BAB III

PERANCANGAN DAN PEMBUATAN ALAT

3.1. Diagram Blok

Diagram blok sistem pembaca kartu digital berbasis mikrokontroler AT89S8252 ini terdiri dari beberapa bagian yakni masukan berupa *keypad* matrik 3 x 4 dan rangkaian *optocoupler* sebagai pembaca kartu digital, pengendalinya menggunakan mikrokontroler AT89S8252 dan keluarannya menggunakan penampilnya LCD (*Liquid crystal display*) 16 x 2 karakter serta miniatur pintu geser elektronik. Diagram blok pembaca kartu digital dapat dilihat pada Gambar 3.1.

Gambar 3.1. Diagram blok pembaca kartu digital berbasis mikrokontroler.

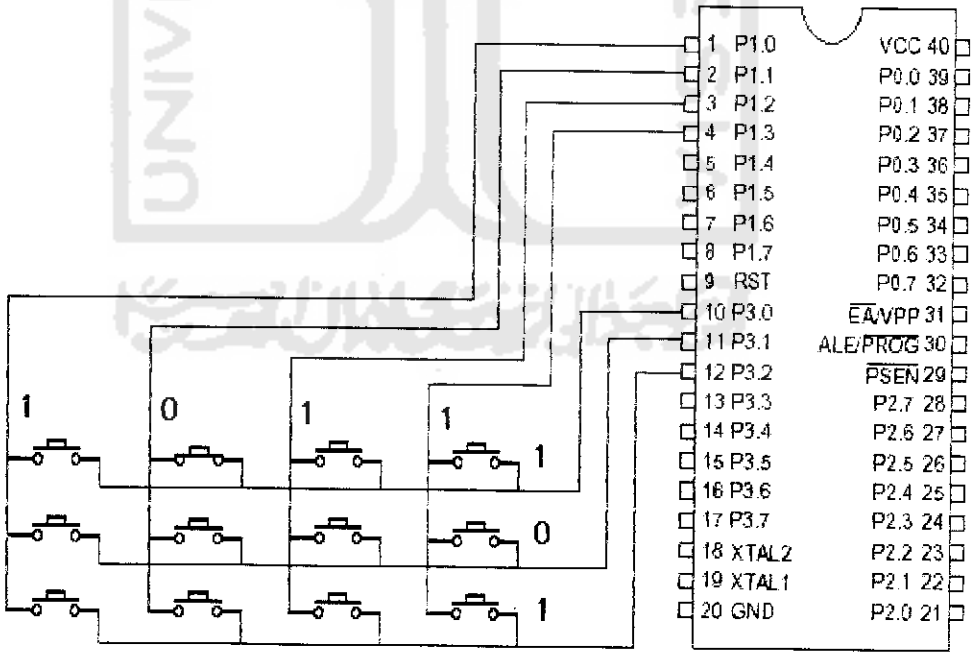
3.2. Perancangan Perangkat Keras

Perancangan perangkat keras sistem pembaca kartu digital ini memiliki tiga bagian utama, yaitu: masukan, pengontrol dan keluaran. Piranti masukannya berupa *keypad* dan rangkaian *optocoupler* (pembaca kartu), piranti kontrolnya adalah mikrokontroler AT89S8252, sedangkan piranti keluarannya adalah penampil LCD dan rangkaian penggerak pintu geser elektronik.

3.2.1. Rangkaian *Keypad*

Keypad merupakan rangkaian saklar yang berfungsi sebagai masukan, dalam alat ini digunakan *keypad* matrik 3 x 4, yang mana 3 bit barisnya dihubungkan dengan P3.0 – P3.2, sedangkan 4 bit kolomnya dihubungkan dengan P1.0 – P1.3.

P3.0-P3.2 berfungsi sebagai keluaran *keypad* mikrokontroler dan P1.0 - P1.3 berfungsi sebagai masukan *keypad* mikrokontroler. Cara kerja dari *keypad* matrik ini adalah dengan sistem *scaning* tombol satu persatu. Mikrokontroler akan memberikan kondisi '0' pada setiap port keluaran secara bergantian sehingga apabila pada saat saat yang sama ada penekanan tombol maka pada port masukan akan memiliki kondisi yang sama yaitu '0' seperti yang ditunjukkan pada Gambar 3.2.



Gambar 3.2. Contoh aplikasi *keypad* matrik 3 x 4

Apabila port keluaran dikondisikan dengan bit 101, lalu pada tombol kolom kedua tersebut ditekan, maka masukan yang dihubungkan oleh tombol tersebut menjadi '0' dan kemudian oleh mikrokontroler diterjemahkan menjadi angka tertentu, seperti yang ditunjukkan pada Tabel 3.1.

Tabel 3.1. Kombinasi baris dan kolom *keypad* matrik 3 x 4.

No	Data tombol	Kombinasi kolom $P_{1.3} - P_{1.0}$	Kombinasi baris $P_{3.2} - P_{3.0}$
1	**1**	1110	110
2	**2**	1101	
3	**3**	1011	
4	**4**	0111	
5	**5**	1110	101
6	**6**	1101	
7	**7**	1011	
8	**8**	0111	
9	**9**	1110	011
10	**0**	1101	
11	**CLEAR**	1011	
12	**OK**	0111	

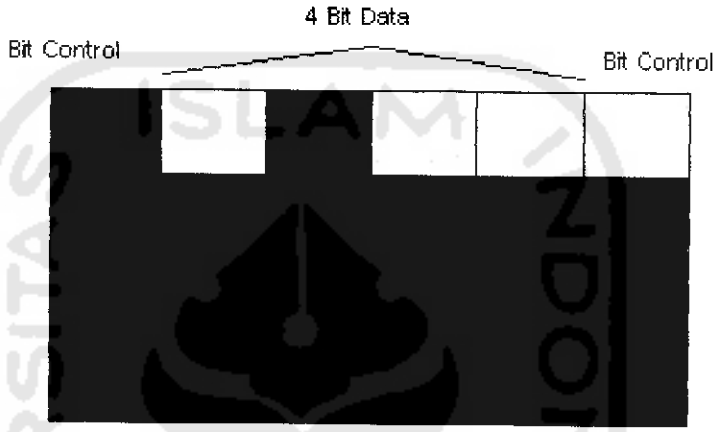
Bantuk tampilan *keypad* matrik 3 x 4 pada alat ini dapat dilihat pada Gambar 3.3.

1	**2**	**3**	**4**
5	**6**	**7**	**8**
9	**0**	**CLEAR**	**OK**

Gambar 3.3 Tampilan *keypad* matrik 3 x 4

3.2.2. Kartu Digital dan Rangkaian *Optocoupler*

Kartu digital dibuat dari lempeng plastik yang diisi dengan kode blok gelap dan terang seperti yang ditunjukkan pada Gambar 3.4



Gambar 3.4. Kartu digital

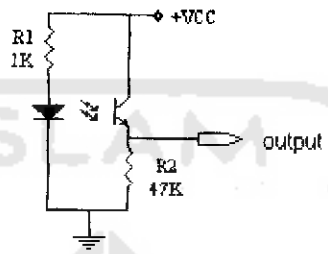
Kondisi gelap-terang kartu tersebut terdiri dari 2 bit kontrol yang digunakan untuk mengecek apakah kartu yang dimasukkan dalam posisi yang benar atau salah dan 4 bit yang lain merupakan data dari kartu tersebut. Kondisi gelap-terang tersebut dibaca oleh *optocoupler* sebagai kode biner '0' dan '1', yang mana kondisi gelap sama dengan '0' dan kondisi terang sama dengan '1'. Kode-kode tersebut menjadi masukan *Port* 3 mikrokontroler, dimana pengecekan bit kontrol dilakukan pada P3.7, dan 4 bit data kartu pada P3.3 - P3.6. *Port-port* yang lain yaitu P3.0-P3.2 terhubung pada masukan *keypad* yang diberi kondisi '1' agar *keypad* tidak aktif selama pengecekan kartu. Kombinasi data kartu pada *port* 3 dapat dilihat pada Tabel 3.2.

Tabel 3.2. Data kartu yang terbaca pada *port* 3.

No kartu	Bit kontrol	Data kartu				Masukan *keypad*			*Port* 3 (hexa)
	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	
1	0	0	1	1	1	1	1	1	3F
2	0	1	0	1	1	1	1	1	5F
3	0	0	0	1	1	1	1	1	1F
4	0	1	1	0	1	1	1	1	6F
5	0	0	1	0	1	1	1	1	2F
6	0	1	0	0	1	1	1	1	4F
7	0	0	0	0	1	1	1	1	0F
8	0	1	1	1	0	1	1	1	77
9	0	0	1	1	0	1	1	1	37
10	0	1	0	1	0	1	1	1	57
11	0	0	0	1	0	1	1	1	17
12	0	1	1	0	0	1	1	1	67
13	0	0	1	0	0	1	1	1	27
14	0	1	0	0	0	1	1	1	47

Port 3.7 kondisinya harus '0' yaitu data pertama kartu yang merupakan bit kontrol dan jika kondisinya '1' (tidak nol) tidak akan dilakukan pengecekan pada data kartu selanjutnya. Kode 0000 dan 1111 tidak dimasukkan dalam data kartu untuk menghindari pemasukan kartu yang tidak sesuai, sehingga kombinasi data kartu yang dimungkinkan sebanyak 14 buah.

Data pada kartu digital hanya bisa dibaca jika *optocoupler* pertama membaca '0'. Hal ini digunakan untuk mengetahui apakah kartu yang dimasukkan dalam posisi yang benar atau tidak.

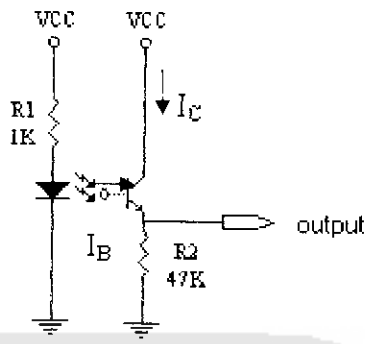


Gambar 3.5. Rangkaian *optocoupler* pembaca kartu digital

Pembacaan kartu digital oleh *optocoupler* ditunjukkan pada Gambar 3.5. Dalam *datasheet optocoupler* H21A3 (dalam lampiran) arus maju maksimum dioda I_F(max) = 60 mA dan tegangan maju V_F = 1,7 V. Dalam perancangan digunakan resistor dioda (R_d) sebesar 1K dan Vcc 5V, sehingga didapat nilai I_F sebesar:

$$I_F = \frac{V_{CC} - V_F}{R_d} = \frac{3,3}{1.10^3} = 3,3mA$$

Dengan demikian I_F perancangan tidak melebihi nilai dari I_F maksimumnya. Saat *phototransistor* menerima cahaya dari *infrared*, kondisi transistor ON karena basisnya mendapatkan bias dari cahaya *infrared*. Arus basis I_B > 0 yang menyebabkan tegangan pada emiternya (V_E) besar mendekati tegangan V_{CC}. Untuk memaksimalkan tegangan keluaran *optocoupler* maka *phototransistor* dibuat dalam keadaan jenuh. Tegangan $V_{CE(sat)}$ untuk I_C = 1,8 mA adalah 0,4 Volt sehingga tegangan keluaran maksimalnya adalah



Gambar 3.6. Rangkaian *optocoupler* saat ON

Vout $= V_{CC} - V_{CE(sat)}$

$= 5V - 0,4\ V$

$= 4,6$ Volt

Karena $V_{CE(sat)}$ sangat kecil sehingga arus yang mengalir pada emiternya sangat besar hampir sama dengan arus kolektornya ($I_C \approx I_E$), sehingga resistor R_E minimal yang dapat digunakan adalah:

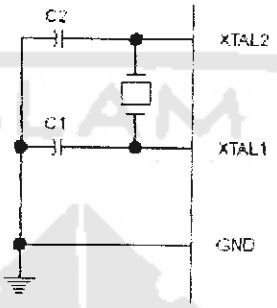
$$R_{E(min)} = \frac{V_{out}}{I_E} = \frac{4,6}{1,8mA} = 2555,5\Omega = 2,5\ K\Omega$$

Dalam perancangan digunakan resistor sebesar 4K7 melebihi nilai $R_{E(min)}$ dan cukup untuk membuat *phototransistor* jenuh. Jika cahaya *infrared* dihalangi, kondisi transistor OFF karena basisnya tidak mendapatkan cahaya *infrared*. Arus basis $I_B = 0$ menyebabkan tegangan pada colektor Vc maksimum mendekati tegangan V_{CC} dan tegangan V_E nya sangat kecil mendekati nol sehingga tidak melewatkan tegangan keluaran Vout.

3.2.3. Mikrokontroler AT89S8252

3.2.3.1. Osilator

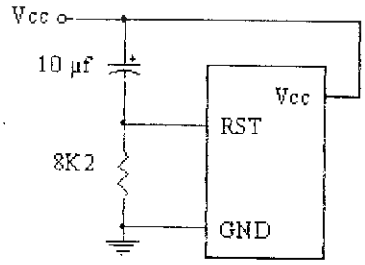
Pada rangkaian osilator ini digunakan kristal 12 MHz sepertti pada Gambar 3.7 berikut ini:



Gambar 3.7. Rangkaian osilator

3.2.3.2. Reset

Mikrokontroler direset pada transisi tegangan rendah ke tegangan tinggi dan mengeksekusi program pada saat reset (RST) dalam keadaan logika rendah. Oleh karena itu pada pin RST dipasang kapasitor yang terhubung ke VCC dan resistor ke ground yang akan menjaga RST bernilai 1 pada saat pengisian kapasitor dan akan kembali 0 sesaat kemudian, dengan demikian mikrokontroler akan direset setiap kali diberi catu daya.



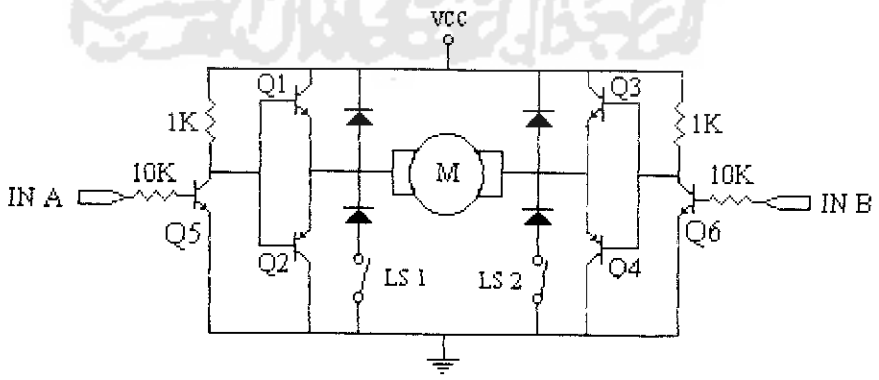
Gambar 3.8. Rangkaian reset pada mikrokontroler

3.2.4. Rangkaian Penggerak Pintu Geser Elektronik

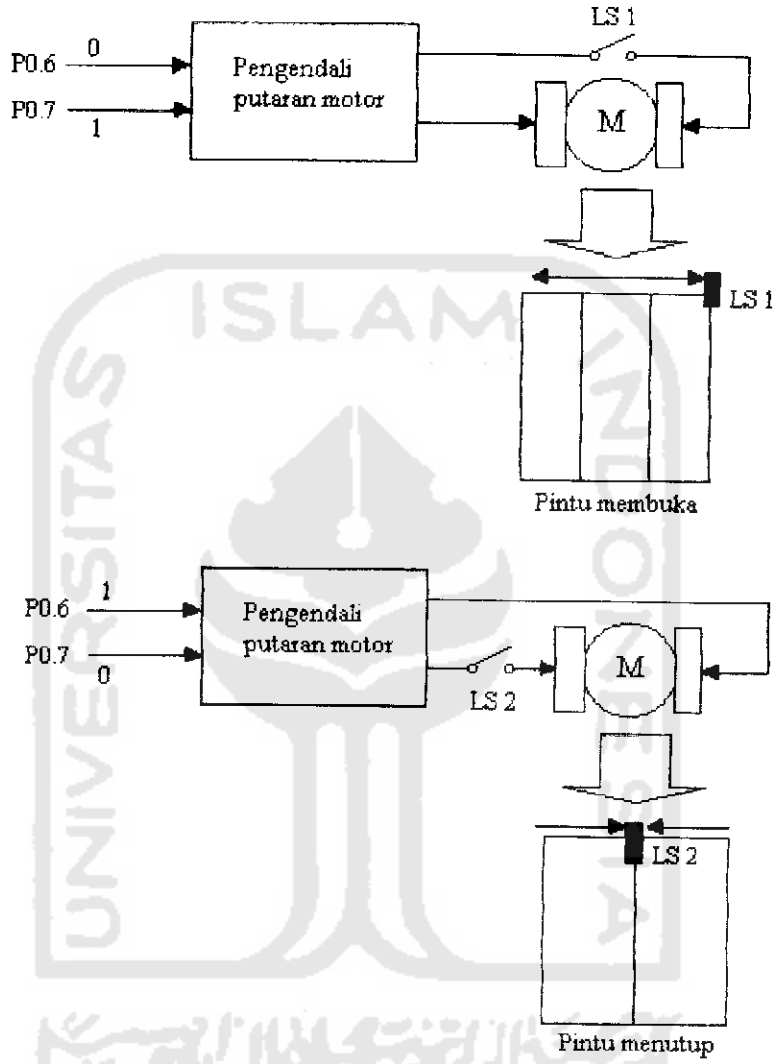
Rangkaian pintu geser elektronik terdiri dari sebuah motor DC yang digerakkan oleh dua buah transistor PNP dan transistor NPN. Prinsisip kerja rangkaian ini adalah membuat perbedaan polaritas antara kedua masukannya.

Saat input A diberi tegangan positif dan input B diberi tegangan nol maka akan mengaktifkan transisitor NPN Q1 dan transistor PNP Q4 sehingga motor berputar *clockwise* (CW). Sedangkan saat input A diberi tegangan nol dan input B diberi tegangan positif maka transistor yang aktif adalah transistor Q2 dan transistor Q3 sehingga motor berputar *counter clockwise* (CCW). Jika kedua input tersebut diberi tegangan yang sama motor akan diam (tidak berputar) karena tidak ada beda potensial pada motor tersebut. Saat pintu terbuka penuh akan menghubungkan saklar LS1 (*limit switch*) dan saat menutup penuh akan mengaktifkan LS2.

Gambar rangkaian penggerak motor dapat dilihat pada Gambar 3.9, sedangkan pemasangan saklar LS1 dan LS2 pada pintu geser elektronik dapat dilihat pada Gambar 3.10.

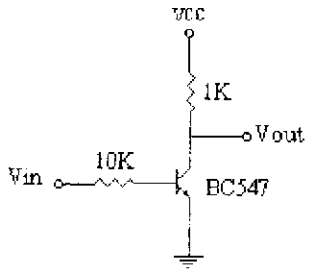


Gambar 3.9. Rangkaian penggerak motor



Gambar 3.10. Pemasangan saklar LS1 dan LS2 pada pintu geser elektronik

Pemasangan dioda pada kaki emiter kolektor transistor pada rangkaian penggerak motor berfungsi untuk meneruskan arus induksi dari motor agar tidak membebani transistor. Untuk memperkuat torsi motor maka pada kedua input A dan B diberi rangkaian penguat arus seperti pada Gambar 3.11.



Gambar 3.11. Penguat arus

Rangkaian penguat arus berfungsi untuk meningkatkan arus keluaran seperti berikut ini:

$$I_{C(maks)} = \frac{V_{CC}}{R_C} = \frac{12}{10^3} = 12\ mA$$

Sehingga untuk membuat transistor jenuh dengan β = 125 (hasil pengukuran) besarnya $I_{B(Min)}$ adalah:

$$I_{B(Min)} = \frac{I_C}{\beta} = \frac{12mA}{125} = 96\ \mu A$$

Dan jika digunakan resistor basis sebesar 10 K untuk V_B = 5 V maka I_B nya adalah:

$$I_B = \frac{V_B - V_E}{R_B} = \frac{5-0,7}{10^4}$$

$$= \frac{4,3}{10^4} = 4,3.10^{-4}\ A$$

$$= 0,43\ mA$$

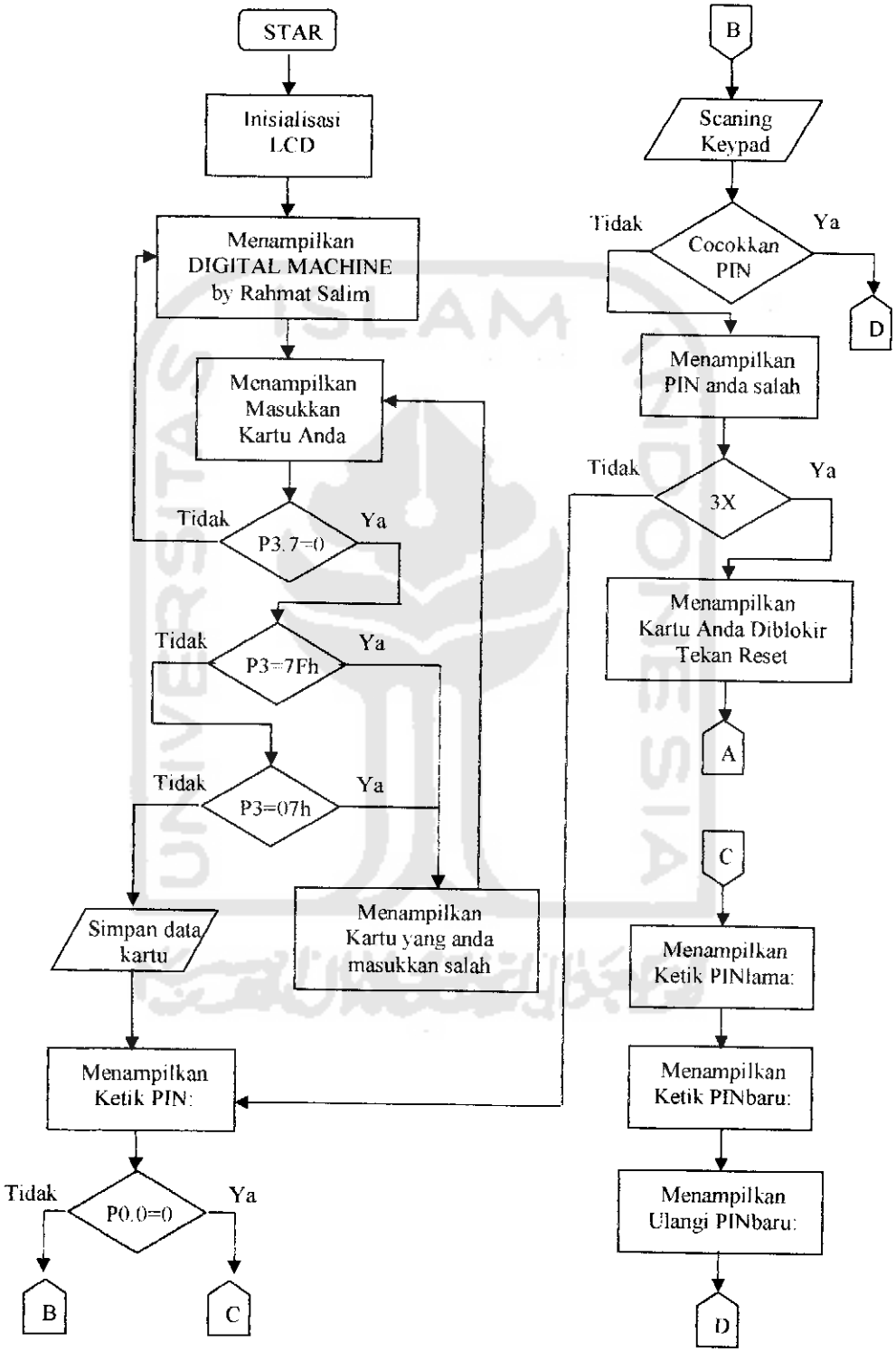
Dengan demikian untuk R_B sebesar 10 K besarnya $I_{B(Min)} < I_B$ hal ini akan membuat transistor jenuh sehingga arus yang masuk ke rangkaian penggerak pintu geser elektronik maksimum.

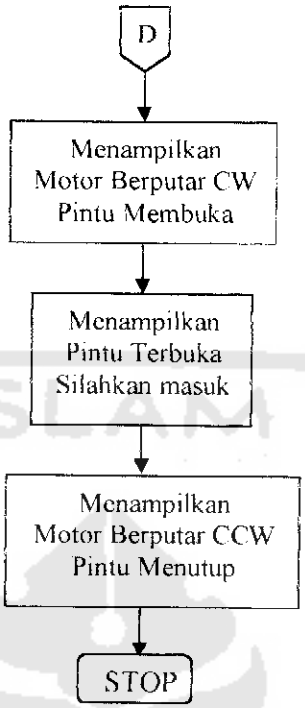
3.3. Perancangan Program

Dalam perancangan program terlebih dahulu di buat algoritmanya, yang berisi urutan proses pembuatan program, yaitu sebagai berikut:

1. Program selalu diawali dengan melakukan inisialisasi untuk memudahkan proses selanjutnya.
2. Lakukan pengecekan terhadap masukannya apakah kartu yang dimasukkan sudah benar atau tidak kalau kartu sudah benar maka data dibaca dan disimpan pada alamat yang ditentukan.
3. Masukan kode PIN dari *keypad*, kode PIN ini disimpan dialamat yang sudah ditentukan. Pembacaan *keypad* selesai jika terjadi penekanan pada tombol OK. Apabila menekan tombol *clear* maka program akan menghapus kode PIN yang sudah dimasukkan dan jika kode yang dimasukkan tersebut sudah diakhiri dengan penekanan tombol OK maka akan dilakukan proses pencocokan kode PIN yang dimasukkan dengan kode kartunya yang sudah disimpan dalam memori EEPROM.
4. Pergantian PIN dapat dilakukan setelah menekan tombol untuk mengganti PIN, kode PIN yang baru disimpan pada alamat EEPROM yang sudah disesuaikan dengan data kartu.
5. Apabila semua proses diatas dilaksanakan maka program akan membuka dan menutup pintu dengan selang waktu tertentu.

3.3.1. Diagram Alir Sistem Pembaca kartu Digital





Gambar 3.12. Diagram alir sistem pembaca kartu digital

3.3.2. Implementasi Diagram Alir

3.3.2.1. Inisialisasi.

Inisialisasi program terdiri dari pemetaan alamat RAM untuk menyimpan data yang dapat dimodifikasikan serta pendataan SFR WMCON yang merupakan register khusus yang digunakan dalam pembacaan data pada EEPROM, mengidentifikasikan Program yang dimulai pada alamat Flash 0000H, serta inisialisasi LCD seperti yang ditunjukkan dibawah ini

```
Ban_P3      equ     030h
Kartu       equ     031h
Cari_data   equ     032h
tekan       equ     033h
bounc       equ     034h
```

```
baris        equ        035h
jml_pin      equ        036h
data_baris   equ        037h
ban_pin      equ        038h
per_baris    equ        039h
data_3       equ        03Ah
data_inp     equ        03Bh
salah_pin    equ        03Ch
titip        equ        03Dh
ubah         equ        03Eh
cek          equ        03Fh
```

Pemetaan RAM untuk menyimpan data yang dapat di modifikasi dari alamat 30H – 3FH

```
WMCON        DATA       96h
EEMEN        equ        00001000b
EEMWE        equ        00010000b
WDTRST       equ        00000010b
DPS          equ        00000100b
```

Pendataan SFR WMCON untuk membaca dan menulis EEPROM

```
ORG    0000h
```

Program dimulai pada alamat flash 0000H

```
Disp:  mov    R1,#00111000b   ;inisialisasi LCD untuk 8 bit
       acall  write_inst      ;operasi dan dua line
       mov    R1,#00001110b   ;mengaktifkan kursor
       acall  write_inst
       ret
```

Inisialisasi LCD sebagai penampil 16 x 2

MILIK
PERPUSTAKAAN-FTI-UII
YOGYAKARTA

3.3.2.2. Menampilkan Karakter pada LCD

Untuk menampilkan karakter pada LCD terlebih dahulu dilakukan inisialisasi LCD untuk digunakan sebagai penampil 16 x 2 karakter seperti yang ditunjukkan pada program diatas. Untuk menuliskan instruksi pada LCD pin RS dan R/W diset pada kondisi 0 0, sedangkan untuk menuliskan data ke LCD RS dan R/W dikondisikan 1 0. Setiap selesai menuliskan instruksi maupun data ke LCD pin E diberikan *delay* untuk membuat *clock* seperti program berikut:.

```
write_inst:
        clr     P1.4            ;untuk menulis instruksi
        clr     P1.5            ;pada LCD
        Mov     P2,R1
        setb    P1.6            ;memberi clock pada pin E LCD
        acall   delay
        clr     P1.6
        acall   delay
        ret

write_data:
        setb    P1.4            ;untuk menulis data
        clr     P1.5            ;pada LCD
        mov     P2,R1
        setb    P1.6
        acall   delay
        clr     P1.6
        acall   delay
        ret
```

3.3.2.3. Pembacaan Kartu

Program pembacaan data kartu dimulai dengan pengecekan kartu apakah sudah sesuai atau belum, seperti yang sudah disampaikan dalam dasar teori di atas bahwa kode blok kartu yang pertama adalah gelap sehingga oleh *optocoupler*

dibaca 0 maka apabila bit pertama kartunya belum 0 program tidak akan melakukan pembacaan data kartu selanjutnya. Hal ini dimaksudkan agar pemasukan (posisi) kartu pada saat pembacaan harus benar (sesuai). Setelah bit pertamanya dibaca 0 maka dilakukan pengecekan data kartunya itu 1111 atau 0000 jika ya maka program akan memberi tahu bahwa data kartu salah, jika tidak maka data yang terbaca itu dibandingkan dengan data yang tersimpan di memori, jika sama maka data tersebut merupakan data kartu yang kemudian disimpan dialamat tertentu seprti program berikut:

```
        jnb     P3.7,ulang1         ;apakah P3.7=0,jika ya ke ulang1
        ljmp    terus               ;jika tidak balik ke terus
ulang1: mov     A,P3                ;baca data dari P3
        cjne    A,#7Fh,baca         ;jika tidak sama dg 7Fh lompat ke baca
        sjmp    test1
Baca:   cjne    A,#07h,baca1        ;jika tidak sama dg 07h ke baca1
        sjmp    test1
Baca1:  mov     cari_data,A         ;data disimpan dicari_data
        mov     DPTR,#user          ;set data pointer di alamat user
        mov     ban_pin,#0          ;utk alamat cacah PIN
        mov     titip,#14
cari:   clr     A       ;nolkan akumulator
        movc    A,@A+DPTR           ;ambil data yang ditunjuk pointer
        inc     DPTR                ;alamat pointer ditambah 1
        inc     ban_pin
        cjne    A,Cari_data,salah1  ;lompat ke salh1 jika data tdk sama
        mov     data_inp,A          ;jika sama simpan data di data_input
        dec     ban_pin
        mov     titip,ban_pin       ;cacah pin disimpan di titip
        mov     salah_pin,#3
        ljmp    lanjut              ;lompat ke label mulai
salah1: djnz    titip,cari          ;ulangi sebanyak data kartu
        ljmp    test1
```

3.3.2.4. Pembacaan dan Penulisan EEPROM

Pembacaan EEPROM menggunakan SFR WMCON yang mana saat pembacaa internal EEPROM bit EEMEN (bit ke 3 EMCON) harus dalam kondisi 1 seperti yang ditunjukkan pada program dibawah ini:

Isi WMCON	= 00001010b	
Isi EEMEN	= 00001000b	XOR
Isi WMCON menjadi	**= 00000010b**	

Sedangkan untuk penulisan EEPROM digunakan bit EEMWE set 1 sebelum inisialisasi program sudah selesai seperti contoh berikut:

```
WMCON     DATA     96h
EEMEN     equ      00001000b
EEMWE     equ      00010000b
WDTRST    equ      00000010b
DPS       equ      00000100b
```

Pembacaan EEPROM

```
        orl     WMCON,#EEMEN        ;enable EEPROM accesses
        mov     DPTR,#000h          ;alamat awal EEPROM
        mov     A,ban_pin
        jz      lagi                ;lompat ke lagi jika A=0
jump:   inc     DPTR                ;mencari alamat DPTR yang
        djnz    ban_pin,jump        ;cocok sesuai data kartunya
lagi:   movx    A,@DPTR
        inc     DPTR
        mov     data_inp,A
        mov     A,@R0               ;membandingkan apakah data dari
        cjne    A,data_inp,slh1     ;keypad sama dgn data PIN
        inc     R0          ;di EEPROM
        djnz    jml_pin,lagi        ;ulangi sebanyak 4x
        xrl     WMCON,#EEMEN        ;disable EEPROM accesses
```

Penulisan EEPROM

```
        orl     WMCON,#EEMEN
```

```
        orl     WMCON,#EEMWE
        mov     DPTR,#000h      ; alamat awal EEPROM
        mov     A,ban_pin
        jz      lagi0           ; lompat ke lagi jika A =0
jump0:  inc     DPTR            ; mencari alamat DPTR yang
        djnz    ban_pin,jump0   ; cocok sesuai data kartunya
lagi0:  mov     A,@R0
        movx    @DPTR,A
loop2:  mov     A,WMCON
        anl     A,#WDTRST
        jz      loop2
        inc     R0
        inc     DPTR
        djnz    jml_pin,lagi0
        xrl     WMCON,#EEMWE
        xrl     WMCON,#EEMEN
```

3.3.2.5. Pengecekan PIN

Kode PIN yang digunakan untuk mengakses alat ini terdiri dari 4 digit angka yang telah disimpan didalam EEPROM sehingga apabila tegangan catunya dimatikan data PIN masih tetap tersimpan didalam memori. Kode-kode PIN ini dimiliki oleh setiap kartu atau dengan kata lain setiap kartu memiliki kode PIN. Tabel 3.3 menunjukkan kode PIN masing-masing kartu.

Nomor PIN yang sudah dimasukkan dari *keypad* dibandingkan dengan nomor PIN yang disimpan di EEPROM jika sama akan membuka pintu secara otomatis dan jika tidak sama akan dilakukan penulisan nomor PIN lagi. Kesalahan pemasukan nomor PIN hanya diperbolehkan sebanyak 3 kali. Program pengecekan dan pembacaan kode PIN seperti program berikut ini:

```
banding_pin:
        mov     A,ubah          ;ambil data dari alamat ubah
        cjne    A,#1,tidak      ;jika 1 lompat ke ganti_pin
        ljmp    ganti_pin

tidak:  mov     R0,#7Ah         ;alamat RAM
        mov     A,titip         ;utk menyesuaikan pion dg
        mov     B,#4            ;data kartu
        mul     AB
        mov     ban_pin,A
        mov     jml_pin,#4
        orl     WMCON,#EEMEN            ;enable EEPROM accesses
        mov     DPTR,#000h      ;alamat awal EEPROM
        mov     A,ban_pin
        jz      lagi            ;lompat ke lagi jika A=0

jump:   inc     DPTR            ;mencari alamat DPTR yang
        djnz    ban_pin,jump    ;cocok sesuai data kartunya

lagi:   movx    A,@DPTR
        inc     DPTR
        mov     data_inp,A
        mov     A,@R0           ;membandingkan apakah data dari
        cjne    A,data_inp,slh1 ;keypad sama dgn data PIN
        inc     R0      ;di EEPROM
        djnz    jml_pin,lagi            ;ulangi sebanyak 4x
        xrl     WMCON,#EEMEN            ;disable EEPROM accesses
        mov     A,cek
        cjne    A,#0,ceker
        mov     R1,#00000010b
        acall   write_inst      ;menampilkan SILAHKAN
        mov     DPTR,#cocok             ;MASUK pada LCD
        acall   baris1
```

Tabel 3.3. Kode PIN tiap kartu.

No	Data Kartu	Kode PIN
1	0FH	1213
2	17H	1214
3	1FH	1324
4	27H	3425
5	2FH	4356
6	37H	9412
7	3FH	9578
8	47H	9971
9	4FH	0202
10	57H	1111
11	5FH	7887
12	65H	6541
13	6FH	9757
14	77H	6903

3.3.2.6. Pembacaan *Keypad*

Pembacaan *keypad* dilakukan dengan sistem *scaningpad* yang mana P3.0 – P3.2 sebagai input dan P1.0 - P1.3 sebagai output.

```
scan_tombol:
        clr     P3.0
        clr     P3.1
        clr     P3.2
lomlagiah:
        jnb     P1.0,lompatah
```

```
        jnb     P1.1,lompatah
        jnb     P1.2,lompatah
        jnb     P1.3,lompatah
        sjmp    start
;
lompatah:
        djnz    bounc,lomlagiah
        mov     bounc,#10
        mov     DPTR,#tombol
        mov     baris,#3
        mov     data_3,#0
        mov     data_baris,#0
        mov     per_baris,#0
        clr     P3.0
        setb    P3.1
        setb    P3.2
balik:  jb      P1.0,key1       ; pengecekan pada kolom pertama
        djnz    bounc,balik
        mov     bounc,#10
ambil:  mov     A,per_baris     ; bila P3.0 = 0 ambil data
        add     A,data_3
        movc    A,@A+DPTR
        mov     data_inp,A
        mov     @R0,data_inp    ; data disimapan di alamat yang
        inc     R0              ; ditunjuk reg 0 (7Ah)
        mov     A,data_inp
        cjne    A,#0Ch,trus     ; pengecekan thd tombol clear
        mov     A,cek
        cjne    A,#0,mulih0
        mov     A,ubah
        cjne    A,#0,mulih0
        ljmp    clear
trus:   cjne    A,#0AAh,trus1   ; pengecekan thd tombol OK
        ljmp    banding_pin
trus1:  lcall   ketik           ; tampilkan * pada LCD
        ljmp    start
;
```

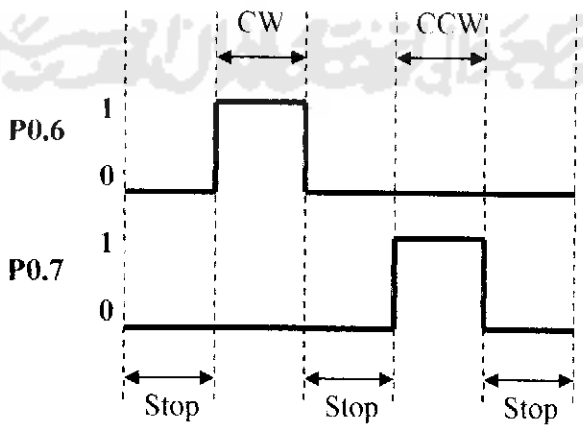
```
key1:   inc     per_baris       ; pengecekan pada kolom kedua
bonc1:  jb      P1.1,key2
        djnz    bounc,bonc1
        mov     bounc,#10
        sjmp    ambil
;
key2:   inc     per_baris       ; pengecekan pada kolom ketiga
bonc2:  jb      P1.2,key3
        djnz    bounc,bonc2
        mov     bounc,#10
        sjmp    ambil
;
key3:   inc     per_baris       ; pengecekan pada kolom ketiga
bonc3:  jb      P1.3,key4
        djnz    bounc,bonc3
        mov     bounc,#10
        sjmp    ambil
;
key4:   inc     per_baris
        dec     baris
        mov     A,baris         ; inisialisasi scanning pada baris kedua
        cjne    A,#2,next
        Setb    P3.0
        clr     P3.1
        setb    P3.2
        ljmp    balik

next:   mov     A,baris
        cjne    A,#1,next1
        Setb    P3.0
        setb    P3.1
        clr     P3.2
        ljmp    balik

next1:  ljmp    start
```

3.3.2.7. Mengaktifkan Pintu Geser Elektronik

Program untuk mengaktifkan pintu geser elektronik menggunakan 2 bit port yang memiliki kondisi yang berlawanan, yaitu P0.6 dan P0.7. . Jika kondisi P0.6 sama dengan '1' dan kondisi P0.7 sama dengan '0', akan mebuat motor berputar kekanan (CW). dan pintu akan membuka. Dan jika kondisi P0.6 sama dengan '0' dan kondisi P0.7 sama dengan '1', akan mebuat motor berputar kekiri (CCW) dan pintu akan, kondisi P0.6 dan P0.7 harus sama yaitu 00 atau 11, seperti diperlihatkan pada program berikut:

```
setb    P0.6        ;untuk mebuat motor putar kekanan
clr     P0.7        ;selama periode delay
acall   sela
clr     P0.6        ;utk mematikan motor
acall   sela
clr     P0.6        ;program utk membuat motor putar
setb    P0.7        ;kekiri selama periopde delay
acall   sela
clr     P0.7        ;utk mematikan motor
lcall   rstlagi
```

dari keterangan diatas siklus putaran motor dapat dijelaskan pada Gambar 3.13.



Gambar. 3.13. Siklus putaran motor