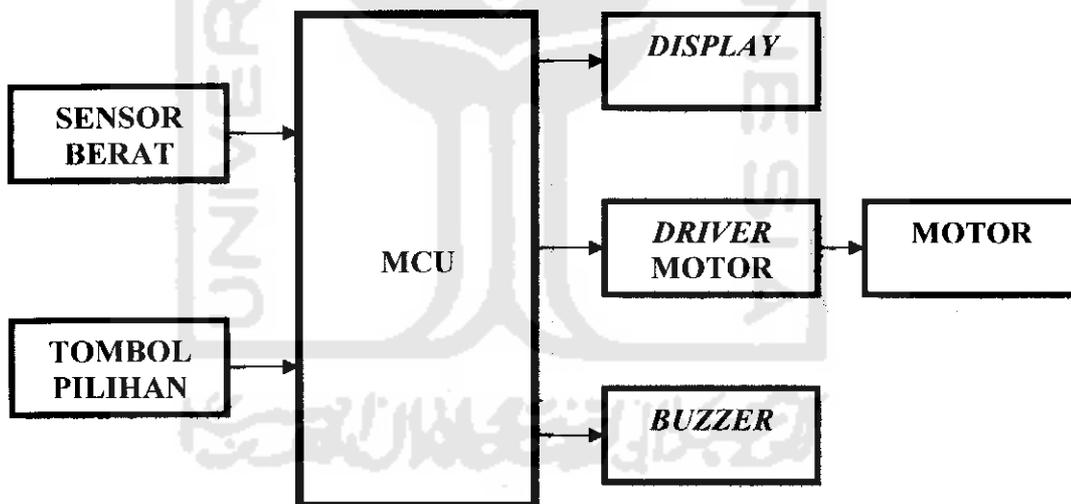


### BAB III

#### PERANCANGAN SISTEM

##### 3.1. Gambaran Umum Sistem

Sistem pengendali *lift* ini terdiri dari beberapa rangkaian yaitu: rangkaian sensor berat beban, rangkaian *keypad* (tombol pilihan), rangkaian sensor tingkat lantai, rangkaian mikrokontroler, rangkaian *display*, dan rangkaian *driver* untuk motor.



Gambar 3.1. Diagram Blok Sistem

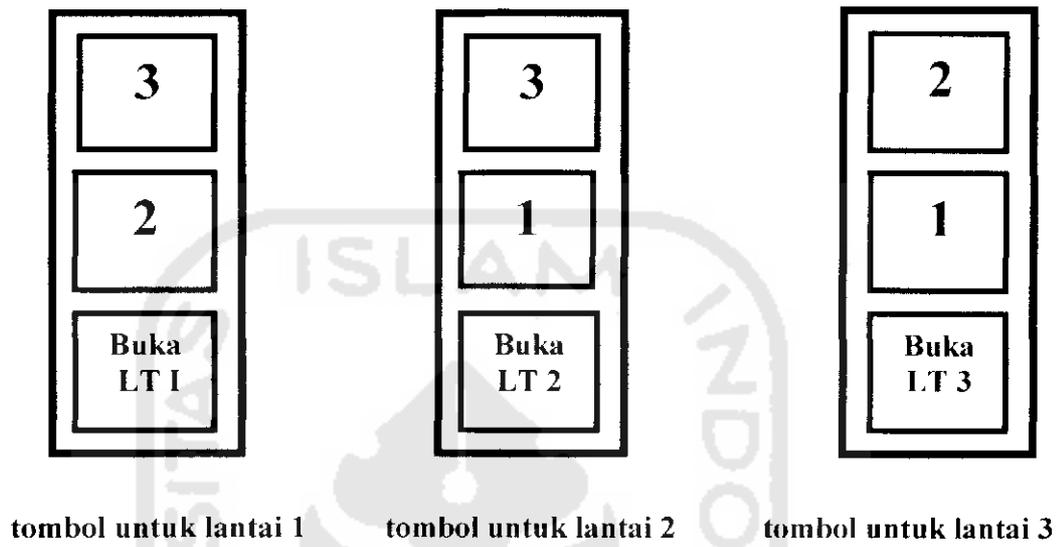
Gambar 3.1 merupakan diagram blok dari keseluruhan sistem. Proses dimulai dari pembacaan sensor berat yang ditempatkan pada alas lift, sensor ini berupa dua saklar yang ditempatkan di bawah lantai lift yang akan bekerja bila ada beban berlebih yang ditempatkan di atas sensor tersebut. Keluaran dari sensor ini akan diumpungkan pada mikrokontroler apakah beban yang diberikan pada sensor tersebut melebihi batas maksimal ataukah masih dibawah batas maksimal. Selain mendapat masukan dari sensor berat mikrokontroler juga akan menunggu masukan dari tombol pilihan lantai. Setelah mendapat masukan dari sensor berat dan tombol pilihan lantai maka mikrokontroler akan mengeksekusi permintaan yang diberikan dengan *output* berupa *display*, *motor* dan *buzzer*.

## **3.2. Perancangan Perangkat Keras**

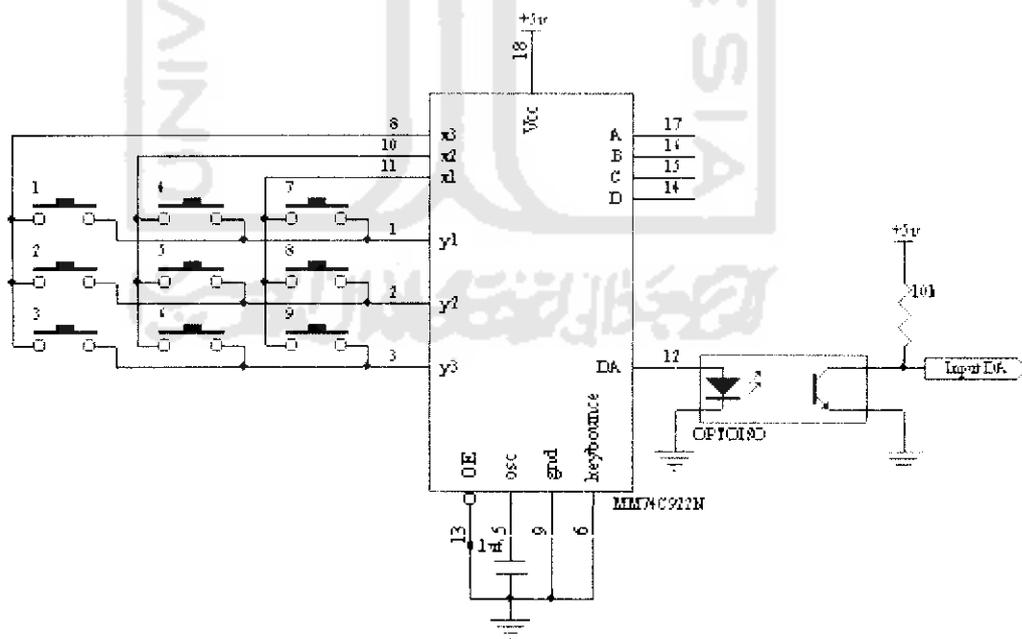
### **3.2.1. Perancangan Tombol Pilihan**

Tombol pilihan dibuat dengan sebuah IC *keypad* dengan nomor seri MM74C922N. IC ini merupakan IC *keypad* dengan 4 bit keluaran berupa kode-kode biner sesuai dengan tombol yang ditekan.

Tombol pilihan lantai dibuat dengan matriks 3x3 dengan konfigurasi yang berbeda tiap lantainya. Untuk tombol pada lantai 1 dibuat dengan konfigurasi 3, 2 dan buka. Untuk tombol pada lantai 2 dibuat dengan konfigurasi 3,1 dan buka. Untuk tombol pada lantai 3 dibuat dengan konfigurasi 2,1 dan buka. Susunan tombol pilihan lantai 1, 2 dan 3 dapat dilihat pada gambar 3.2.



Gambar 3.2. Tombol Pilihan Lantai

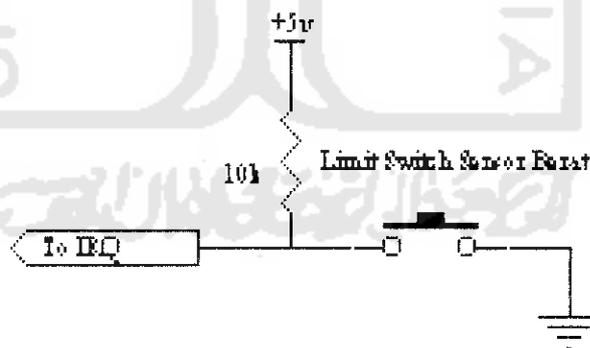


Gambar 3.3. Rangkaian tombol Pilihan

### 3.2.2. Perancangan Sensor Berat

Rangkaian sensor berat ini di buat dengan menggunakan dua saklar *switch* yang dihubung secara paralel. Tujuan digunakannya dua saklar adalah agar dalam pendeteksiannya lebih sensitif walau beban tidak diletakkan tepat diatas saklar. Saklar *switch* ini mempunyai tiga kaki yaitu C (*Common*), NC (*Normally Close*) dan NO (*Normally Open*). Pada perancangan sensor berat ini ground dihubungkan dengan C dan 5V pada NC. Sehingga apabila diukur dengan multimeter tegangan pada saklar yang belum ditekan akan selalu 5V dan bila ditekan atau ada beban maka tegangan akan 0V.

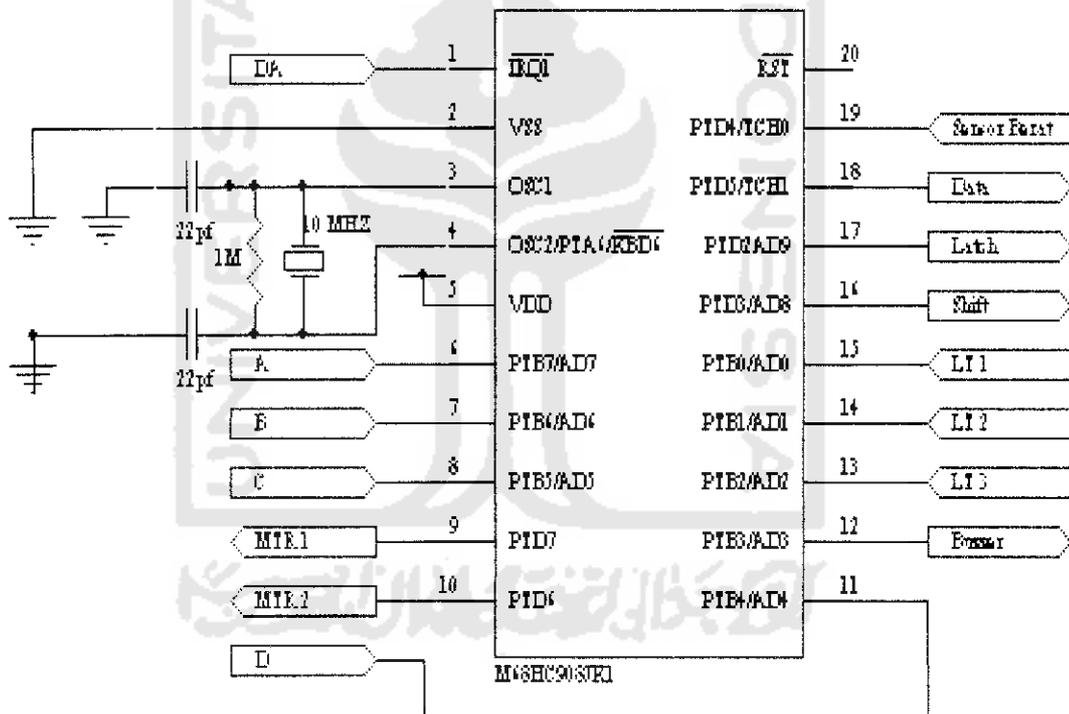
Tegangan 5V ini diumpankan ke port IRQ (*Interrupt Request*) yang terus memantau apakah saklar itu dalam kondisi *ON* atau *OFF*.



Gambar 3.4. Rangkaian Sensor Berat

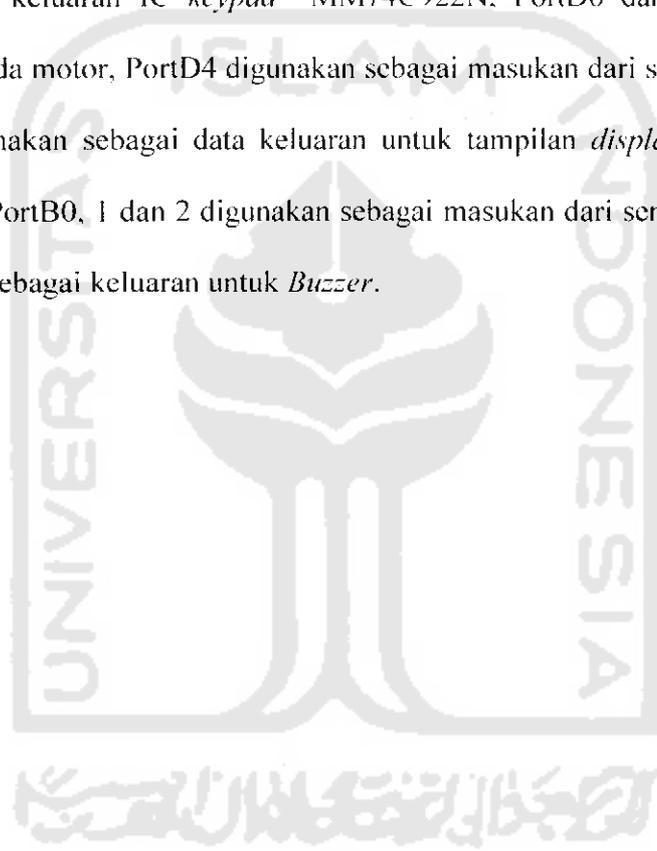
### 3.2.3. Perancangan *Microcontroller Unit (MCU)*

Sistem pengendali yang dibuat menggunakan mikrokontroler keluaran motorola dengan nomor seri MC68HC908JK1. Gambar 3.5 merupakan gambar ke-20 pin dari MC68HC908JK1 yang masing-masing pinnya sudah difungsikan untuk pengendalian seluruh sistem.



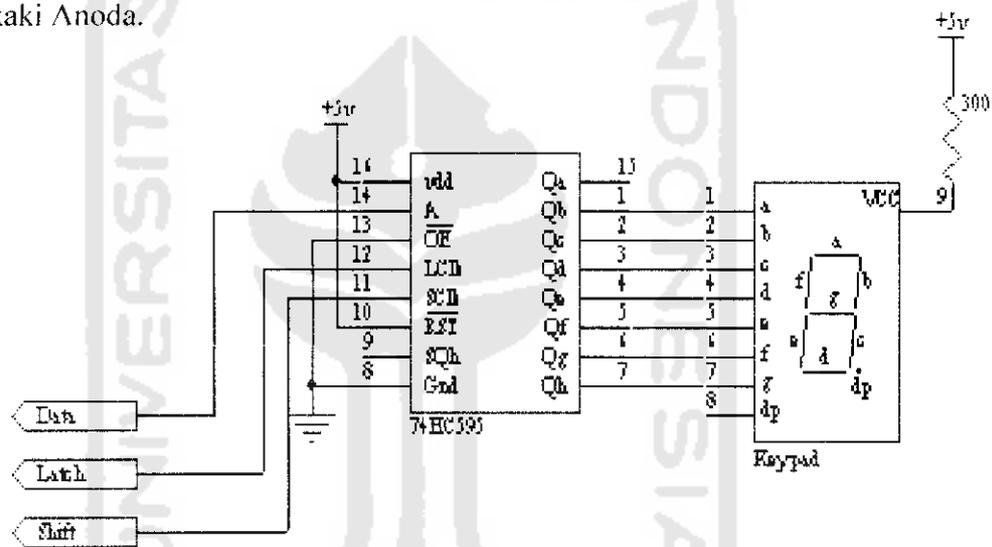
Gambar 3.5. *Port Input Output* yang digunakan pada MCU

Secara keseluruhan semua pin pada mikrokontroler MC68HC908JK1 tersebut difungsikan sebagai *port Input/Output (I/O)*, yaitu : Port IRQ1 digunakan sebagai masukan dari *Optocoupler*, PortB4, 5, 6 dan 7 sebagai masukan untuk menerima data 4 bit dari keluaran IC *keypad* MM74C922N, PortD6 dan 7 digunakan untuk keluaran pada motor, PortD4 digunakan sebagai masukan dari sensor berat, PortD2, 3 dan 5 digunakan sebagai data keluaran untuk tampilan *display* pada *shift register* 74HC595, PortB0, 1 dan 2 digunakan sebagai masukan dari sensor lantai dan PortB3 digunakan sebagai keluaran untuk *Buzzer*.



### 3.2.4. Perancangan *Display*

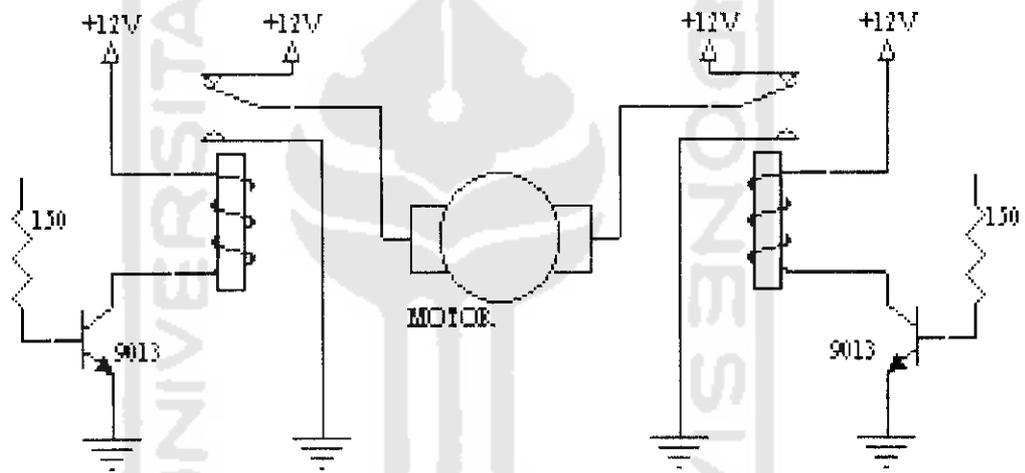
Dalam pembuatan *display* digunakan 1 buah *7-segment common anoda* sebagai tampilan tingkat lanjut. Dalam perancangan *display* ini menggunakan IC 74HC595 yang berfungsi mengubah data serial menjadi data paralel. Tipe *7-segment* ini adalah *Common Anoda*, sehingga pada perancangannya tegangan 5V dimasukkan pada kaki Anoda.



Gambar 3.6. Rangkaian *Display*

### 3.2.5. Perancangan Rangkaian *Driver*

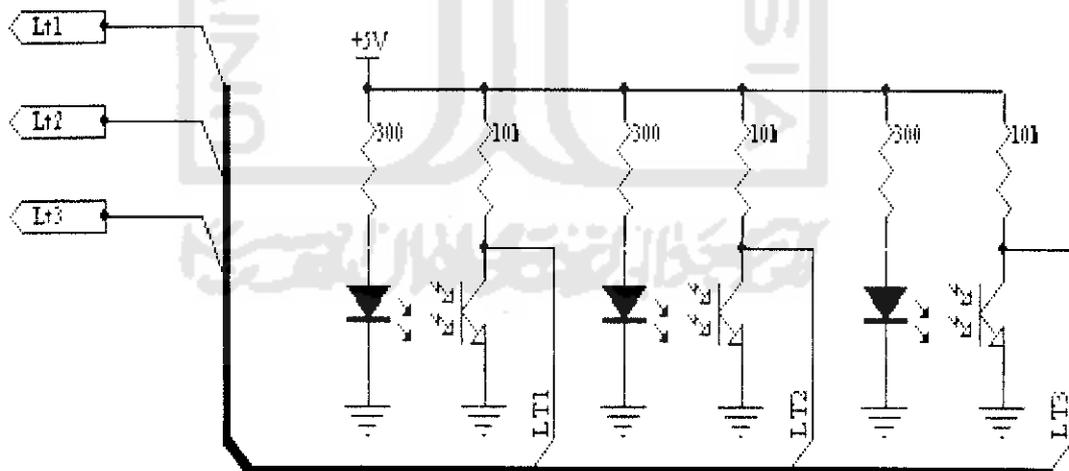
Rangkaian *driver* motor menggunakan *relay* untuk memutus dan menyambungkan tegangan 12volt DC ke motor. Pada perancangan sistem ini digunakan dua *relay* untuk membalik arah putaran motor kiri dan kanan. *Relay* diaktifkan oleh sebuah transistor yang difungsikan sebagai *switch* transistor.



Gambar 3.7. Rangkaian *driver* motor

### 3.2.6. Perancangan Sensor Lantai

Untuk menghentikan lift di lantai 1, 2 dan 3 digunakan Infra Merah dan *fototransistor*. Infra merah digunakan untuk menghasilkan cahaya yang mempunyai intensitas tinggi dan fototransistor digunakan sebagai saklar otomatis yang akan memberi *interrupt* pada mikrokontroler bila terkena cahaya dari Infra merah. Dalam perancangan sistem ini Infra merah diberi tegangan 5V langsung dari *power supply*. *Fototransistor* yang digunakan adalah fototransistor besi yang mempunyai 3 kaki yaitu C (*Collector*), B (*Basis*) dan E (*Emitter*). Pada sistem ini kaki *phototransistor* yang digunakan adalah kaki C dan E, karena basis sudah mendapat masukan dari Infra merah yang berupa cahaya berintensitas tinggi.



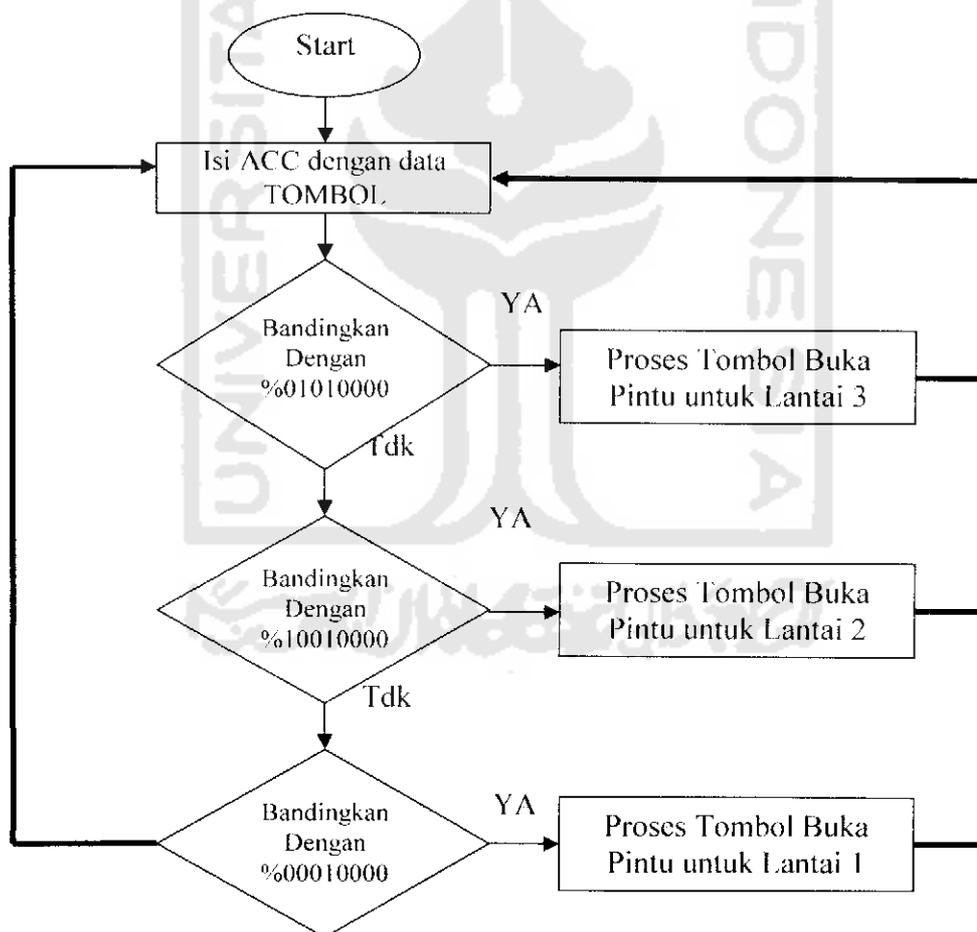
Gambar 3.8. Rangkaian Sensor Lantai

### 3.3. PERANCANGAN PROGRAM

#### 3.3.1. Diagram Alir Proses Pengendalian Sistem

Proses pengendalian sistem yang dikerjakan mikrokontroller secara garis besar dapat dibagi menjadi empat bagian utama, yaitu: Program utama, Program *external Interrupt*, Proses Tombol buka pintu, Proses naik dan turun.

##### 3.3.1.1. Diagram Alir Program Utama



Gambar 3.9. Diagram alir Program Utama

Diagram alir diatas menggambarkan program utama dari keseluruhan sistem yang terdapat didalam program mikrokontroler. Ketika start pertama kali maka *Accumulator* akan diisi dengan data dari tombol masukan. Data-data ini sebelumnya telah didapat dari IC Keypad MM74C922N. Data-data dalam bentuk biner tersebut merupakan perintah “BUKA” dalam konfigurasi tombol yang dibuat.

Setelah *accumulator* diisi dengan data-data tombol maka mikrokontroler akan men-*scan* data-data tersebut. Data-data tersebut merupakan perintah buka pada masing-masing lantai yaitu : %01010000 merupakan perintah buka untuk lantai 3, %10010000 merupakan perintah buka untuk lantai 2 dan %00010000 merupakan perintah buka untuk lantai 1. Pada gambar diatas ketika terjadi pertanyaan bandingkan dengan %01010000, %10010000 atau %00010000 maka mikrokontroler akan memproses apakah jawabannya YA atau Tidak. Ketika terjadi jawaban Tidak maka mikrokontroler akan bertanya pada lantai selanjutnya dan begitu seterusnya apabila jawaban yang didapat selalu Tidak maka program akan berjalan pada sirkulasi tersebut. hal ini berarti tidak ada masukan atau dalam arti lain alat tidak berkerja.

Tetapi ketika terjadi jawaban YA maka mikrokontroler akan mengeksekusi perintah tersebut sesuai dengan data yang ditekan. Ketika *switch* yang ditekan adalah yang mempunyai data %01010000 maka itu merupakan perintah buka pintu untuk lantai 3, %10010000 maka itu merupakan perintah buka pintu untuk lantai 2 dan %00010000 untuk perintah buka pintu untuk lantai 1.

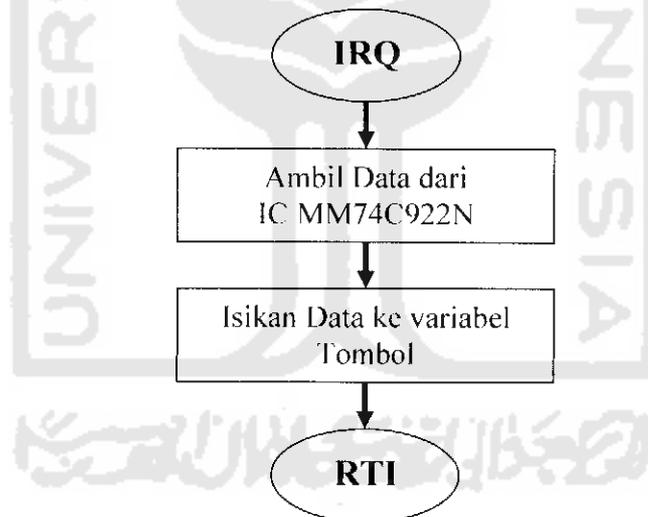
```

;RUTIN PEMROSESAN DATA DARI TOMBOL
MAIN_LOOP:  BRASET  Sensor_LT1,Scan1
            MOV      #!1,Desimal_Lantai
Scan1      BRASET  Sensor_LT2,Scan2
            MOV      #!2,Desimal_Lantai
Scan2      BRASET  Sensor_LT3,Proses_Tombol
            MOV      #!3,Desimal_Lantai

Proses_Tombol LDA      Tombol
            CBEQA    #%01010000,Proses3
            CBEQA    #%10010000,Proses2
            CBEQA    #%00010000,Proses1
            JMP      MAIN_LOOP

```

### 3.3.1.2. Diagram Alir Program *External Interrupt*



Gambar 3.10. Diagram alir Program *External Interrupt*

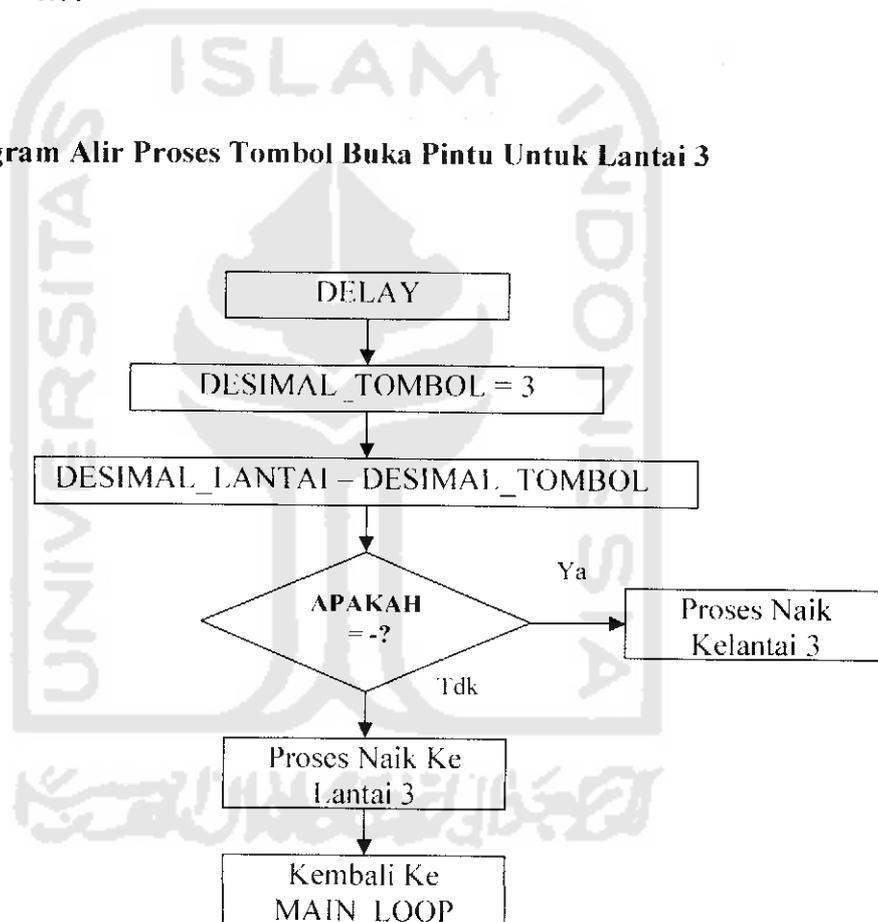
Gambar diatas menunjukkan ketika proses setelah tombol ditekan. Proses diawali dari mengambil data 4 bit dari ic 74C922N yang selajutnya data tersebut dimasukkan ke *variabel* Tombol. Data akan digunakan untuk proses *scanning* di program utama dan program permintaan lantai.

```

IRQ  BITSET  ACK1
        MOV    PortB,Tombol
        BITCLR Data0
        BITCLR Data1
        BITCLR Data2
        BITCLR Data3
        RTI

```

### 3.3.1.3. Diagram Alir Proses Tombol Buka Pintu Untuk Lantai 3



Gambar 3.11. Diagram alir Proses Buka Pintu untuk Lantai 3

Proses ini merupakan proses Buka pintu untuk lantai 3. Dimulai dengan penekanan tombol BUKA pada lantai 3. Desimal\_Lantai diisi dari display 7-segment, dan Desimal\_Tombol diisi dari tombol Buka yang ditekan. Proses konversi variabel Desimal\_Tombol dari biner menjadi heksa dengan menggunakan program *MOV #!3,Desimal\_Tombol* (Isikan 3 desimal ke variabel Desimal\_Tombol). Setelah tombol 3 ditekan maka Desimal lantai akan dikurangi Desimal\_tombol. Apakah hasilnya plus atau minus, apabila plus maka lantai akan naik kelantai 3 dan bila minus lantai juga akan tetap naik kelantai 3, hal ini terjadi karena nominal 3 pada sistem ini adalah yang terbesar sehingga hasilnya tidak mungkin plus.

Sehingga ketika lantai berada pada lantai 1 atau 2 dan ketika tombol BUKA pada lantai 3 ditekan maka lantai akan bergerak menuju lantai 3.

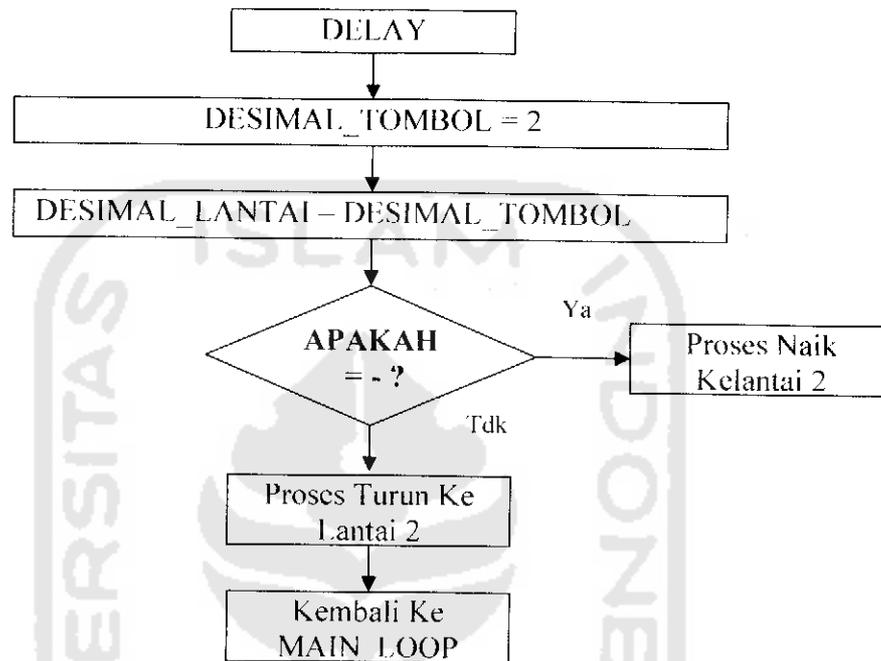
\*\*\* RUTIN UNTUK TOMBOL BUKA PINTU LANTAI 3 \*\*\*

```

Proses3      LDX      #!10
Proses3_12   JSR      Delay
              DECX
              BNE     Proses3_12
              MOV     #!3,Desimal_Tombol
              LDA     Desimal_Lantai
              SUB     Desimal_Tombol
              BMI     Loncat_Naik3
              BPL     Loncat_Naik3
              JMP     MAIN_LOOP

```

### 3.3.1.4. Diagram Alir Proses Tombol Buka Pintu Untuk Lantai 2



Gambar 3.12. Diagram alir Proses Buka Pintu untuk Lantai 2

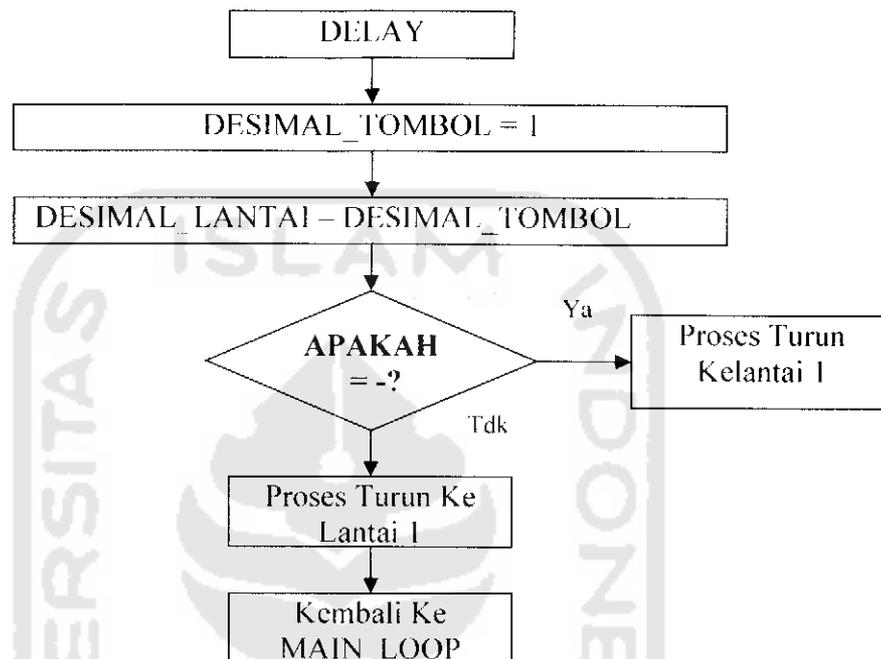
Proses diatas hampir sama dengan proses Buka pintu pada lantai 3, yang membedakan adalah ketika didapat jawaban YA maka lift akan melakukan proses naik kelantai 3.

\*\*\* RUTIN UNTUK TOMBOL BUKA PINTU LANTAI 2 \*\*\*

```

Proses2      LDX      #!10
Proses2_12   JSR      Delay
              DECX
              BNE     Proses2_12
              MOV     #!2,Desimal Tombol
              LDA     Desimal_Lantai
              SUB     Desimal_Tombol
              BMI     Loncat_Naik2
              BPL     Turun2
              JMP     MAIN_LOOP
  
```

### 3.3.1.5. Diagram Alir Proses Tombol Buka Pintu Untuk Lantai 1



Gambar 3.13. Diagram alir Proses Buka Pintu untuk Lantai 1

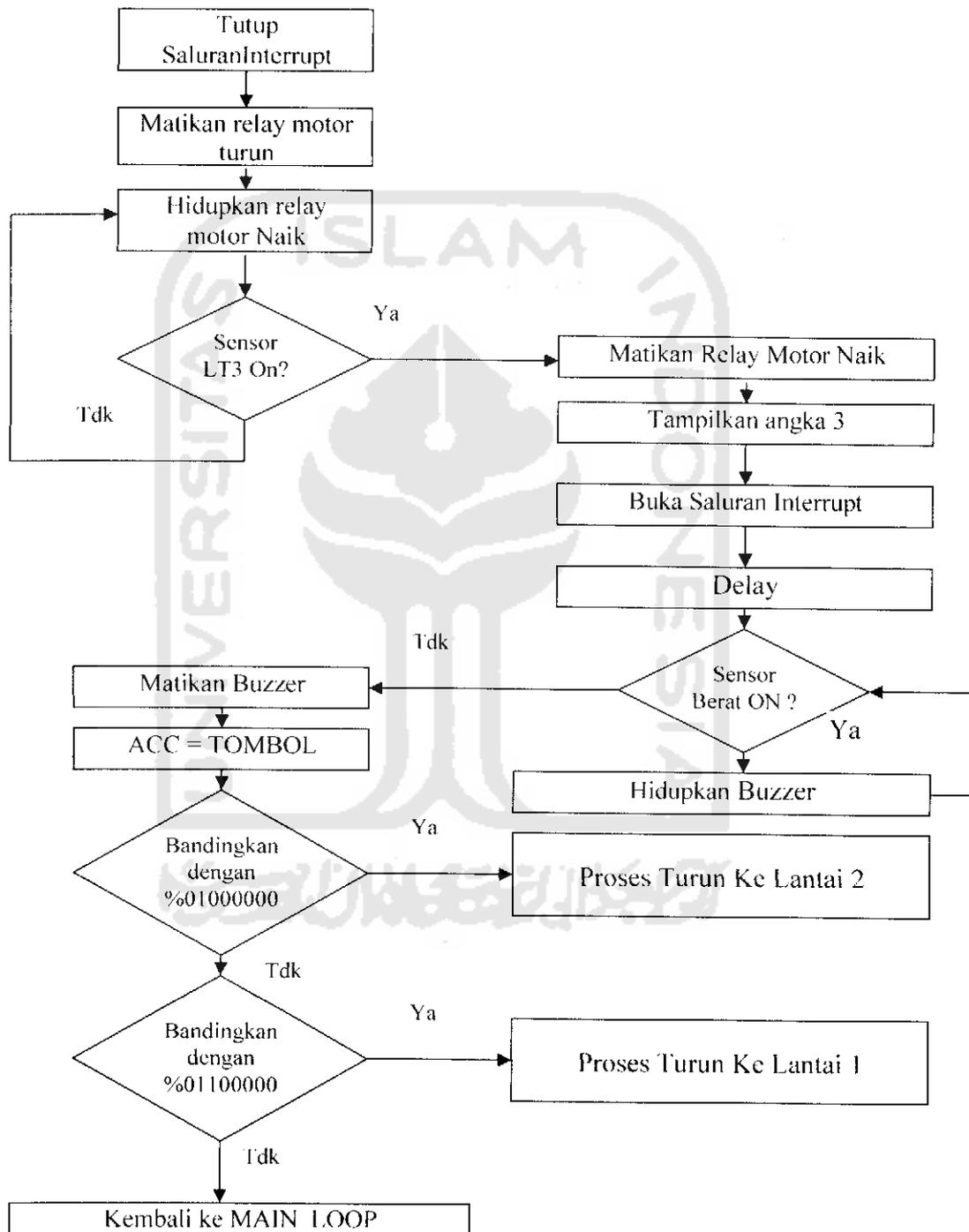
Proses diatas hampir sama dengan proses pada buka pintu untuk lantai 2 dan 3 yang membedakan nya adalah ketika didapat jawaban YA atau Tdk lift akan melakukan proses turun kelantai 1.

\*\*\* RUTIN UNTUK TOMBOL BUKA PINTU LANTAI 1 \*\*\*

```

Proses1      LDX      #!10
Proses1_12   JSR      Delay
              DECX
              BNE     Proses1_12
              MOV     #!1.Desimal_Tombol
              LDA     Desimal_Lantai
              SUB     Desimal_Tombol
              BPL     Turun1
              BMI     Turun1
              JMP     MAIN_LOOP
Loncat_Naik2 JMP     NAIK2
Loncat_Naik3 JMP     NAIK3
  
```

### 3.3.1.6. Diagram Alir Proses Naik ke Lantai 3



Gambar 3.14. Diagram alir Proses Naik Ke Lantai 3

Proses naik kelantai 3 ini dimulai dengan penekanan tombol buka pada lantai 3. Setelah tombol buka pada lantai 3 ditekan maka mikro akan mematikan *relay* motor turun dan menghidupkan *relay* motor naik (pada proses ini mikro mematikan saluran *interrupt*). Proses ini terjadi ketika posisi lantai berada pada lantai 1 dan 2. Ketika lantai sudah sampai pada lantai 3 dan menyentuh sensor lantai maka mikro akan mematikan relay motor naik dan kemudian mikro akan menampilkan angka 3 melalui IC 74HC595 dan membuka saluran *interrupt* (menunggu perintah kelantai mana).

Setelah diberi masukan lantai maka mikro akan memberi dua *delay* yaitu *delay* buka-tutup pintu dan *delay* untuk sensor berat. Ketika sensor berat diberi beban maka mikro akan menunggu apakah terjadi interupsi pada sensor berat (interupsi disini berupa pendeteksian beban apakah sudah maksimal atau tidak). Apabila beban sudah maksimal maka mikro akan menghidupkan *buzzer* dan motor tidak akan bergerak selama beban tidak dikurangi. Dan program akan berputar disitu.

Apabila beban berada pada kondisi yang diperbolehkan maka mikro akan mematikan *buzzer* dan mengisi *accumulator* dengan data tombol, apakah data tersebut adalah %01000000 jika ya maka akan masuk ke proses turun 2, jika %01100000 maka akan masuk ke proses turun 1 dan jika tidak ada masukan (perintah) tombol maka program akan lari ke MAIN\_LOOP dan membuka saluran interupsi.

*:RUTIN UNTUK MEMBUAT LIFT NAIK KE LANTAI3*

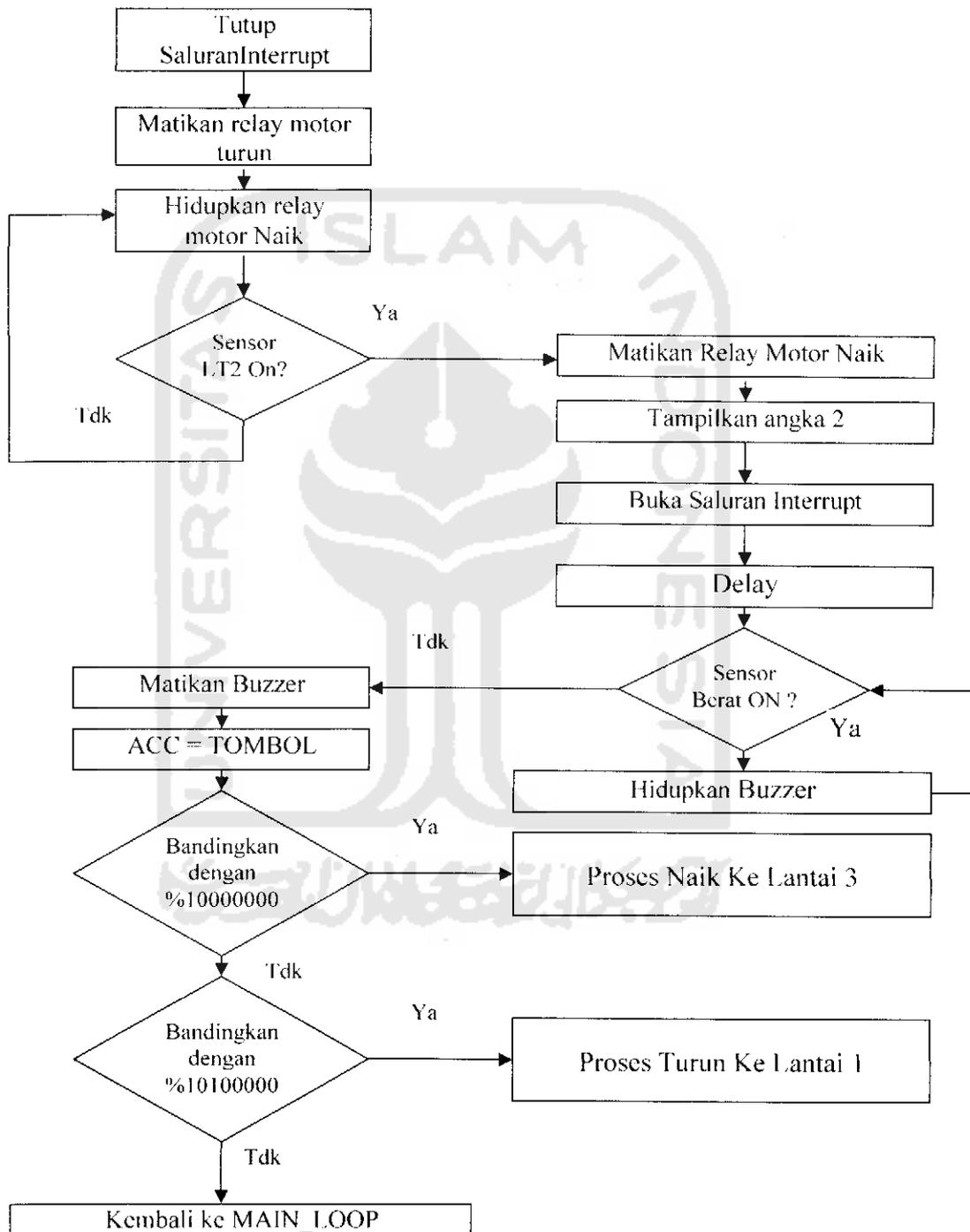
```

Naik3      SEI
           BITCLR   Mtr_Naik
           BITCLR   Mtr_Turun
           NOP
ScanLT2    BITSET   Mtr_Naik
           BRACLR   Sensor_LT3,Tampil2
           BRASET   Sensor_LT2,ScanLT2
           LDA      #S48
           JSR      SPI
Tampil2    BRA      ScanLT2
           BITCLR   Mtr_Naik
           LDA      #S60
           JSR      SPI
           MOV      #!3.Desimal_Lantai
           CLI
           LD      #!20
Proses7_1  JSR      Delay
Proses4_17 BRASET   Sensor_Berat,Proses4_16
           BITSET   Buzzer
           BRA      Proses4_17

Proses4_16 BITCLR   Buzzer
           DECX
           BNE     Proses7_1
           LDA     Tombol
           CMP     #%01000000
           BEQ     Loncat_Turun2
           CMP     #%01100000
           BEQ     Loncat_Turun1
           JMP     MAIN_LOOP

```

### 3.3.1.7. Diagram Alir Proses Naik ke Lantai 2



Gambar 3.15. Diagram alir Proses Naik Ke Lantai 2

Proses naik kelantai 2 ini secara umum hampir sama dengan proses naik kelantai 3 yang membedakan adalah perintah tombol dilantai 2 yaitu naik ke lantai 3 dan turun kelantai 1.

*;RUTIN UNTUK MEMBUAT LIFT NAIK KE LANTAI2*

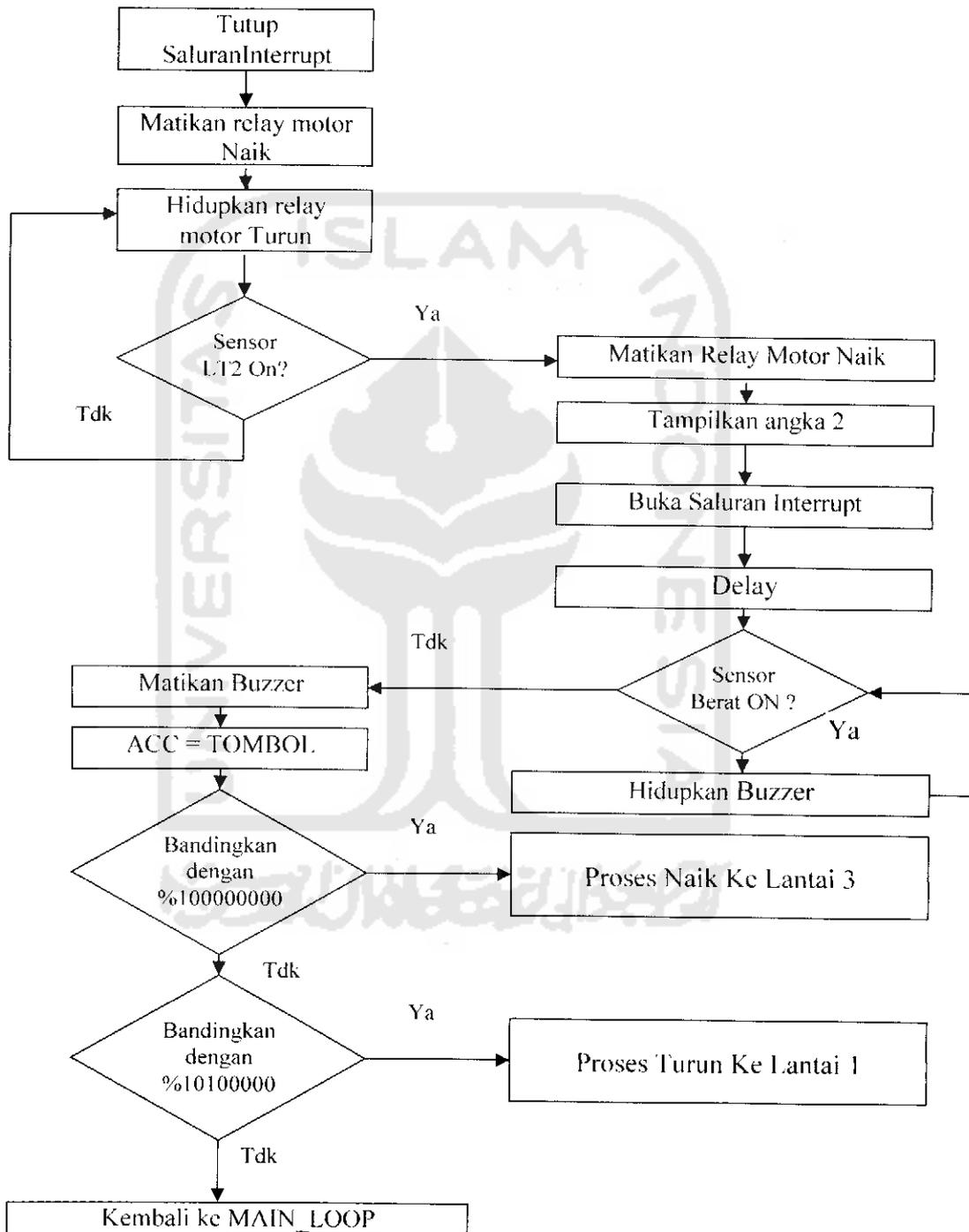
```

Naik2      SEI
           BITCLR   Mtr_Naik
           BITCLR   Mtr_Turun
           NOP
Terus2     BITSET   Mtr_Naik
           BRASET   Sensor_LT2,S
           BITCLR   Mtr_Naik
           LDA      #S48
           JSR      SPI
           MOV      #!2,Desimal_Lantai
           CLI
           LDX      #!20
Proses6_1  JSR      Delay
Proses4_15 BRASET   Sensor_Berat,Proses4_14
           BITSET   Buzzer
           BRA      Proses4_15

Proses4_14 BITCLR   Buzzer
           DECX
           BNE     Proses6_1
           LDA     Tombol
           CMP     #%10000000
           BEQ     Naik3
           CMP     #%10100000
           BEQ     Loncat_Turun1
           JMP     MAIN_LOOP
Loncat_Turun1 JMP     Turun

```

### 3.3.1.8. Diagram Alir Proses Turun ke Lantai 2



Gambar 3.16. Diagram alir Proses Turun Ke Lantai 2

Proses turun kelantai 2 hampir sama dengan proses naik kelantai 2 yang membedakan adalah ketika setelah perintah SEI mikro mematikan relay motor naik.

*:RUTIN UNTUK MEMBUAT LIFT TURUN KE LANTAI2*

```

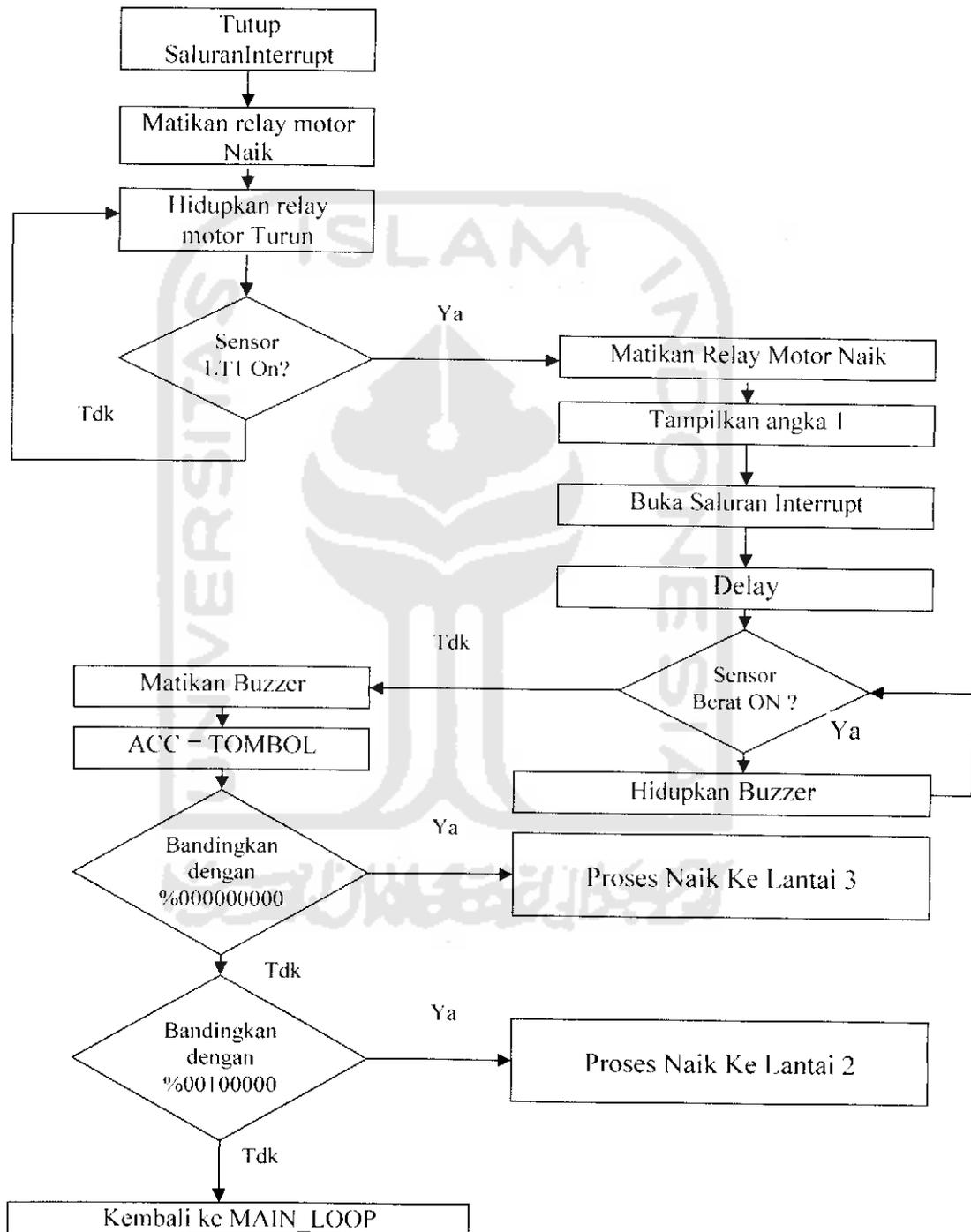
Turun2      SEI
            BITCLR  Mtr_Naik
            BITCLR  Mtr_Turun
            NOP
            BITSET   Mtr_Turun

            BRASET  Sensor_LT2,S
            BITCLR  Mtr_Turun
            LDA     #S48
            JSR     SPI
            MOV     #!2,Desimal_Lantai

            CLI
            LD      #!20
Proses5_1   JSR     Delay
Proses4_13  BRASET  Sensor_Berat,Proses4_12
            BITSET  Buzzer
            BRA     Proses4_13

Proses4_12  BITCLR  Buzzer
            DECX
            BNE     Proses5_1
            LDA     Tombol
            CMP     %%10000000
            BEQ     Naik3
            CMP     #%10100000
            BEQ     Turun1
            JMP     MAIN_LOOP
  
```

### 3.3.1.9. Diagram Alir Proses Turun ke Lantai 1



Gambar 3.17. Diagram alir Proses Turun Ke Lantai 1

Proses turun kelantai 1 ini hampir sama dengan proses turun ke lantai 2 yang membedakan adalah perintah tombol dilantai 3 yaitu naik ke lantai 3 dan naik kelantai 2.

*:RUTIN UNTUK MEMBUAT LIFT TURUN KE LANTAI*

```

Turun1      SEI
            BITCLR   Mtr_Naik
            BITCLR   Mtr_Turun
            NOP
ScanLT      BITSET   Mtr_Turun
            BRACLR   Sensor_LT1,Tampil1
            BRASET   Sensor_LT2,ScanLT
            LDA      #$48
            JSR      SPI
Tampil1     BRA      ScanLT
            BITCLR   Mtr_Turun
            LDA      #$F2
            JSR      SPI
            MOV      #!1,Desimal_Lantai
            CLI
Proses4_1   LDX      #!20
            JSR      Delay
Proses4_11  BRASET   Sensor_Berat,Proses4_10
            BITSET   Buzzer
            BRA      Proses4_11
Proses4_10  BITCLR   Buzzer
            DECX
            BNE     Proses4_1
            LDA     Tombol
            CMP     #%00000000
            BEQ     Loncat_Naik3
            CMP     #%00100000
            BEQ     Naik2
            JMP     MAIN_LOOP

```