

PERANCANGAN MINIATUR SISTEM LIFT 4 LANTAI DENGAN MENGGUNAKAN MIKROKONTROLLER AT89S51

Yudhi Andrian¹

Rika Rosnelly²

Utawi Handika Sari³

Email :

yudhi@potensi-utama.ac.id, rika@potensi-utama.ac.id, ika@potensi-utama.ac.id

Diterima: 6 Agustus 2009 / Disetujui: 27 Agustus 2009

ABSTRACT

The bug buildings that have more than one floor needed the stairs that can connect one floor to another floor. However, by using the stairs are still in efficient. Thus, it is needed an elevator that can be used to connect the whole floors in the building. To control an elevator needed a motor and control system. In this research, a motor that is used is that Motor Stepper AT89S51. Motor Stepper is used to lift up and down the elevator. While Microcontroller is used to feed the data into Motor Stepper so that Motor Stepper can move like clock rotation or inverse (lift up and down the elevator). To know that the must be down or up used manual button. This button is placed on each floor so Microcontroller can detect the pressing button to decide the elevator that can move up and down.

Kata kunci : AT89S51, Mikrokontroller, Motor Stepper

Abstraksi

Bangunan yang memiliki lebih dari satu lantai diperlukan tangga yang dapat menghubungkan satu lantai ke lantai lain. Namun, dengan menggunakan tangga masih belum efisien. Oleh karena itu, diperlukan sebuah lift yang dapat digunakan untuk menghubungkan seluruh lantai di gedung. Untuk mengontrol sebuah elevator butuh motor dan sistem kontrol. Dalam penelitian ini, sebuah motor yang digunakan adalah bahwa Motor Stepper AT89S51. Motor Stepper digunakan untuk mengangkat ke atas dan ke bawah lift. Sementara Microcontroller digunakan untuk memberi sinyal data ke Stepper Motor sehingga dapat bergerak searah dengan jarum jam atau sebaliknya (mengangkat ke atas dan ke bawah lift). Untuk tahu bahwa harus ke bawah atau ke atas menggunakan tombol manual. Tombol ini ditempatkan

-
1. **Dosen Jurusan Teknik Informatika, STMIK Potensi Utama**
Jl. K.L. Yos Sudarso Km 6,5 Telp (061) 6640525
 2. **Dosen Jurusan Teknik Informatika, STMIK Potensi Utama**
Jl. K.L. Yos Sudarso Km 6,5 Telp (061) 6640525
 3. **Dosen Jurusan Sistem Informasi, STMIK Potensi Utama**
Jl. K.L. Yos Sudarso Km 6,5 Telp (061) 6640525

pada setiap lantai sehingga dapat mendeteksi Microcontroller menekan tombol untuk memutuskan lift yang dapat bergerak ke atas dan ke bawah.

Kata kunci: AT89S51, Mikrokontroler, Motor Stepper

PENDAHULUAN

Pada gedung - gedung besar yang memiliki lebih dari satu lantai diperlukan tangga yang dapat menghubungkan satu lantai ke lantai lainnya. Namun dengan menggunakan tangga masih kurang efisien. Karena itu dibutuhkan sebuah lift yang dapat menghubungkan seluruh lantai yang terdapat dalam gedung tersebut.

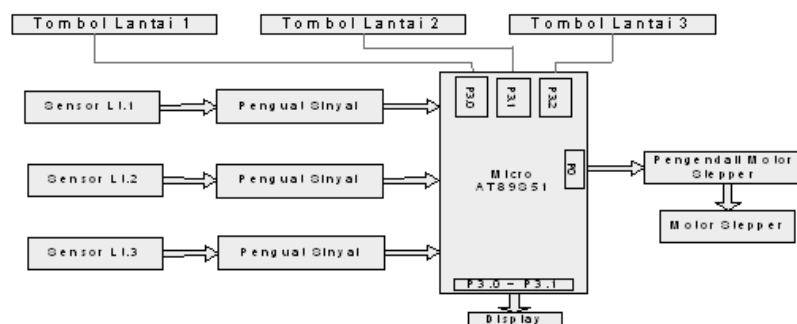
Untuk merancang sebuah lift diperlukan sebuah sistem kendali. Untuk itu dapat digunakan sebuah komputer untuk mengendalikannya. Namun dengan menggunakan komputer masih kurang efisien. Karena itu komputer dapat digantikan dengan sebuah mikrokontroler.

Mikrokontroler merupakan sebuah IC yang dapat ditulis dan dihapus sampai 1000 kali. Penggunaan mikrokontroler sebagai pengganti dari sebuah komputer untuk mengendalikan suatu sistem sudah sering digunakan. Karena itu penggunaan mikrokontroler akan sangat membantu dalam membuat sebuah sistem kendali. Dalam hal ini adalah mengendalikan sebuah lift.

RANCANGAN SISTEM

Diagram Blok

Secara garis besar rangkaian untuk lift 4 lantai terdiri dari 5 blok utama, yaitu rangkaian tombol, rangkaian display, rangkaian mikrokontroler, rangkaian *driver motor stepper* dan *motor stepper*. Diagram blok rangkaian tampak seperti gambar 1.



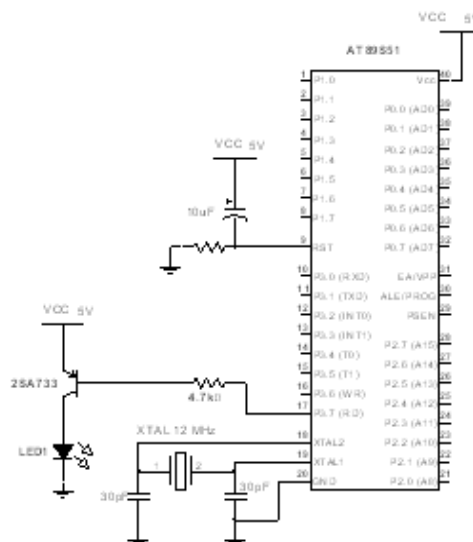
Gambar 1. Diagram Blok Rangkaian

Gambar di atas merupakan gambar diagram blok dari rangkaian untuk lift 3 lantai. Tombol berfungsi sebagai tombol pemanggil dari masing-masing lantai, dimana tombol

ini berada pada setiap lantai. Rangkaian mikrokontroler berfungsi untuk mengolah datayang diterima dari masing-masing tombol kemudian menampilkan pada display dan menggerakkan lift ke atas atau ke bawah. Rangkaian display berfungsi untuk menampilkan nilai dari penekanan tombol, semisal pengguna akan menuju lantai 3, sehingga menekan tombol 3, maka pada display akan tampil angka 3. *Driver motor* stepper berfungsi untuk mengendalikan pergerakan *motor stepper*, dan *motor stepper* sendiri berfungsi untuk menaikkan dan menurunkan lift.

Rangkaian Mikrokontroler AT89S51

Rangkaian Mikrokontroler berfungsi untuk mengolah sinyal yang dikirimkan oleh sensor 1 dan 2 kemudian menghitung kecepatan dari kendaraan dalam satuan kilometer per jam (Km/Jam) untuk ditampilkan ke *seven segmen*. Gambar rangkaian mikrokontroler AT89S51 ditunjukkan pada gambar 2.



Gambar 2. Rangkaian Mikrokontroler AT89S51

Mikrokontroler ini memiliki 32 port I/O, yaitu port 0, port 1, port 2 dan port 3. Pin 40 dihubungkan ke sumber tegangan 5 volt. Dan pin 20 dihubungkan ke *ground*. Rangkaian mikrokontroler ini menggunakan komponen kristal 12 MHz sebagai sumber *clocknya*. Nilai kristal ini akan mempengaruhi kecepatan mikrokontroler dalam mengeksekusi suatu perintah tertentu.

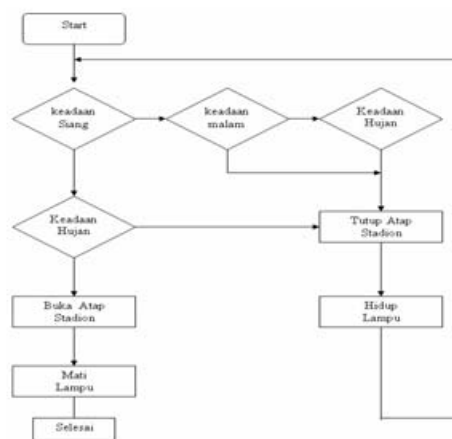
Pada pin 9 dihubungkan dengan sebuah kapasitor 10 uF yang dihubungkan ke positif dan sebuah resistor 10 Kohm yang dihubungkan ke *ground*. Kedua komponen ini berfungsi agar program pada mikrokontroler dijalankan beberapa saat setelah power aktif.

Lamanya waktu antara aktifnya power pada IC mikrokontroler dan aktifnya program adalah sebesar perkalian antara kapasitor dan resistor tersebut. Jika dihitung maka lama waktunya adalah :

(1)

Jadi 1 mili detik setelah power aktif pada IC kemudian program aktif. Pin 17 yang merupakan P3.7 dihubungkan dengan transistor dan sebuah LED. Ini dilakukan hanya untuk menguji apakah rangkaian minimum mikrokontroller AT89S51 sudah bekerja atau belum. Dengan memberikan program sederhana pada mikrokontroller tersebut, dapat diketahui apakah rangkaian minimum tersebut sudah bekerja dengan baik atau tidak. Jika LED yang terhubung ke Pin 17 sudah bekerja sesuai dengan perintah yang diberikan, maka rangkaian minimum tersebut telah siap digunakan.

Diagram Alir Pemrograman



Gambar 3. Diagram Alir Pemrograman

Program diawali dengan start, yang berarti bahwa rangkaian diaktifkan. Selanjutnya program akan mengecek kondisi hujan (sensor hujan), jika tidak ada sinyal dari sensor ini, maka program akan mengecek kondisi sensor malam (sensor malam), seterusnya program akan terus mengecek kondisi dari semua sensor. Sensor malam merupakan sensor pendeteksi siang hari.

Jika salah satu dari keadaan hujan, keadaan malam, keadaan siang mengirimkan sinyal ke mikrokontroler, maka program akan memerintahkan motor untuk berputar menutup atap stadion. Dan jika salah satu dari ke 3 sensor 1 mengirimkan sinyal, maka program akan memerintahkan motor untuk berputar membuka atap stadion. Selanjutnya program akan kembali ke rutin awal untuk kembali mengecek kondisi dari masing-masing sensor.

PENGUJIAN ALAT DAN PROGRAM

Pengujian Rangkaian Power Supplay (PSA)

Pengujian pada bagian rangkaian power supplay ini dapat dilakukan dengan mengukur tegangan keluaran dari rangkaian ini dengan menggunakan volt meter digital. Dari hasil pengujian diperoleh tegangan keluaran pertama sebesar + 5,1 volt. Tegangan ini dipergunakan untuk mensupplay tegangan ke seluruh rangkaian. Mikrokontroler AT89S51 dapat bekerja pada tegangan 4,0 sampai dengan 5,5 volt, sehingga tegangan 5,1 volt ini cukup untuk mensupplay tegangan ke mikrokontroler AT89S51. Dengan demikian rangkaian ini sudah dapat bekerja dengan baik. Sedangkan tegangan keluaran kedua adalah sebesar +12,3 volt.

Pengujian Rangkaian Mikrokontroler AT89S51

Untuk mengetahui apakah rangkaian mikrokontroler AT89S51 telah bekerja dengan baik, maka dilakukan pengujian. Pengujian bagian ini dilakukan dengan memberikan program sederhana pada mikrokontroler AT89S51. Programnya adalah sebagai berikut:

Loop:

```
Setb P3.7
Acall tunda
Clr P3.7
Acall tunda
Sjmp Loop
```

Tunda:

```
Mov r7,#255
```

```
Tnd: Mov r6,#255
      Djnz r6,$
      Djnz r7,tnd
      Ret
```

Program di atas bertujuan untuk menghidupkan LED yang terhubung ke P3.7 selama $\pm 0,13$ detik kemudian mematikannya selama $\pm 0,13$ detik secara terus menerus. Perintah Setb P3.7 akan menjadikan P3.7 berlogika high yang menyebabkan transistor aktif, sehingga LED menyala. Acall tunda akan menyebabkan LED ini hidup selama beberapa saat. Perintah Clr P3.7 akan menjadikan P3.7 berlogika low yang menyebabkan transistor tidak aktif sehingga LED akan mati. Perintah Acall tunda akan menyebabkan LED ini mati selama beberapa saat. Perintah Sjmp Loop akan menjadikan program tersebut berulang, sehingga akan tampak LED tersebut tampak berkedip.

Lamanya waktu tunda dapat dihitung dengan perhitungan sebagai berikut :

Kristal yang digunakan adalah kristal 12 MHz, sehingga 1 siklus mesin membutuhkan waktu = mikrodetik.

Pengujian Rangkaian Tombol

Pengujian pada rangkaian ini dapat dilakukan dengan menekan tombol, kemudian mengukur tegangan output dari rangkaian tersebut. Dari hasil pengujian didapatkan pada saat tombol tidak ditekan, maka output dari rangkaian ini adalah 4,9 volt. Ketika terjadi penekanan tombol, maka output dari rangkaian ini adalah 0 volt. Dengan demikian rangkaian ini telah berfungsi dengan baik.

Pengujian selanjutnya adalah dengan menghubungkan rangkaian tombol ini dengan rangkaian mikrokontroler yang telah diberi program sebagai berikut :

Jb P1.0,\$

Setb P3.7

.....

Program di atas akan menunggu adanya sinyal low yang dikirimkan tombol, dimana tombol tersebut dihubungkan dengan P1.0. Program akan terus menunggu sampai ada sinyal low yang dikirimkan oleh tombol. Jika ada sinyal low yang dikirimkan oleh tombol, maka program akan menghidupkan LED indikator yang dihubungkan ke P3.7.

Pengujian Rangkaian Display Seven Segment

Pengujian pada rangkaian ini dapat dilakukan dengan menghubungkan rangkaian ini dengan rangkaian mikrokontroler, kemudian memberikan data tertentu pada port serial dari mikrokontroler. *Seven segmen* yang digunakan adalah common anoda, dimana semen akan menyala jika diberi logika low (0) dan sebaliknya segmen akan mati jika diberi logika high (1).

Dari hasil pengujian diperoleh data yang harus dikirimkan ke port serial untuk menampilkan angka desimal adalah sebagai berikut:

Tabel 1 Penampilan Angka Decimal

<u>Angka</u>	<u>Data yang dikirim</u>
1	0ECH
2	18H
3	88H

Program yang diisikan pada mikrokontroler untuk menampilkan nilai-nilai tersebut adalah sebagai berikut:

```
bil1 equ 0echbil2 equ 18hbil3 equ 88h
```

```
Loop:
```

```
mov sbuf,#bil0
```

```
Jnb ti,$
```

```
Clr ti
sjmp loop
```

Program di atas akan menampilkan angka 0 pada semua *seven segmen*. Sedangkan untuk menampilkan 3 digit angka yang berbeda pada *seven segmen* adalah dengan mengirimkan ke 3 data angka yang akan ditampilkan pada seven segmen. Programnya adalah sebagai berikut :

Loop:

```
mov sbuf,#bil1
Jnb ti,$
Clr ti
mov sbuf,#bil2
Jnb ti,$
Clr ti
mov sbuf,#bil3
Jnb ti,$
Clr ti
sjmp loop
```

Program di atas akan menampilkan angka 1 pada seven segmen ketiga, angka 2 pada seven segmen kedua dan angka 3 pada seven segmen pertama.

Pengujian Rangkaian Driver *Motor Stepper*

Pengujian pada rangkaian jembatan H ini dilakukan dengan menghubungkan input rangkaian *driver motor stepper* ini dengan rangkaian mikrokontroler AT89S51 dan menghubungkan output dari rangkaian driver motor stepper ini dengan motor stepper, kemudian memberikan program sebagai berikut:

```
mov a,#11hputar:   mov P0,a   acall tunda   Rl a   jmp putar
```

Program diawali dengan memberikan nilai 11h pada pada accumulator (a), kemudian program akan memasuki rutin buka pintu. Nilai a diisikan ke port 0, sehingga sekarang nilai port 0 adalah 11h. ini berarti P0.0 dan P0.4 mendapatkan logika high sedangkan yang lainnya mendapatkan logika *low*, seperti table di bawah ini,

Tabel 2. Tabel Logika

	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
P0	0	0	0	1	0	0	0	1

Program dilanjutkan dengan memanggil rutin tunda. Lamanya tunda akan mempengaruhi kecepatan perputaran motor. Semakin lama maka tunda, maka perputaran motor akan semakin lambat. Perintah berikutnya adalah RI a, perintah ini akan memutar nilai yang ada pada accumulator (a), seperti tampak pada table di bawah ini :

Tabel 3. Tabel Nilai Akumulator

a	0	0	0	1	0	0	0	1
	← RI							
a	0	0	1	0	0	0	1	0
<u>Dst</u>								

Nilai pada accumulator (a) yang awalnya 11h, setelah mendapat perintah RI a, maka nilai pada accumulator (a) akan berubah menjadi 22h. Kemudian program akan melihat apakah kondisi sensor buka pintu dalam keadaan high (1) atau low (0). Jika high (1), Nilai yang ada pada accumulator (a), akan kembali diisikan ke port 0, maka nilai di port 0 akan berubah menjadi 22h, ini berarti P0.1 dan P0.5 mendapatkan logika high sedangkan yang lainnya mendapatkan logika low, seperti table di bawah ini,

Tabel 4. Tabel Logika

	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
P0	0	0	1	0	0	0	1	0

Sebelumnya telah dibahas bahwa P0.0, P0.1, P0.2, dan P0.3 dihubungkan ke masukan driver motor stepper, dengan program di atas maka P0.0, P0.1, P0.2, dan P0.3 akan mendapatkan nilai high (1) secara bergantian. Hal ini menyebabkan *motor stepper* akan berputar membuka pintu. Hal yang sama juga berlaku ketika motor berputar kearah sebaliknya, perbedaannya hanya pada perintah *rotate*. Jika pada perintah berlawanan arah jarum jam digunakan *rotate left* (RI), maka pada perintah searah jarum jam digunakan perintah *rotate right* (Rr). Perputaran perintah Rr diperlihatkan pada table berikut,

Tabel 4. Rotate

a	1	0	0	0	1	0	0	0
	→ Rr →							
a	0	1	0	0	0	1	0	0
<u>Dst</u>								

Pengujian Program

```

tombol1          bit    p3.5
tombol2          bit    p3.6
tombol3          bit    p3.7
tombol_turun     bit    p3.4
tombol_naik      bit    p3.3
sensor1          bit    p2.0
sensor2          bit    p2.1
sensor3          bit    p2.2
bil1    equ    0ech    ;untuk angka1
bil2    equ    18h     ;untuk angka2
bil3    equ    89h     ;untuk angka3

```

utama:

```

    mov p0,#0h
    mov a,#11h

```

; = = = tombol naik = = =;

naik:

```

    jb tombol_naik,turun
    sjmp run_naik

```

; = = = tombol turun = = =;

turun:

```

    jb tombol_turun,naik
    sjmp run_turun

```

; = = = routine lift naik = = =;

run_naik:

```

    call tunda
    call tunda
    call tunda
    call tunda
    call tunda
    call tunda
    call tunda

```

naik_lantai2:

```

    jb tombol2,naik_lantai3
    call lantai2

```

```
sjmp naik

naik_lantai3:
    jb tombol3,naik_lantai2
    call lantai3
    sjmp naik

lantai2:
    mov p0,a
    call delay_step
    rr a
    jnb sensor2,lantai2
    call angka2
    mov p0,#0h
    ret

lantai3:
    mov p0,a
    call delay_step
    rr a
    jnb sensor3,lantai3
    call angka3
    mov p0,#0h
    ret

; = = = routine lift turun = = = ;
run_turun:
    call tunda
    call tunda
    call tunda
    call tunda
    call tunda
    call tunda
    call tunda

turun_lantai1:
    jb tombol1,turun_lantai2
    call floor1
    ljmp turun

turun_lantai2:
```

```
        jnb tombol2,turun_lantai1
        call floor2
        ljmp turun
```

```
floor1:
        mov p0,a
        call delay_step
        rl a
        jnb sensor1,floor1
        call angka1
        mov p0,#0h
        ret
```

```
floor2:
        mov p0,a
        call delay_step
        rl a
        jnb sensor2,floor2
        call angka2
        mov p0,#0h
        ret
```

```
angka1:
        mov sbuf,#bil1
        jnb ti,$
        clr ti
        ret
```

```
angka2:
        mov sbuf,#bil2
        jnb ti,$
        clr ti
        ret
```

```
angka3:
        mov sbuf,#bil3
        jnb ti,$
        clr ti
        ret
```

; = = = routine tunda = = = ;

```
tunda:
    mov r7,#255
tnd:
    mov r6,#255
    djnz r6,$
    djnz r7,tnd
    ret
delay_step:
    mov r7,#50
dly:
    mov r6,#90
    djnz r6,$
    djnz r7,dly
    ret
end
```

KESIMPULAN

Dari hasil pelaksanaan perancangan alat hingga pengujian dan pembahasan sistem maka penulis dapat menarik kesimpulan, antara lain:

- a. Untuk mengangkat/menurunkan lift digunakan *motor stepper*, dimana motor stepper ini hanya bisa mengangkat beban yang ringan.
- b. Motor stepper tidak dapat dikendalikan langsung oleh mikrokontroler, sehingga dibutuhkan rangkaian *driver* sebagai sarana untuk mengendalikannya.
- c. Digunakan *motor stepper* pada alat ini karena pergerakan motor stepper dapat diatur presisi.

DAFTAR PUSTAKA

1. Agfianto (2004). *Belajar Mikrokontroler AT89C51/52/55 Teori dan Aplikasi*, Edisi Kedua, Penerbit: Gava Media, Yogyakarta
2. Agfianto (2002). *Teknik Antarmuka Komputer: Konsep dan Aplikasi*, Edisi Pertama, Penerbit: Graha Ilmu, Yogyakarta
3. Andi (2003). *Panduan Praktis Teknik Antarmuka dan Pemrograman Mikrokontroler AT89C51*, Penerbit PT Elex Media Komputindo, Jakarta

4. Andrian Yudhi, Haryanto Edy Viktor dkk (2009). *Perancangan Miniatur Tempat Parkir Mobil Dengan Menggunakan Sensor Dan Tampilan Berbasis Mikrokontroler AT89S51*, Computer Science Research and Its Development Center Journal, No.1 Vol 10, STMIK Potensi Utama
5. Andrian Yudhi, Puspasari Ratih dkk (2009). *Perancangan Pintu Otomatis Dengan Sistem Pembacaan Kode Bar Sebagai Kartu Identitas Berbasis Mikrokontroler AT89S51*, Computer Science Research and Its Development Center Journal, No.1 Vol 10, STMIK Potensi Utama
6. Haryanto Edy Viktor, Umami Khairul dkk (2009). *Perancangan Dan Implementasi Perangkat Lunak Untuk Mengendalikan Peralatan Elektronik Dan Listrik Notebook*, Computer Science Research and Its Development Center Journal, No.1 Vol 1, STMIK Potensi Utama
7. Malvino (2003). Albert Paul, *Prinsip-prinsip Elektronika*, Jilid 1 & 2, Edisi Pertama, Penerbit: Salemba Teknika, Jakarta