat digunakan untuk pengontrolan peralatan dalam hal ini mengirim dan menerima data. Gambar antarmuka printer ditunjukkan dalam Gambar 4.7



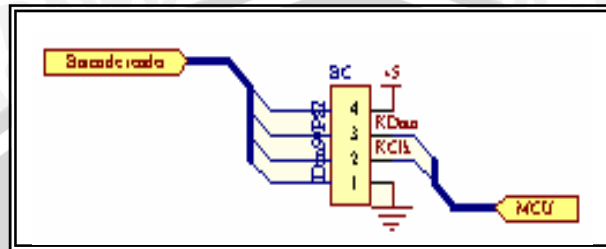
Gambar 4.7 Rangkaian antarmuka printer

Pin-pin yang digunakan adalah:

- **Pin 2-9**
Pin ini dihubungkan pada port 0 mikrokontroler. Pin ini merupakan saluran data, berisi perintah dan data yang akan dicetak *printer*
- **Pin 1**
Pin ini adalah sinyal Strobe yang digunakan untuk memberitahu printer bahwa karakter sudah dikirimkan. Sinyal ini aktif rendah. Dihubungkan pada mikrokontroler pin 2.5.
- **Pin 11**
Pin ini adalah sinyal *Busy*, sinyal ini berisi informasi kondisi printer yang tidak bisa menerima data. Kondisi ini disebabkan karena printer sedang menerima data, dalam proses mencetak dan dalam kondisi offline. Sinyal ini aktif *high*. Dihubungkan pada mikrokontroler pin 2.4.
- **Pin 18, pin 20-25**
Pin ini merupakan terminal *ground*, dihubungkan ke *ground*.

4.3.5 Antarmuka *Barcode Reader*

Dalam sistem ini barcode reader berfungsi sebagai pembaca kode baris kartu mahasiswa untuk akses NIM mahasiswa yang bersangkutan dengan antarmuka PS2 . Gambar antarmuka *Barcode reader* ditunjukkan dalam Gambar 4.8.



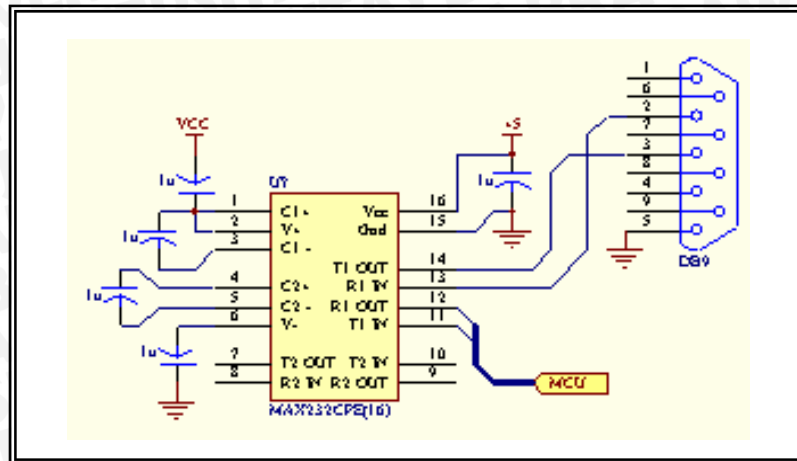
Gambar 4.8 Antarmuka *Barcode Reader*

Pin-pin yang digunakan adalah:

- **Pin 1**
Pin ini merupakan terminal *ground*, dihubungkan ke *ground*.
- **Pin 2**
Pin ini adalah K Data yang digunakan jalur komunikasi data antar Barcode Reader dengan MCU. Dihubungkan pada mikrokontroller pin 2.1.
- **Pin 3**
Pin ini adalah Clock . Dihubungkan pada mikrokontroller pin 2.0.

4.3.6 Rangkaian Komunikasi Serial

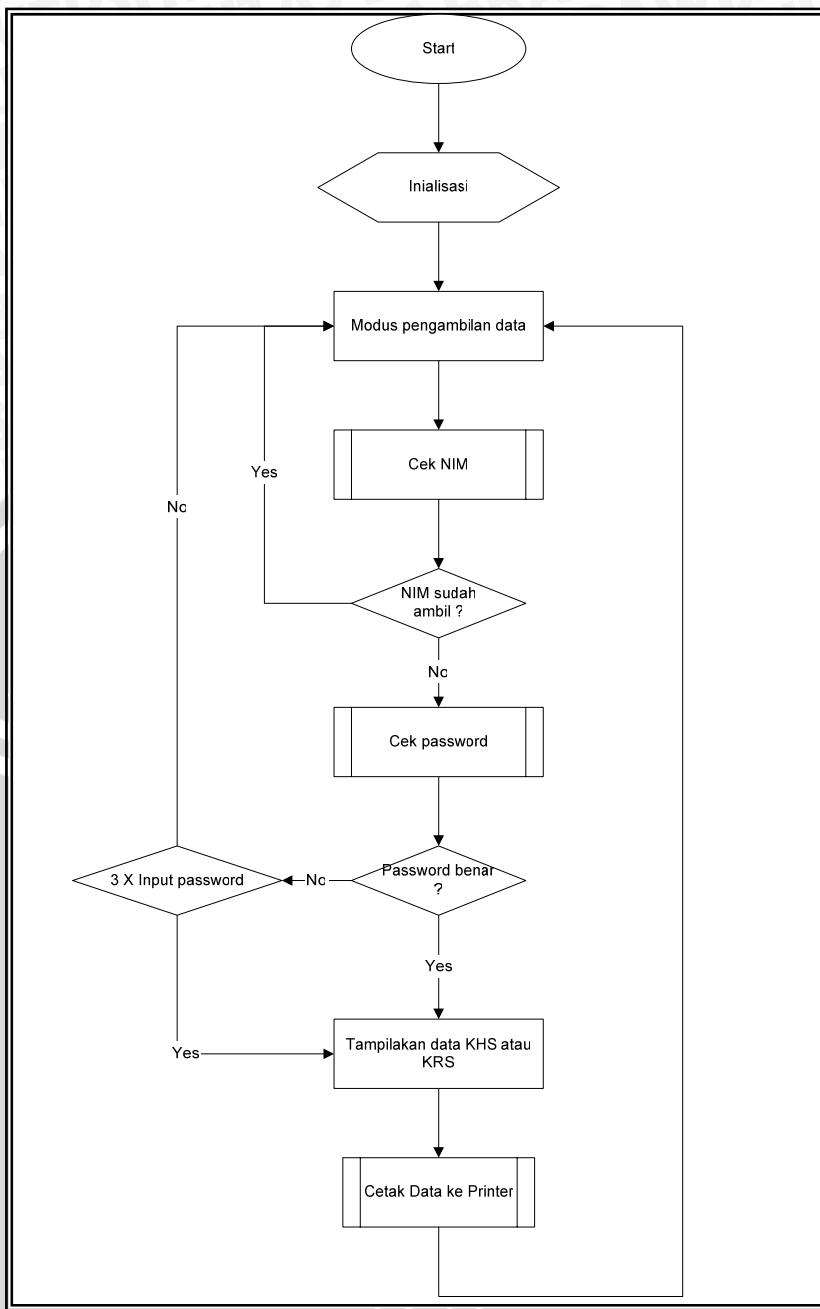
Karena port serial komputer tidak bekerja pada level tegangan TTL, melainkan level tegangan RS-232. untuk itu diperlukan komponen tambahan yang berfungsi untuk mengkonversi level tegangan TTL ke level tegangan RS-232, begitu juga sebaliknya. Komponen yang digunakan adalah MAX232CPE yang memerlukan beberapa komponen tambahan berupa lima buah kapasitor yang nilainya telah ditentukan datasheet, yaitu 1 μ F untuk seluruh kapasitor. MAX232CPE memiliki sepasang terminal masukan level tegangan TTL yang berkorespondensi dengan sepasang terminal keluaran level tegangan RS-232, juga sepasang terminal masukan level RS-232 yang berkorespondensi dengan sepasang terminal keluaran level tegangan TTL. Rangkain antarmuka komunikasi serial ke PC ditunjukkan dalam Gambar 4.9



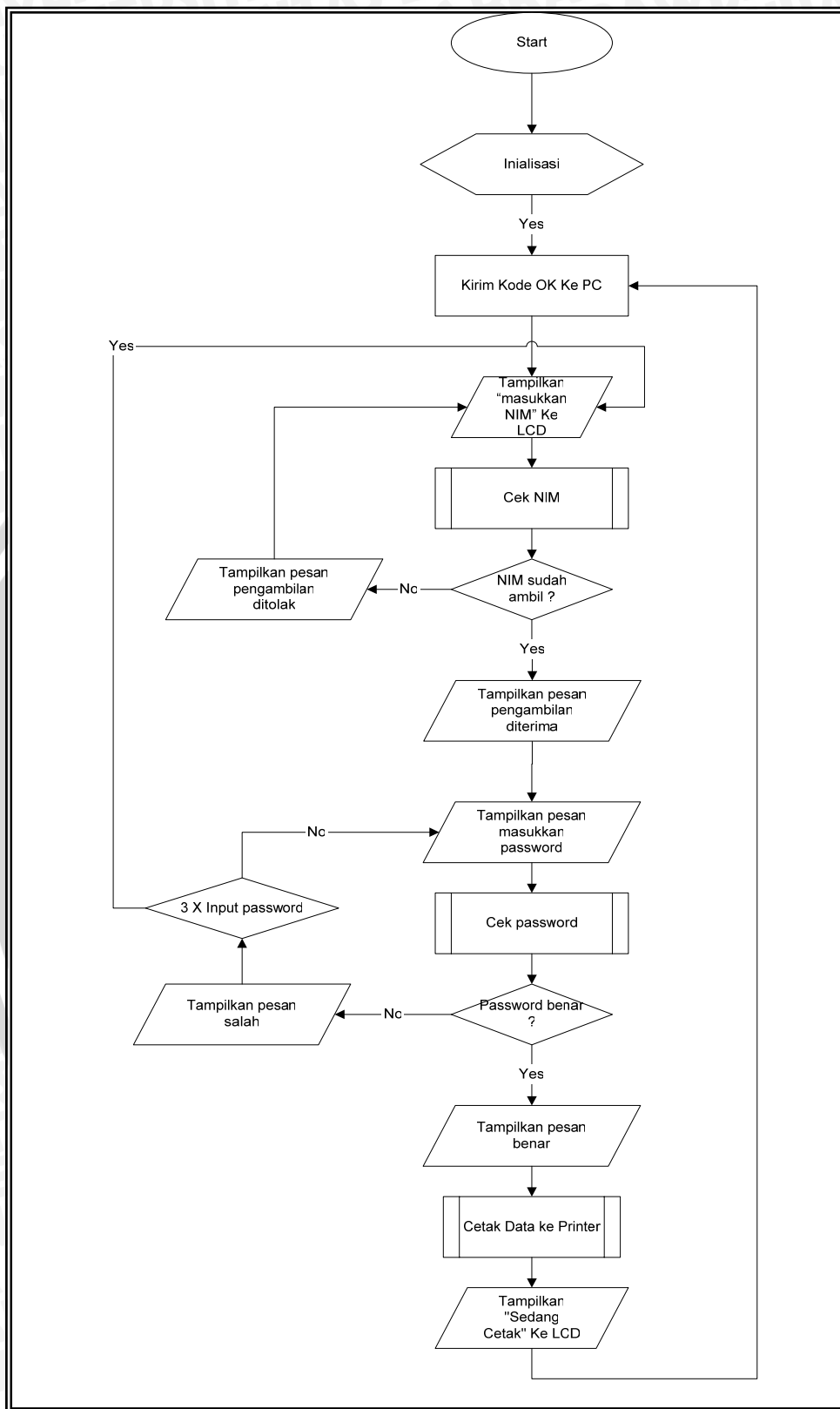
Gambar 4.9 Rangkaian Komunikasi Serial ke COM Komputer

4.4 Perancangan Perangkat Lunak

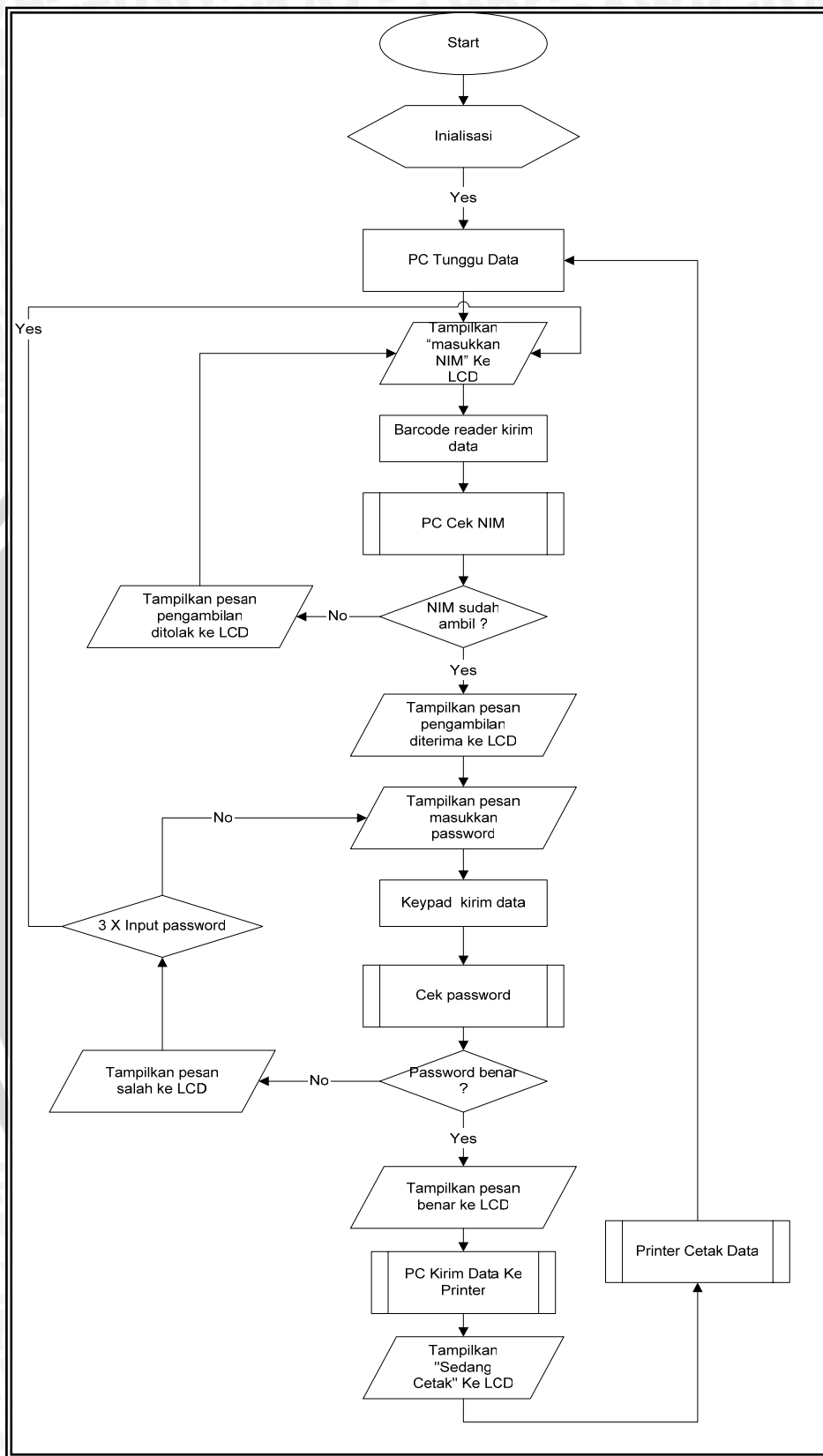
Perangkat lunak untuk mengendalikan sistem ini terdiri dari proses pengiriman dan pengambilan data dari unit masukan berupa barcode reader dan keypad, pengiriman dan pengambilan data pada PC berupa database dengan pemrograman Borland Delphi, pengiriman dan pengambilan data dari unit keluaran berupa printer dan program tampilan pada LCD. *Flowchart* pengiriman dan pengambilan data pada PC berupa database dengan pemrograman *Borland Delphi* ditunjukkan dalam Gambar 4.10, Gambar 4.11 merupakan *flowchart* pengendalian sistem yang dilakukan oleh Mikrokontroler sedangkan Gambar 4.12 menunjukkan *flowchart* utama atau diagram alur sistem pada alat otomatisasi pengambilan lembar hasil studi dan lembar hasil rencana studi



Gambar 4.10 Flowchart pengiriman dan pengambilan data pada PC berupa database



Gambar 4.11 flowchart Mikrokontroler



Gambar 4.12 flowchart Utama

BAB V

PENGUJIAN ALAT

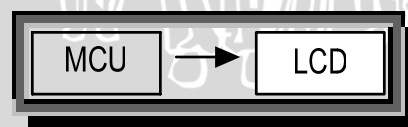
Dalam pengujian alat ini, dilakukan tahap-tahap pengujian terhadap perangkat keras (*hardware*) maupun perangkat lunak (*software*). Pengujian alat tersebut terdiri dari:

1. Pengujian mikrokontroler AT89C51 – LCD (*Liquid Crystal Display*)
2. Pengujian rangkaian *keypad*
3. Pengujian *barcode reader*
4. Pengujian komunikasi serial PC dengan MCU
5. Pengujian MCU - Printer
6. Pengujian keseluruhan sistem

Pengujian alat ini bertujuan untuk mengetahui unjuk kerja rangkaian tiap blok dan keseluruhan sistem. Apakah rangkaian yang telah dirancang dapat bekerja dengan baik sesuai dengan spesifikasi yang direncanakan.

5.1 Pengujian Mikrokontroler AT89C51 – LCD (*Liquid Crystal Display*)

Pengujian terhadap mikrokontroler dan LCD ini ditujukan untuk mengetahui apakah mikrokontroler dan LCD dapat bekerja dengan baik atau tidak. Pengujian ini dilakukan dengan mengisi program inisialisasi LCD ke dalam mikrokontroler. Selanjutnya mikrokontroler diberi masukan beberapa karakter dan ditampilkan melalui LCD. Bagan pengujian mikrokontroler dan LCD ditunjukkan dalam Gambar 5.1.



Gambar 5.1 Blok Diagram Pengujian Mikrokontroler dan LCD

Tampilan pada LCD dari hasil pengujian ditunjukkan dalam Gambar 5.2.



Gambar 5.2 Tampilan pada LCD dari Hasil Pengujian

Dari hasil pengujian, data yang dimasukkan ke dalam mikrokontroler dapat ditampilkan dalam LCD, hal ini berarti rangkaian mikrokontroler dan LCD dapat bekerja dengan baik.

5.2 Pengujian Keypad

Pengujian pada *keypad* ini bertujuan untuk mengetahui apakah data yang telah dikirim karena penekanan *keypad* telah sesuai atau tidak. Pengiriman data dilakukan melalui penekanan tombol *keypad* matrik baris dan kolom. Persambungan antara baris dan kolom yang terdeteksi dengan cara *scanning* akan menghasilkan data keluaran tertentu dalam bentuk biner. Pemrosesan data dari *keypad* dilakukan oleh mikrokontroler dan kemudian data tersebut oleh mikrokontroler dikirim untuk ditampilkan di LCD. Bagan pengujian *keypad* ditunjukkan dalam Gambar 5.3. Kombinasi bit untuk setiap penekanan tombol ditunjukkan dalam Tabel 5.1.



Gambar 5.3 Blok Diagram Pengujian Keypad

Tabel 5.1 Kombinasi Bit Keypad

Tombol	P1.7	X0 P1.6	X1 P1.5	X2 P1.4	Y3 P1.3	Y2 P1.2	Y1 P1.1	Y0 P1.0	HEXA
1	1	0	1	1	1	1	1	0	BE
2	1	1	0	1	1	1	1	0	DE
3	1	1	1	0	1	1	1	0	EE
4	1	0	1	1	1	1	0	1	BD
5	1	1	0	1	1	1	0	1	DD
6	1	1	1	0	1	1	0	1	ED
7	1	0	1	1	1	0	1	1	BB
8	1	1	0	1	1	0	1	1	DB
9	1	1	1	0	1	0	1	1	EB
*	1	0	1	1	0	1	1	1	D7
0	1	1	0	1	0	1	1	1	B7
#	1	1	1	0	0	1	1	1	E7

Untuk pengujian *keypad*, ditekan tombol angka 1, 2, 3, 4, 5 dan 6 pada *keypad*, yang selanjutnya ditampilkan di LCD. Hasil dari pengujian *keypad* ditunjukkan dalam Gambar 5.4.



Gambar 5.4 Tampilan LCD Hasil Pengujian *Keypad*

Angka yang ditampilkan pada LCD adalah sesuai dengan angka hasil penekanan tombol *keypad*. Hal ini menunjukkan bahwa rangkaian *keypad* dapat bekerja dengan baik.

5.3 Pengujian *Barcode Reader*

Pengujian pada *barcode reader* CCD 1000 series ini bertujuan untuk mengetahui apakah data dibaca oleh barcode reader telah dikirim sesuai atau tidak. Pengiriman data dilakukan melalui pembacaan barcode yang tertera pada kartu mahasiswa. Pemrosesan data dari *barcode reader* dilakukan oleh mikrokontroler dan kemudian data tersebut oleh mikrokontroler dikirim untuk ditampilkan pada program delphi di PC.

Cara penulisan barcode dalam pengujian ini menggunakan Barcode *Uniform Symbol Description-1* (USD-1) yang hanya dikodekan untuk angka, yang mewakili 10 digit (0 sampai 9) dan 2 kode untuk kode start dan kode stop. Tabel 5.2 memperlihatkan pengkodean angka ke dalam bentuk kode USD-1.

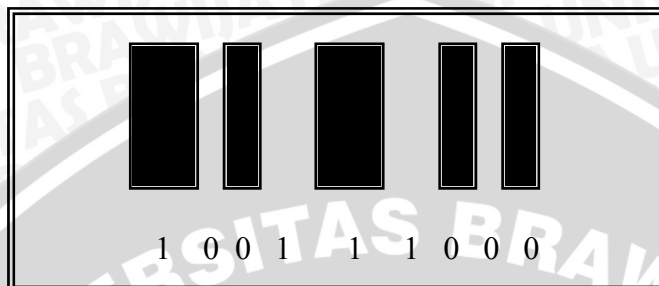
Tabel 5.2. Karakter Barcode USD-1

Data	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5
0	0	0	1	1	0
1	1	0	0	0	1
2	0	1	0	0	1
3	1	1	0	0	0
4	0	0	1	0	1
5	1	0	1	0	0
6	0	1	1	0	0
7	0	0	0	1	1
8	1	0	0	1	0
9	0	1	0	1	0

Sumber : Solomon, 1994:244



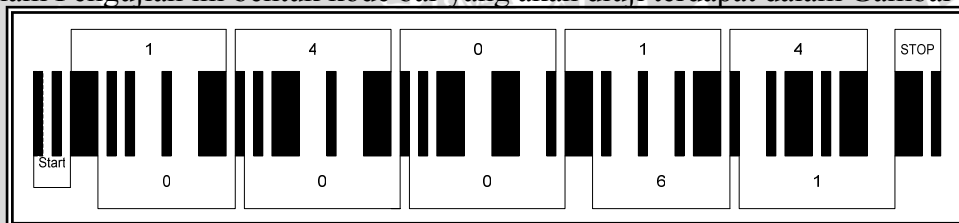
Karakter USD-1 disusun oleh 5 bit yang terdiri dari 2 bit tinggi (1) dan 3 bit rendah (0). Suatu elemen bentuk tebal baik bar maupun spasi akan selalu diartikan sebagai logika tinggi (bit tinggi), sedangkan bentuk tipis baik bar atau spasi akan selalu diartikan logika rendah (bit rendah). Gambar 5.5 memperlihatkan cara pembacaan kode USD-1.



Gambar 5.5 Pembacaan Kode USD-1

Sumber: Solomon, 1994:245

Dalam Pengujian ini bentuk kode bar yang akan diuji terdapat dalam Gambar 5.6



Gambar 5.6 Bentuk barcode yang diuji

Gambar 5.5 memperlihatkan bentuk kode bar USD-1 ini yang terdiri atas 3 bagian yaitu bagian kode start, data dan kode stop. Kode start digunakan untuk mengetahui awal pembacaan. Data merupakan informasi yang akan diterjemahkan ke dalam karakter angka yang bersesuaian. Kode stop untuk mengetahui akhir pembacaan kode.

Untuk angka 1,4,0,1 dan 4(bagian atas) pengkodean diambil dari bentuk bar gelap kode tersebut, yang masing – masing angka terdiri dari 5 elemen bar, sedangkan angka 0, 0,0,6 dan 9 (bagian bawah) pengkodeannya diambil bentuk spasi terang yang masing – masing angka terdiri dari 5 elemen spasi. Seperti dalam Tabel 5.2 dapat dilihat bahwa angka 1 dikodekan 10001 atau dalam urutan tebal, tipis, tipis, tipis, tebal. Hal ini sesuai dengan urutan 5 bar pertama dalam Gambar 5.6, demikian juga dengan angka bagian atas merupakan penerjemahan dari urutan bar berikutnya. Untuk angka 0 dikodekan 00110, seperti terlihat

dalam urutan 5 elemen spasi pertama kode tersebut berupa tipis, tipis, tebal, tebal, tipis. Demikian juga untuk pembacaan angka bagian bawah yang diambil dari urutan spasi kode yang tercetak. Sehingga secara keseluruhan kode tersebut diterjemahkan ke dalam urutan bilangan 1040001641. Hasil Pengujian beberapa *barcode* ditunjukkan dalam Tabel 5.3

Tabel 5.3 Hasil Pengujian *BarcodeReader*

<i>Barcode</i>	Data <i>barcode</i>	Hasil <i>Barcode</i> yang terbaca
	1040001640	MASUKKAN KARTU 1040001640
	1040001641	MASUKKAN KARTU 1040001641
	1040001642	MASUKKAN KARTU 1040001642
	1040001643	MASUKKAN KARTU 1040001643
	1040001644	MASUKKAN KARTU 1040001644
	1040001645	MASUKKAN KARTU 1040001645
	1040001646	MASUKKAN KARTU 1040001646
	1040001647	MASUKKAN KARTU 1040001647
	1040001648	MASUKKAN KARTU 1040001648
	1040001649	MASUKKAN KARTU 1040001649

Dalam pengujian ini *barcode reader* tipe CCD SD 1000 series batas kemampuan baca data *barcode*. Dalam Tabel 5.4 dapat kita lihat ukuran data *barcode* yang mampu dibaca oleh *barcode reader*.

Tabel 5.4. Tabel pengujian ukuran Data barcode yang terbaca oleh barcode reader

Data Barcode	Ukuran (mm ²)	Terbaca
	50 x 2,8	
	40 x 3,8	
	43 x 3,3	1040001641
	85 x 72	1040001641
	86 x 10	1040001641
	87 x 16	1040001641
	88 x 16	1040001641
	89 x 16	1040001641
	91 x 16	1040001641
	92 x 16	

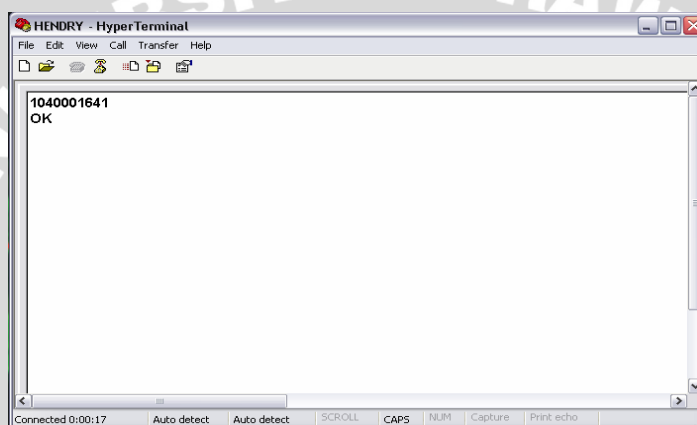
5.4 Pengujian komunikasi serial PC dengan MCU

Pengujian ini bertujuan untuk mengetahui apakah program yang telah ditulis ke MCU AT89C51 berjalan dengan benar. Block diagram pengujian ini dapat kita lihat dalam Gambar 5.7



Gambar 5.7. Blok Diagram Komunikasi PC dengan MCU AT89C51

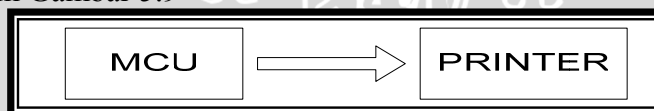
Pengujian ini menggunakan program *hyper terminal* pada windows, dan hasilnya dapat kita lihat dalam Gambar 5.8



Gambar 5.8. Program Hyper Terminal Hasil Pengujian PC dengan MCU AT89C51

5.5 Pengujian MCU - Printer

Pengujian ini bertujuan untuk mengetahui data dari PC yang dituliskan ke MCU dapat dicetak dengan baik oleh printer. Block diagram pengujian ini dapat kita lihat dalam Gambar 5.9



Gambar 5.9 Block diagram pengujian MCU dengan Printer

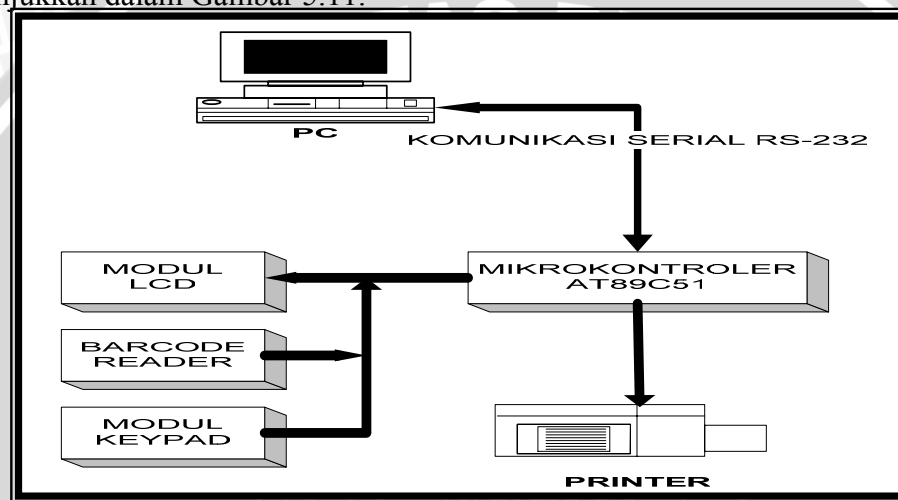
Pada saat printer mencetak pada LCD akan menampilkan tulisan "Harap tunggu sedang dicetak". Hasil pengujian dapat kita lihat dalam Gambar 5.10



Gambar 5.10 Hasil pengujian MCU dengan Printer

5.6 Pengujian Keseluruhan Sistem

Pengujian ini bertujuan untuk menguji kerja dari sistem yang telah dirancang telah bekerja sesuai dengan yang diharapkan pada perancangan. Langkah pertama adalah merangkai alat hasil perancangan. Kemudian antara alat hasil perancangan dan PC dihubungkan dengan menggunakan sistem komunikasi serial. Setiap anggota diberi kartu mahasiswa yang menggunakan barcode yang berisi data NIM serta password untuk proteksi keamanan dalam mengakses data. Pengujian dilakukan dengan merangkai keseluruhan blok rangkaian seperti ditunjukkan dalam Gambar 5.11.



Gambar 5.11. Block diagram pengujian keseluruhan sistem

Langkah – langkah pengujian :

Menyiapkan Sistem

1. Menghubungkan terminal pada konektor serial PC melalui kabel serial RS 232
2. Menghubungkan printer pada konektor printer terminal lalu hidupkan printer.
3. Siapkan kertas khusus pada printer untuk pencetakan modul yang diinginkan.
4. Hidupkan PC dan jalankan program aplikasi “Sistem Otomatisasi Pengambilan Lembar Hasil dan Rencana Studi

Menyiapkan database

Masukan file database sumber dari pusat data ke PC dan konversi. Mengaktifkan Modus Pengambilan. Setelah PC melakukan inisialisasi, terminal menunggu pemasukan kode pada Barcode. Hal ini dapat kita lihat dalam Gambar 5.12



Gambar 5.12. Database dalam modus pengambilan data

Terminal yang telah siap menerima masukan dapat dipakai mahasiswa untuk mengambil Lembar Hasil atau Rencana Studi secara terpisah tanpa melalui loket-loket yang disediakan.

Cara pemakaian terminal tersebut adalah sebagai berikut:

Mahasiswa yang ingin mengambil Lembar Hasil dan Rencana Studi memasukan NIM-nya dengan bantuan Kartu mahasiswa yang diberi barcode yang discan melalui pembaca barcode pada terminal. Kemudian melihat pada LCD untuk keterangan status NIM nya. Jika NIM benar dan ia belum pernah melakukan pengambilan sebelumnya maka ia diminta untuk memasukan kode akses melalui keypad .Sedangkan jika NIM tidak benar atau ia telah melakukan pengambilan sebelumnya maka ia akan ditolak dengan keterangan pada LCD bahwa pengambilan ditolak. Jika pada saat pemasukan kode akses terjadi kesalahan maka ia dapat mengulanginya dengan menekan tombol bintang (*). Pengulangan masukan kode akses maksimal adalah 3 kali, setelah itu sistem akan meminta untuk memasukan NIM kembali. Setelah pemasukan kode akses diterima, mahasiswa dapat mengambil Lembar Hasil atau Rencana Studi langsung dari alat pencetak (printer) yang telah disediakan. Gambar 5.13 merupakan tampilan program setelah memasukkan NIM dan *Password* .Gambar 5.14. merupakan tampilan pada saat pembacaan NIM pada *Barcode*. Gambar 5.15. merupakan tampilan KHS yang siap dicetak



Gambar 5.13. Tampilan pada saat pembacaan NIM pada Barcode



Gambar 5.14. Tampilan program setelah memasukkan NIM dan Password

KHS Online

File Setting

NIM: 1040001641
 Nama: HENDRY
 Semester: 1

KODEMK	Matakuliah	Sks	Nilai	Nk	Ket
TKE132	MATEK	3	B	9	
TKE121	PLC	2	B	6	
TKE123	KALKULUS 1	4	B+	14	
TKE141	DTE	3	C	6	
TKE131	FISIKA 1	4	B	12	
TKE151	KIMIA	2	A	8	
TKE125	FALSAFAH	2	B	6	

Jumlah Bobot: 61
 Jumlah SKS: 20

IP: 3.05
 Beban Studi: 7

Gambar 5.15. Tampilan KHS yang siap dicetak

BAB VI PENUTUP

6.1. Kesimpulan

Berdasarkan perancangan dan pengujian alat, maka dapat ditarik kesimpulan sebagai berikut :

1. *Barcode reader* yang digunakan untuk mengakses NIM dan *keypad* untuk memasukkan *password* adalah *input* yang dibutuhkan untuk mengakses data pada PC
2. PC digunakan sebagai *Database* menggunakan *program Delphi 7* yang digunakan sebagai penyimpan data mahasiswa berupa KHS dan KRS
3. *Prntr* digunakan sebagai *output* data berupa KHS atau KRS yang diakses dari PC untuk dicetak
4. Waktu yang diperlukan sistem setelah melakukan penekanan *keypad* untuk akses *password* sampai selesai mencetak dengan kisaran waktu 3 menit 37 detik sampai dengan 4 menit 47 detik
5. Pembacaan *barcode reader* tipe CCD 1000 series dapat membaca data *barcode* dengan data 10 angka yaitu 1040001641 dengan ketentuan yaitu lebar bar dan spasi antara tersempit minimal 1 mm dengan panjang data *barcode* maksimal 9,1 mm, dan untuk tinggi data *barcode* minimal 3,3 mm. Perbandingan bar dan spasi antar yang tebal dan yang tipis adalah 3 : 1

6.2 Saran

Untuk memperbaiki kekurangan dan untuk pengembangan lebih lanjut maka dapat dilakukan :

1. Untuk lebih efektif dan lebih handal *barcode reader* yang digunakan menggunakan tipe Laser.
2. Untuk program database dapat lebih variatif dengan tampilan-tampilan yang lebih baik

DAFTAR PUSTAKA

- Anonymous. 1995. *Barcode Reader*. Wikipedia Corporation. Akses dari: http://en.wikipedia.org/wiki/Barcode_reader Tanggal Akses: 15 Desember 2005.
- Anonymous. 1996. *Low Power, Slew-Rate-Limited RS-485/RS422 Transceivers*. Maxim corporation. Akses dari: www.maxim.com. Tanggal Akses: 10 Desember 2005.
- Anonymous. 1997. *8-Bit Microcontroller with 4 kBytes Flash*. Atmel corporation. Akses dari: www.atmel.com. Tanggal Akses: 28 September 2005.
- Anonymous. 1997. *Liquid Crystal Display Modules*. DataSheet corporation. Akses dari: www.DatasheetCatalog.com Tanggal akses: Oktober 2005.
- Anonymous. 1998. *Akses Keypad 4x3*. Delta-electronic corporation. Akses dari: <http://www.delta-electronic.com/>. Tanggal akses: 3 Desember 2005.
- Anonymous. 1987. : *M1632 User Manual*. . DataSheet corporation. Akses dari: www.DatasheetCatalog.com. Tanggal akses: oktober 2004.
- Anonymous. 1983. *Oscillator for microcontroller*. Intell corporation. Akses dari: www.intel.com. Tanggal Akses: 10 September 2006.
- Anonymous. 2000. *Interfacing the Standard Parallel Port*. National Semiconductor. Akses dari: <http://www.beyondlogic.org/spp/parallel.htm> Tanggal Akses: 10 Desember 2005.
- M1632, 1987. *User Manual Liquid Cristal display Moddule M1632*. Japan: Seiko Instruments Inc..
- Nalwan, Paulus Andi. 2002. *Panduan Praktis Teknik Antarmuka dan Pemrograman Mikrokontroler AT89C51*. Jakarta: PT. Elex Media Komputindo.
- Putra, Eko Agfianto. 2002. *Belajar Mikrokontroler AT89C51/52/55 (Teori dan Aplikasi)*. Yogyakarta: Gava Media.

LAMPIRAN L-1.1

RANGKAIAN LENGKAP



LAMPIRAN L-2.1

FOTO ALAT



LAMPIRAN L-3.1

LISTING PROGRAM



LAMPIRAN L-3.1.1
LISTING PROGRAM
MIKROKONTROLER



LAMPIRAN L-3.1.2

LISTING PROGRAM DELPHI



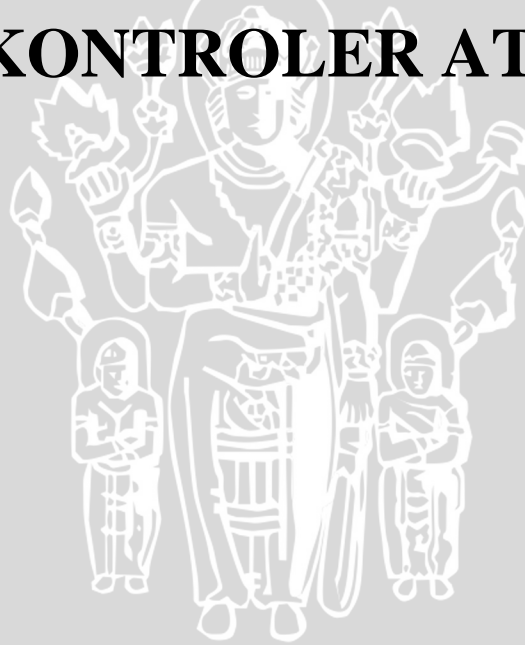
LAMPIRAN L-4.1

LEMBAR *DATASHEETS*



LAMPIRAN L-4.1.1

LEMBAR DATA MIKROKONTROLER AT89C51



LAMPIRAN L-4.1.2

LEMBAR DATA RS-232

