

**OTOMATISASI AREAL PARKIR pada PUSAT  
PERBELANJAAN BERBASIS MIKROKONTROLLER R8C TINY 13**

**SKRIPSI**

**Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik**

**UNIVERSITAS BRAWIJAYA**



Disusun Oleh :

**RATIH PERMANASARI  
NIM. 0210633070-63**

**DEPARTEMEN PENDIDIKAN NASIONAL  
JURUSAN TEKNIK ELEKTRO  
UNIVERSITAS BRAWIJAYA  
FAKULTAS TEKNIK  
MALANG  
2007**



**OTOMATISASI AREAL PARKIR pada PUSAT  
PERBELANJAAN BERBASIS MIKROKONTROLLER R8C TINY 13**

**SKRIPSI**

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik



Disusun oleh :

**RATIH PERMANASARI**  
**NIM. 0210633070-63**

Telah diperiksa dan disetujui oleh  
Dosen Pembimbing :

Adharul Muttaqin, ST, MT.  
NIP.132 311 886

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



**OTOMATISASI AREAL PARKIR pada PUSAT  
PERBELANJAAN BERBASIS MIKROKONTROLLER R8C TINY 13**

**SKRIPSI**

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik

Disusun oleh :

**RATIH PERMANASARI**  
**NIM. 0210633070-63**

Skripsi ini telah diuji dan dinyatakan lulus pada  
Tanggal 14 Juni 2007

Telah diperiksa dan disetujui oleh  
Dosen Penguji :

**Suprpto, ST, MT.**  
**NIP.132 149 320**

**Ir. Nanang Sulistiyanto**  
**NIP. 132 090 389**

**Ir. M. Julius. St., MS.**  
**NIP.131 124 655**

**Tibyani, ST, MT.**  
**NIP. 131 135 200**

**Ir. Heru Nurwasito, M.Kom.**  
**NIP. 131 879 033**



## PENGANTAR

Puji syukur atas kehadiran Allah SWT yang telah memberikan limpahan rahmat dan karuniaNya sehingga skripsi ini yang berjudul “ Otomatisasi Areal Parkir pada Pusat Perbelanjaan Berbasis Mikrokontroler R8C Tiny 13 ” dapat diselesaikan dengan baik.

Dalam perancangan dan pembuatan skripsi ini, banyak bimbingan, dukungan, bantuan, dan doa yang diberikan. Untuk itu ucapan terimakasih yang begitu besardisampaikan kepada :

- Ø Ir. Heru Nurwasito, M.Kom. selaku ketua Jurusan Teknik Elektro Universitas Brawijaya.
- Ø Rudy Yuwono, ST, MSc. selaku Sekretaris Jurusan Teknik Elektro.
- Ø Ir. Ponco Siwindarto selaku Ketua Kelompok Dasar Keahlian Elektronika.
- Ø Adharul Muttaqin, ST, MT. dan R. Arief setyawan, ST, MT. selaku dosen pembimbing yang banyak memberikan bantuan, kritik, dan saran sehingga dengan bimbingan Beliau, skripsi ini dapat diselesaikan dengan baik.
- Ø Para dosen Jurusan Teknik Elektro.
- Ø Seluruh karyawan Jurusan Teknik Elektro yang telah memberikan ilmu bermanfaat.
- Ø Kedua orang tua, Ir. Hari Subagyo dan Ellyta Furaningrum beserta keluarga yang selalu memberikan semangat dan doa.
- Ø Ibu Rukiana dan Niken Yunita Lestari, yang selalu memberikan bantuan, dukungan, dan doa.
- Ø Seluruh teman dan sahabat di fakultas Teknik Elektro yang tidak dapat disebutkan satu persatu.
- Ø Ir. Taufik Hamzah, Ibu Lely Setyani, dan seluruh karyawan CV. Sahabat Electric Madang yang memberikan dukungan sehingga skripsi ini dapat diselesaikan dengan baik.
- Ø Semua pihak yang berjasa atas bantuannya dalam penyelesaian skripsi ini.

Menyadari bahwa penulisan skripsi ini masih jauh dari sempurna, maka kritik dan saran sangat diperlukan guna perbaikan di masa yang akan datang.

Akhir kata, semoga penulisan skripsi ini banyak membawa manfaat dan pengetahuan bagi semua pihak.

Malang, Juni 2007

Penulis



## Ringkasan

Ratih Permanasari, Fakultas Teknik, Teknik Elektro Universitas Brawijaya Malang, Juni 2007, Judul : “ Otomatisasi Areal Parkir pada Pusat Perbelanjaan Berbasis Mikrokontroler R8C Tiny 13 “.

Kebutuhan sarana transportasi yang semakin meluas terlihat dari banyaknya penggunaan kendaraan bermotor. Pertambahan kendaraan bermotor tidak sebanding dengan pertambahan luas areal parkir. Di pusat perbelanjaan sering terjadi antrean panjang akibat pengaturan sistem parkir menggunakan tenaga manual. Alternatif solusi permasalahan tersebut adalah membuat suatu sistem otomatis pada areal parkir yang dilengkapi dengan palang pintu otomatis, penunjuk lantai parkir, dan penghitung jumlah mobil per hari. Tujuannya adalah untuk menghemat waktu bagi pengunjung agar tidak terjebak antrean panjang, disamping itu ketika seluruh lantai parkir penuh maka pintu tertutup otomatis. Penuh atau tidaknya areal parkir juga ditandai dengan tampilan LCD di luar gedung.

Skripsi ini merancang sebuah otomatisasi yang diwujudkan dalam prototipe areal parkir dua tingkat. Mikrokontroler R8C Tiny 13 sebagai pengontrol utama sistem melakukan proses Receive dan transmit data serial dengan mikrokontroler AT89S51. Kondisi awal, palang pintu utama dideteksi oleh sensor 1 hingga sensor 3 dan sensor 10 hingga sensor 12 dari 12 pasang sensor yang dihubungkan dengan AT89S51 A. Data dari sensor 2 dan 3 berfungsi untuk memicu proses pengetikan identitas mobil masuk, sehingga data tersebut diolah oleh mikrokontroler utama untuk memerintahkan AT89S51 membuka dan menutup palang pintu masuk. Proses pengetikan mobil keluar, proses cetak karcis, dan pergerakan palang pintu keluar dideteksi oleh sensor 10. Sensor 4 dan 5 mendeteksi jumlah mobil pada lantai satu, sensor 6 dan 7 mendeteksi jumlah mobil pada lantai dua, sedangkan sensor 8 dan 9 mendeteksi mobil keluar dari lantai dua. Kapasitas mobil pada lantai satu dan dua ditampilkan pada display berupa seven segment. Jika jumlah mobil = 20, display LCD pada lobi bertuliskan “ Parkir Penuh “, Jika tidak, maka bertuliskan “ Parkir Tersedia “.

**DAFTAR ISI**

	<b>Halaman</b>
<b>LEMBAR PERSETUJUAN</b> .....	iii
<b>PENGANTAR</b> .....	iv
<b>RINGKASAN</b> .....	vi
<b>DAFTAR ISI</b> .....	vii
<b>DAFTAR GAMBAR</b> .....	xii
<b>DAFTAR TABEL</b> .....	xiv
<b>BAB I PENDAHULUAN</b>	
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	3
1.3. Ruang Lingkup .....	4
1.4. Tujuan .....	4
1.5. Sistematika Kerja .....	4
<b>BAB II DASAR TEORI</b>	
2.1. Mikrokontroler R8C Tiny 13 .....	6
2.1.1. Arsitektur Renesas .....	6
2.1.2. Arsitektur Renesas .....	6
2.1.3. Arsitektur R8C Tiny 13 .....	14
2.1.3. Reset .....	14
2.1.3.1. <i>Hardware Reset</i> .....	14
2.1.3.2. <i>Software Reset</i> .....	15
2.1.3.3. <i>Watchdog Timer Reset</i> .....	15
2.1.4. Proteksi .....	16
2.1.5. Komunikasi Serial .....	16
2.1.5.1. <i>Clock Synchronous Serial I/O Mode</i> .....	16
2.1.5.2. <i>Clock Asynchronous Serial I/O Mode</i> .....	17
2.1.5.3. Bit Rate .....	18

2.2. Mikrokontroler AT89S51 .....	19
2.2.1. Arsitektur Dalam AT89S51 .....	20
2.2.2. Reset .....	28
2.2.3. Sistem Port (Masukan dan Keluaran) .....	29
2.2.3.1. Port 0 .....	30
2.2.3.2. Port 1 .....	30
2.2.3.3. Port 2 .....	30
2.2.3.4. Port 3 .....	30
2.2.4. Mode-mode Transmisi Data Serial .....	31
2.2.4.1. Mode 0 .....	31
2.2.4.2. Mode 1 .....	32
2.2.4.3. Mode 2 .....	33
2.2.4.4. Mode 3 .....	34
2.3. Transistor Sebagai Saklar .....	34
2.4. Sensor .....	36
2.4.1. LED <i>Infrared</i> .....	36
2.4.2. Fotodiode .....	37
2.5. Modul LCD M1632 .....	38
2.6. Papan Tombol .....	40
2.7. Peraga <i>Seven Segment</i> .....	40
2.8. <i>Driver Relay</i> .....	41
2.9. Enkoder .....	42
2.10. Rangkaian RTC .....	43
2.11. Rangkaian EEPROM .....	44

### **BAB III METODE PENELITIAN**

3.1. Studi Literatur .....	46
3.2. Penentuan Spesifikasi Alat .....	47
3.3. Perealisasian Alat .....	47
3.3.1. Perancangan Perangkat Keras dan Perealisasian Tiap Blok .....	47

3.3.2. Perancangan dan Perealisasian Perangkat Lunak .....	47
3.4. Pengujian Alat .....	48
3.4.1. Pengujian Perangkat Keras .....	48
3.4.2. Pengujian Perangkat Lunak .....	48
3.4.3. Pengujian Keseluruhan Sistem .....	48
3.5. Analisa Data dan Pengambilan Kesimpulan .....	48

## **BAB IV PERANCANGAN ALAT**

4.1. Penentuan Spesifikasi Alat .....	49
4.2. Pembuatan Alat .....	50
4.3. Prinsip Kerja Alat .....	52
4.4. Perancangan Perangkat Keras .....	54
4.4.1. Rangkaian Kontrol Menggunakan AT89S51 .....	54
4.4.1.1. Pengaturan <i>Baud Rate</i> .....	54
4.4.1.2. Rangkaian Reset .....	55
4.4.2. Perancangan Karakter Huruf pada LCD .....	56
4.4.3. Rangkaian Tombol Masukan ( <i>Keypad</i> ) .....	59
4.4.4. Rangkaian <i>Driver Relay</i> .....	60
4.4.5. Rangkaian Sensor Inframerah .....	62
4.4.5.1. Rangkaian Pemancar .....	62
4.4.5.2. Rangkaian Penerima .....	63
4.4.6. Rangkaian RTC .....	67
4.4.7. Rangkaian EEPROM .....	68
4.5. Flowchart Keseluruhan Sistem .....	69

## **BAB V PENGUJIAN DAN ANALISIS**

5.1. Pengujian Rangkaian Driver Relay .....	72
5.1.1. Tujuan .....	72
5.1.2. Peralatan yang Digunakan .....	72
5.1.3. Prosedur Pengujian .....	73
5.1.4. Hasil Pengujian dan Analisis .....	73

5.2. Pengujian Rangkaian Sensor Mobil .....	73
5.2.1. Tujuan .....	73
5.2.2. Peralatan yang Digunakan .....	73
5.2.3. Prosedur Pengujian .....	74
5.2.4. Hasil Pengujian dan Analisis .....	74
5.3. Pengujian Tampilan LCD .....	74
5.3.1. Tujuan .....	74
5.3.2. Peralatan yang Digunakan .....	75
5.3.3. Prosedur Pengujian .....	75
5.3.4. Hasil Pengujian dan Analisis .....	75
5.4. Pengujian Unit Masukan Keypad .....	75
5.4.1. Tujuan .....	75
5.4.2. Peralatan yang Digunakan .....	76
5.4.3. Prosedur Pengujian .....	76
5.4.4. Hasil Pengujian dan Analisis .....	76
5.5. Pengujian RTC .....	77
5.5.1. Tujuan .....	77
5.5.2. Peralatan yang Digunakan .....	77
5.5.3. Prosedur Pengujian .....	77
5.5.4. Hasil Pengujian dan Analisis .....	77
5.6. Pengujian Rangkaian Motor DC .....	78
5.6.1. Tujuan .....	78
5.6.2. Peralatan yang Digunakan .....	77
5.6.3. Prosedur Pengujian .....	77
5.6.4. Hasil Pengujian dan Analisis .....	79
5.7. Pengujian Sistem Keseluruhan .....	79
5.7.1. Tujuan .....	79
5.7.2. Peralatan yang Digunakan .....	79
5.7.3. Prosedur Pengujian .....	80
5.7.4. Hasil Pengujian dan Analisis .....	80
5.7.4.1. Saat Mobil Masuk .....	80



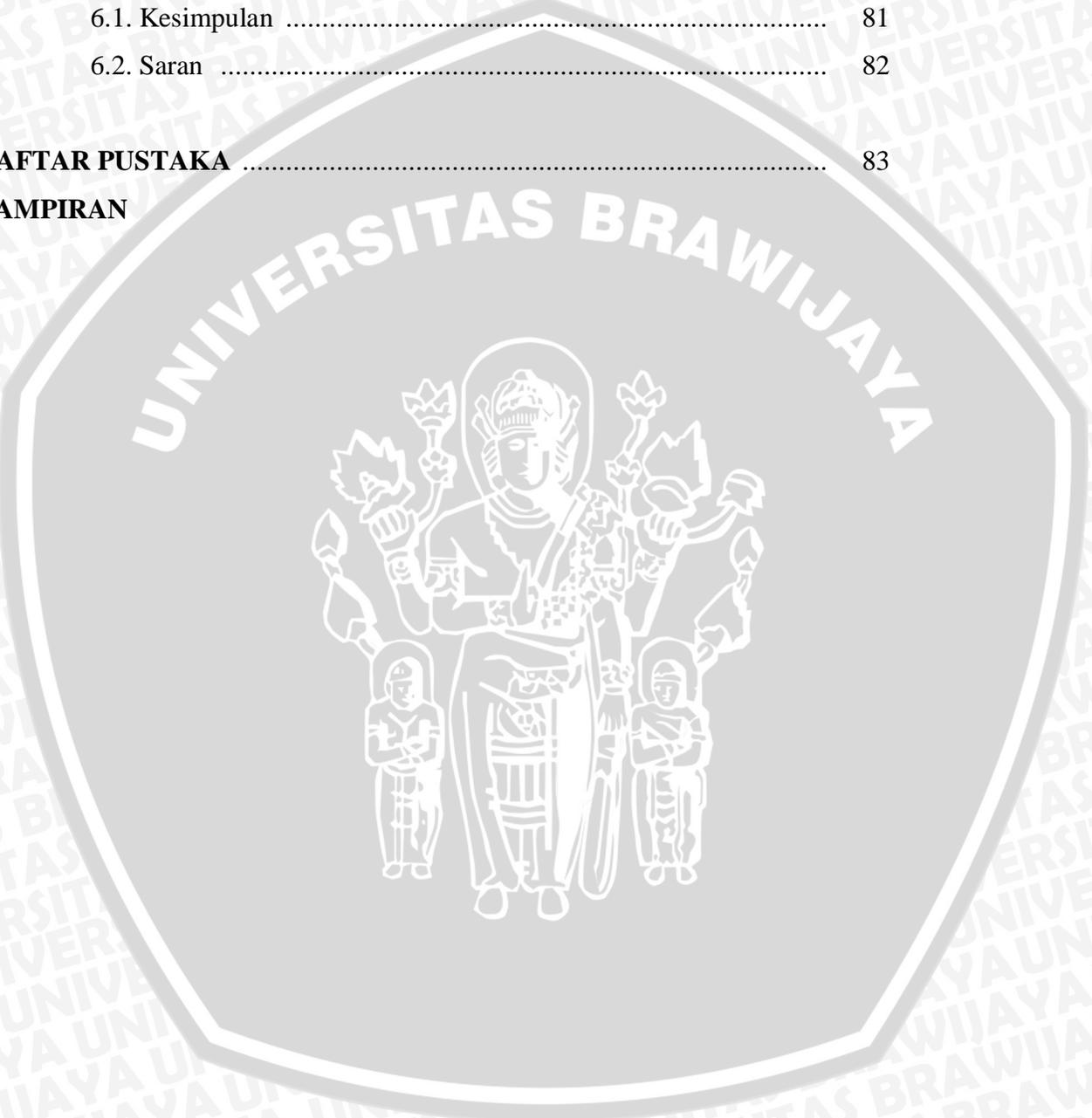
5.7.4.2. Saat Mobil Keluar ..... 80  
5.7.4.3. Saat Parkir Penuh ..... 80

**BAB VI PENUTUP**

6.1. Kesimpulan ..... 81  
6.2. Saran ..... 82

**DAFTAR PUSTAKA** ..... 83

**LAMPIRAN**



## DAFTAR GAMBAR

	<b>Halaman</b>
<b>Gambar 2.1.</b> Blok Diagram R8C Tiny 13 .....	9
<b>Gambar 2.2.</b> Konfigurasi Pin R8C Tiny 13 .....	10
<b>Gambar 2.3.</b> Konfigurasi Pin AT89S51 .....	20
<b>Gambar 2.4.</b> Arsitektur Dalam AT89S51 .....	21
<b>Gambar 2.5.</b> Pemetaan Memori Internal AT89S51 .....	22
<b>Gambar 2.6.</b> Definisi Fungsi dan Bit PSW .....	23
<b>Gambar 2.7.</b> Diagram Register (a) TCON dan (b) TMOD .....	24
<b>Gambar 2.8.</b> Definisi Fungsi dan Bit TMOD .....	24
<b>Gambar 2.9.</b> Definisi dan Fungsi TCON .....	25
<b>Gambar 2.10.</b> Diagram Register SCON .....	25
<b>Gambar 2.11.</b> Definisi dan Fungsi SCON .....	26
<b>Gambar 2.12.</b> Definisi dan Fungsi IE .....	26
<b>Gambar 2.13.</b> Definisi dan Fungsi IP .....	27
<b>Gambar 2.14.</b> Definisi dan Fungsi PCON .....	27
<b>Gambar 2.15.</b> (a). Rangkaian n Reset, (b) Rangkaian Ekvivalen Reset .....	28
<b>Gambar 2.16.</b> Format Word Data dalam Mode 1 .....	32
<b>Gambar 2.17.</b> Format Word Data dalam Mode 2 .....	33
<b>Gambar 2.18.</b> Rangkaian Transistor Sebagai Saklar .....	35
<b>Gambar 2.19.</b> Titik Jenuh dan Titik Putus pada Garis Beban DC .....	35
<b>Gambar 2.20.</b> Lambang Skematis dan Contoh Rangkaian dari LED .....	37
<b>Gambar 2.21.</b> Fotodiode .....	38
<b>Gambar 2.22.</b> LCD .....	39
<b>Gambar 2.23.</b> Diagram Blok Modul LCD M1632 .....	39
<b>Gambar 2.24.</b> Rangkaian Keypad 4x4 .....	40
<b>Gambar 2.25.</b> Penampil Seven Segmen .....	40
<b>Gambar 2.26.</b> Rangkaian Driver Relay .....	41
<b>Gambar 2.27.</b> Bagan IC LS74164 .....	42
<b>Gambar 2.28.</b> Arsitektur Dalam IC LS74164 .....	42

<b>Gambar 2.29.</b> Konfigurasi RTC DS1307 .....	43
<b>Gambar 2.30.</b> Arsitektur Dalam RTC DS1307 .....	44
<b>Gambar 2.31.</b> Arsitektur Dalam EEPROM 24C01A .....	44
<b>Gambar 2.32.</b> Pin EEPROM 24C01A .....	45
<b>Gambar 4.1</b> Blok Diagram Sistem .....	51
<b>Gambar 4.2.a)</b> Rangkaian Osilator AT89S51 .....	54
<b>Gambar 4.2.b)</b> Rangkaian Osilator R8C Tiny 13 .....	55
<b>Gambar 4.3.</b> Antarmuka LCD dengan Mikrokontroler AT89S51 .....	57
<b>Gambar 4.4.</b> Antarmuka LCD dengan Mikrokontroler R8C Tiny 13 .....	58
<b>Gambar 4.5.</b> Antarmuka Keypad dengan Mikrokontroler AT89S51 .....	60
<b>Gambar 4.6.</b> Rangkaian Driver Relay .....	61
<b>Gambar 4.7.</b> Rangkaian Pemancar Infrared .....	63
<b>Gambar 4.8.</b> Rangkaian Penerima Infrared .....	64
<b>Gambar 4.9.</b> Konfigurasi Pin RTC 1307 .....	67
<b>Gambar 4.10.</b> Antarmuka EEPROM dengan Mikrokontroler AT89S51 .....	68
<b>Gambar 5.1.</b> Rangkaian Pengujian Driver Relay .....	73
<b>Gambar 5.2.</b> Blok Diagram Pengujian Sensor Mobil .....	74
<b>Gambar 5.3.1.</b> Rangkaian Pengujian LCD dengan Sistem MK AT89S51 ....	75
<b>Gambar 5.3.2.</b> Rangkaian Pengujian LCD dengan Sistem MK R8C Tiny 13	75
<b>Gambar 5.4</b> Rangkaian Pengujian Unit Masukan Keypad .....	76
<b>Gambar 5.5.</b> Rangkaian Pengujian RTC .....	77
<b>Gambar 5.6.</b> Pengujian Motor DC .....	78
<b>Gambar 5.7</b> Pengujian Sistem Keseluruhan .....	79

## DAFTAR TABEL

	<b>Halaman</b>
<b>Tabel 2.1.</b> Karakteristik R8C Tiny 13 .....	8
<b>Tabel 2.2.</b> Kapasitas Memori dan Tipe Famili Mikrokontroler Renesas ...	9
<b>Tabel 2.3.</b> Deskripsi Pin R8C Tiny 13 .....	12
<b>Tabel 2.4.</b> Organisasi Memori R8C Tiny 13 .....	13
<b>Tabel 2.5.</b> Status Pin Saat Level Pin RESET adalah “L” .....	14
<b>Tabel 2.6.</b> Spesifikasi Mode UART Clock Synchronous .....	17
<b>Tabel 2.7.</b> Spesifikasi Mode UART Clock Asynchronous .....	18
<b>Tabel 2.8.</b> Contoh Bit Rate Setting .....	19
<b>Tabel 2.9.</b> Kondisi SFR Reset .....	28
<b>Tabel 2.10.</b> Fungsi Alternatif Port 3 .....	31
<b>Tabel 4.1.</b> Fungsi Pin LCD .....	58
<b>Tabel 5.1.</b> Hasil Pengujian Rangkaian Driver Relay .....	73
<b>Tabel 5.2.</b> Hasil Pengujian Fotodioda .....	74
<b>Tabel 5.3.</b> Hasil Pengujian Keypad Disimulasikan dengan Nyala LED ....	76
<b>Tabel 5.4.</b> Perbandingan Waktu RTC dengan Stopwatch .....	77
<b>Tabel 5.5.</b> Hasil Pengujian Motor DC .....	78

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Perkembangan ilmu pengetahuan dan teknologi di bidang elektronika membawa perubahan besar, hal tersebut dapat dilihat dari tersedianya berbagai macam alat atau sarana yang dapat mempermudah aktifitas manusia. Banyak sistem otomatis yang dibuat sehingga memudahkan manusia untuk beraktifitas. Hal tersebut termasuk pemenuhan kebutuhan masyarakat dalam bidang transportasi baik dalam pemenuhan sarana dan prasarana.

Pertambahan kendaraan bermotor semakin banyak seiring dengan pertambahan jumlah penduduk. Namun hal tersebut tidak sebanding dengan pertambahan luas areal parkir. Oleh karena itu diperlukan suatu sistem otomatis pada areal parkir, dalam hal ini dibahas mengenai areal parkir pada pusat perbelanjaan, yaitu tempat kebanyakan orang menghabiskan waktu baik untuk berbelanja maupun berkunjung saja.

Seringkali pengguna jasa parkir pada pusat perbelanjaan merasa kecewa terhadap pelayanan pusat perbelanjaan karena dianggap kurang efisien dan praktis dalam memberi informasi mengenai kapasitas lantai parkir. Maka dari itu sistem parkir otomatis areal parkir menarik untuk dibahas. Hal tersebut bukan tanpa kendala, kendala di atas pernah dibahas dalam skripsi sebelumnya, yaitu mengenai sistem parkir mobil terkendali PC dalam gedung bertingkat oleh Deddy Tjahjono tahun 2002. Sistem tersebut memiliki kelemahan yaitu kurang praktis dan efisien karena *interfacing* antara sensor lokasi parkir di gedung bertingkat dengan PC secara paralel menggunakan PPI, sehingga pengkabelannya sangat rumit, dan tidak dijelaskan bagaimana kondisi bila areal parkir sudah penuh.

Untuk mengatasi kendala di atas, pada skripsi sebelumnya oleh Rakhmat Ibnu tahun 2004, dirancang suatu simulasi pengontrol sistem parkir mobil dalam gedung bertingkat menggunakan PC dan mikrokontroler AT89C51. Sistem memang lebih baik karena sistem komunikasi data dari sensor di lokasi parkir mobil ke PC dirangkai secara serial.

Sehingga dalam pengkabelan akan lebih sederhana dan jumlah sensor pada lokasi parkir bisa seperti kapasitas sesungguhnya. Namun skripsi tersebut memiliki kelemahan yaitu :

1. Tidak dijelaskan bagaimana otomatisasi sistem parkir bila areal parkir penuh.
2. Tidak dijelaskan bagaimana otomatisasi informasi pada pengguna jasa areal parkir bahwa di lantai mana ada parkir kosong.

Penelitian ini merupakan pengembangan dari kedua judul di atas dan memiliki kelebihan yakni :

1. Menggunakan Mikrokontroler R8C Tiny 13 sebagai pengontrol utama pada areal parkir yang merupakan mikrokontroler keluaran terbaru dengan harga terjangkau yang memiliki kapasitas memori yang lebih besar serta transfer data yang lebih cepat dibanding MCS51, data masih dapat tersimpan bila terjadi lampu mati.
2. Pada saat masuk areal parkir, pengguna jasa parkir bebas masuk areal parkir dengan sebelumnya berbekal informasi yang terpampang pada *seven-segment* mengenai lantai dan kapasitas sisa ruang tiap lantai. Mobil akan terhitung masuk pada lantai tertentu bila mobil berhenti pada satu ruang parkir sehingga dapat langsung mempengaruhi tampilan *seven-segment*.
3. Menggunakan sistem counter untuk mengetahui jumlah mobil dalam areal parkir pada tiap lantai parkir.
4. Sistem otomatis bekerja menginformasikan pengunjung bila areal parkir telah penuh dengan penutupan otomatis palang pintu utama dibarengi tampilan LCD pada lobi depan bertuliskan "Parkir Penuh".
5. Ketika *entry* data, pada simulasinya cukup mengetikkan nomor urutan pengunjung pada waktu masuk dan keluar areal parkir pada saat keluar areal parkir.

Berdasar hal tersebut tercetus ide untuk membuat suatu sistem otomatis pada areal parkir otomatis yang dilengkapi dengan palang pintu otomatis, penunjuk lantai parkir, dan penghitung jumlah mobil perhari.

Tujuan hal tersebut untuk menghemat waktu bagi pengunjung agar tidak terjebak pada antrean panjang, disamping itu ketika seluruh lantai parkir penuh maka palang pintu tertutup secara otomatis tanpa bantuan manual lagi, hal tersebut juga ditandai dengan tampilan pada LCD di luar gedung, dan untuk memudahkan pengunjung memilih tingkat lantai parkir yang mereka inginkan sesuai dengan kapasitas yang masih ada. Untuk pihak Mall, sistem ini juga amat bermanfaat untuk mengevaluasi dan mengkalkulasi seluruh pengunjung yang datang per bulan bahkan pertahun sehingga dapat meningkatkan kualitas pelayanannya.

Sistem otomatis ini dirancang dengan menggunakan sensor *infrared led* dan *fotodiode* serta beberapa komponen pendukung lainnya seperti mikrokontroler R8C Tiny 13 sebagai pengontrol utama sistem dan penggunaan MCS-51 sebagai port tambahan yang diinterfacekan secara serial terhadap mikrokontroler utama. *Seven Segmen* digunakan untuk menampilkan lantai serta kapasitas sisa ruang areal parkir pada tiap lantai. LCD digunakan untuk menampilkan informasi bahwa areal parkir tersebut penuh atau tidak, dan motor dc digunakan untuk menggerakkan palang pintu.

## 1.2 Rumusan Masalah

Mengacu pada permasalahan yang diuraikan pada latar belakang, maka rumusan masalah dapat ditekankan pada:

1. Bagaimana merancang sistem otomatis areal parkir yang dilengkapi dengan palang pintu otomatis, penunjuk lantai parkir dan penghitung jumlah mobil perhari. Dengan adanya alat ini maka diharapkan dapat membantu pusat perbelanjaan meningkatkan kualitas pelayanan pada jasa pengguna parkir.
2. Bagaimana merancang program mikrokontroler R8C Tiny 13 dengan menggunakan bahasa C sehingga mikrokontroler dapat digunakan sebagai pengontrol pada alat ini.
3. Bagaimana merancang rangkaian *driver* motor DC untuk menggerakkan palang pintu pada prototype areal parkir.
4. Bagaimana merancang rangkaian sensor pada tiap lantai untuk memicu counter sebagai penghitung mobil dalam ruang parkir.
5. Bagaimana merancang rangkaian *driver* untuk menampilkan angka pada LCD.

6. Bagaimana merancang rangkaian *driver* untuk menampilkan angka pada *seven segment*.

### 1.3 Batasan Masalah

Dalam ruang lingkup pembahasan ini, permasalahan yang akan dibahas meliputi :

1. Parameter keberhasilan alat adalah pada kemampuan sensor mendeteksi mobil yang memicu palang pintu terbuka dan penunjukan angka pada *seven segment*.
2. Pembahasan *LED infrared* dan *fotodiode* sebagai sensor pada tiap lantai.
3. Pembahasan rangkaian mikrokontroler R8C Tiny 13 menggunakan Bahasa C.
4. Prototype ini dibuat hanya mampu melakukan perhitungan maksimal 20 mobil pada dua lantai.
5. Gedung dibatasi dua tingkat.
6. Mekanisme pembayaran tarif parkir masih dengan batuan operator.
7. Alat yang dibuat adalah miniatur areal parkir.

### 1.4 Tujuan

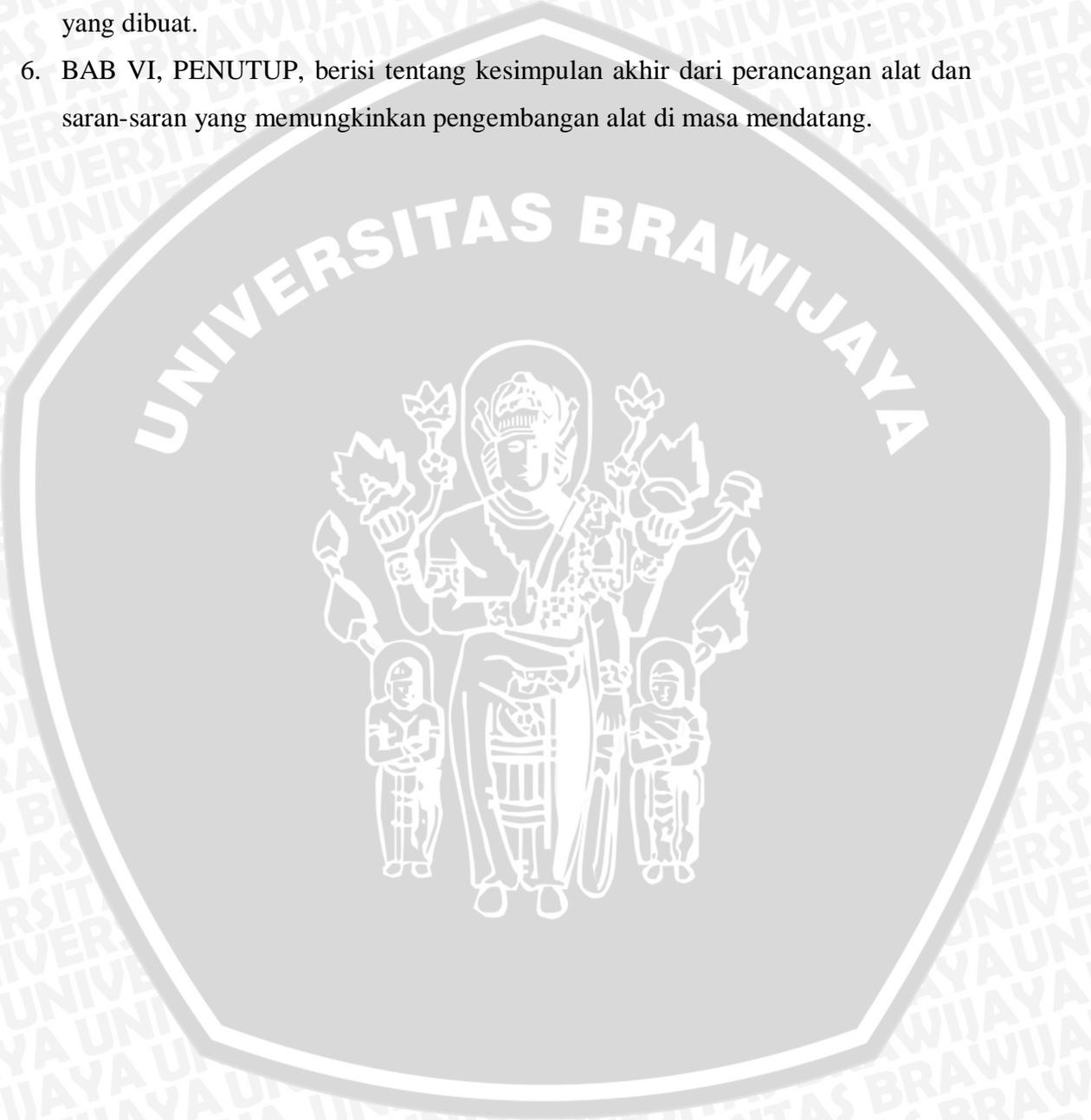
Tujuan perancangan dan pembuatan alat ini adalah untuk merancang sistem otomatis areal parkir yang dilengkapi dengan palang pintu otomatis, penunjuk lantai parkir, dan penghitung jumlah mobil perhari yang diharapkan dapat membantu pusat perbelanjaan meningkatkan kualitas pelayanan pada jasa pengguna parkir.

### 1.5 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penyusunan laporan tugas akhir ini adalah sebagai berikut:

1. BAB I, PENDAHULUAN, menjelaskan tentang latar belakang, rumusan masalah, ruang lingkup masalah, tujuan penulisan, dan sistematika penulisan.
2. BAB II, TINJAUAN PUSTAKA, Membahas teori-teori dasar penunjang perancangan dan pembuatan alat.
3. BAB III, METODOLOGI PENELITIAN, menjelaskan tahap demi tahap penelitian skripsi.

4. BAB IV, PERANCANGAN DAN PEMBUATAN ALAT, Membahas prinsip kerja, perancangan dan pembuatan alat baik perangkat keras maupun perangkat lunak yang akan direalisasikan.
5. BAB V, PENGUJIAN DAN ANALISIS, Memuat hasil pengujian dan analisa alat yang dibuat.
6. BAB VI, PENUTUP, berisi tentang kesimpulan akhir dari perancangan alat dan saran-saran yang memungkinkan pengembangan alat di masa mendatang.



## BAB II

### DASAR TEORI

Miniatur sistem parkir otomatis yang akan dirancang memerlukan komponen elektronika untuk menjalankan otomatisasinya. Komponen tersebut meliputi mikrokontroller Renesas seri R8C Tiny 13 sebagai pengontrol utama sistem, dan komponen-komponen pendukung lainnya seperti Mikrokontroller AT89S51, LED *Infrared* dan *photodiode*, motor DC, *seven segment*, *driver relay*, LCD, RTC, dan EEPROM yang akan dijelaskan secara rinci pada sub bab-sub bab di bawah ini.

#### 2.1 Mikrokotoller R8C Tiny 13

##### 2.1.1 Arsitektur Renesas

Mikrokontroller merupakan sebuah komponen elektronika yang bisa diprogram (*Programmable*) dan bekerja secara mandiri sesuai dengan program yang diberikan dalam memorinya. Mikrokontroller seperti layaknya komputer yang memiliki bagian-bagian seperti ROM, RAM, EEPROM, port paralel, port serial, bahkan ADC. Oleh karena itu mikrokontroller bisa dianggap sebagai komputer mikro.

##### 2.1.2 Arsitektur R8C Tiny 13

Mikrokotoller R8C Tiny 13 merupakan produksi Renesas, famili dari M16C atau M16C PLATFORM. R8C merupakan nama CPU, sedang R8C/Tiny=R8C/11,12,13,dst, yang penjabarannya adalah sebagai berikut :

R = Renesas, keluaran dari Renesas Technology corporation

8 = Internal Bus Width (CPU = 16 bit)

C = Compact

Tiny = Jumlah pin dan *Flash Memory* MCU yang dikemas sangat kecil dan memiliki banyak keunggulan dan fasilitas sebagai berikut :

- Ø Port Paralel 3 buah.
- Ø Port serial 2 buah, salah satu khusus UART.
- Ø ADC 8 bit dan 10 bit 12 kanal.

- Ø Waktu konversi ADC hanya 3 $\mu$ s, yang berarti semakin cepat konversi, maka semakin cepat pemrosesan data.
- Ø 16 bit CISC CPU, kecepatan transfer data tinggi.
- Ø RAM sebesar 768 byte sehingga mampu menyimpan data dengan kapasitas yang besar.
- Ø ROM sebesar 12 KByte lebih hemat 20 % dalam penggunaannya.
- Ø Operasi kecepatan tinggi (Maksimum operasi frekuensi pada 20 MHz), *Low power dissipation operation*, Efisiensi ROM tinggi.
- Ø Harga terjangkau untuk Flash MCU.
- Ø Informasi dan datasheet lengkap mudah didapat.
- Ø Harga R8C tiny yang terjangkau.
- Ø Menghemat penggunaan komponen eksternal, karena R8C tiny 13 telah dikemas dengan *on-chip debugger*, *high speed ring oscillator*, *data flash*, POR, LVD sebagai pengganti penggunaan eksternal dari reset IC, EEPROM, dan Oscillator.

Mikrokontroler R8C Tiny 13 memiliki karakteristik yang berbeda dengan mikrokontroler lainnya. Karakteristik mikrokontroler R8C Tiny 13 dapat dilihat dalam Tabel 2.1.

**Tabel 2.1.** Karakteristik R8C tiny 13

Item		Performance
CPU	Number of basic instruction	89 Instruction
	Shortest instruction execution time	50ns (f(XIN)=20MHz, Vcc=3.0 to 5.0) 100ns (f(XIN)=10MHz, Vcc=2.7 to 5.5)
	Operating Mode	Single chip
	Address space	1M bytes
	Memory Capacity	See Table 1.2
Peripheral Function	Interrupt	Internal : 11 faktors, External : 5 faktors, Software : 4 Faktors, Priority level : 7 levels
	Watchdog Timer	15 bit x 1 (with prescaler)
	Timer	Reset start with function selectable
		Timer X : 8 bit x 1 channel, Timer Y : 8 bit x 1 channel, Timer Z : 8 bit x 1 channel (each timer equipped with 8-bit prescaler)
	Serial Interface	Timer C : 16 bit x 1 channel Circuit of input capture and output compare
	A/D Converter	Ø 1 Channel Clock synchronous, UART
	Clock generation circuit	Ø 1 Channel UART
	Oscillation stop detection function	10-bit A/D converter : 1 circuit, 12 Channels
	Voltage Detection Circuit	2 Circuits
	Power on reset circuit	2 Circuits
	Port	<ul style="list-style-type: none"> <li>Main clock generation circuit (Equipped with an a built-in feedback resistor).</li> <li>On-chip Oscillator(high speed, Low speed). On high cheap Oscillator the frequency adjustment function is usable.</li> </ul>
		Stop Detection of main clock oscillation
	Included	
Included		
Input/Output : 22 (Including LED drive port), Input : 2 (LED Drive I/O port : 8)		
Electrical Characteristic	Power Supply Voltage	Vcc = 3.0 to 5.5V (f(XIN)=20MHz) Vcc = 2.7 to 5.5V (f(XIN)=10MHz)
	Power Consumption	Typ.9mA (Vcc=5.0V, (f(XIN)=20MHz, High-Speed Mode) Typ.5mA (Vcc=3.0V, (f(XIN)=10MHz, High-Speed Mode) Typ.35µA (Vcc= 3.0V, Wait Mode,

		Peripheral clock stops) Typ.0.7 $\mu$ A (Vcc=3.0V, Stop Mode)
Flash Memory	Program/Erase voltage	Vcc = 2.7 to 5.5V
	Number of program/Erase	10.000 times (Data Area) 1.000 times (Program Area)
Operating ambient temperature		-20 to 85°C -40 to 85°C (D-Version)
Package		32-pin plastick mold LQFP

Sumber : Datasheet Renesas

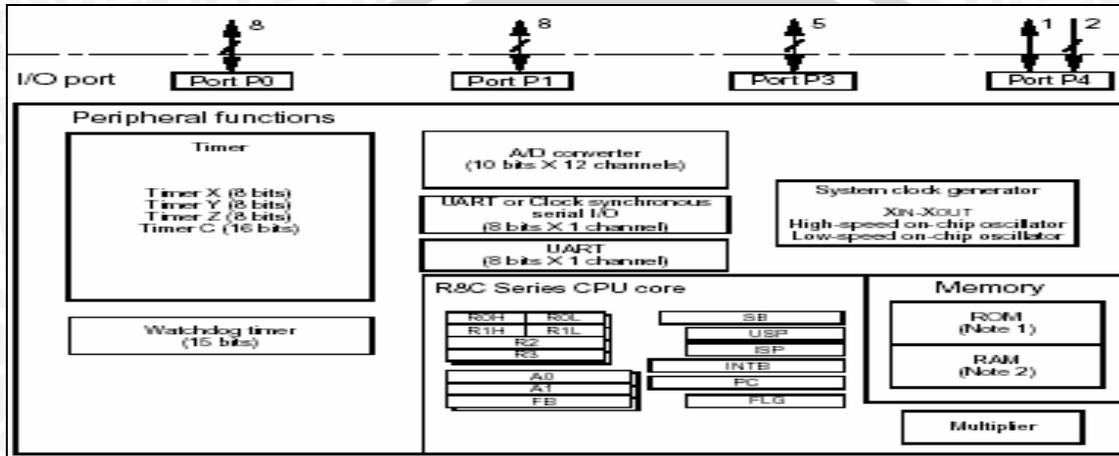
Mikrokontroller R8C Tiny 13 juga memiliki kapasitas memori dan dikemas dalam tipe tertentu yakni tipe nomor R5F21134DFP, dimana mikrokontroller ini juga memiliki famili dengan kapasitas memori dan tipe berbeda. Penjelasan mengenai kapasitas memori dan tipe famili mikrokontroller Renesas dapat dilihat dalam Tabel 2.2.

Type No.	ROM Capacity		RAM Capacity	Package Type	Remarks
	Program Area	Data Area			
RsF21132FP	8Kbytes	2K bytesx2	512 bytes	PLQP0032GB-A	Flash Memory Program
RsF21133FP	12Kbytes	2K bytesx2	768 bytes	PLQP0032GB-A	
RsF21134FP	16Kbytes	2K bytesx2	1K bytes	PLQP0032GB-A	
RsF21132DFP	8Kbytes	2K bytesx2	512 bytes	PLQP0032GB-A	
RsF21133DFP	12Kbytes	2K bytesx2	768 bytes	PLQP0032GB-A	
<b>RsF21134DFP</b>	<b>16Kbytes</b>	<b>2K bytesx2</b>	<b>1K bytes</b>	<b>PLQP0032GB-A</b>	

Sumber : Datasheet Renesas

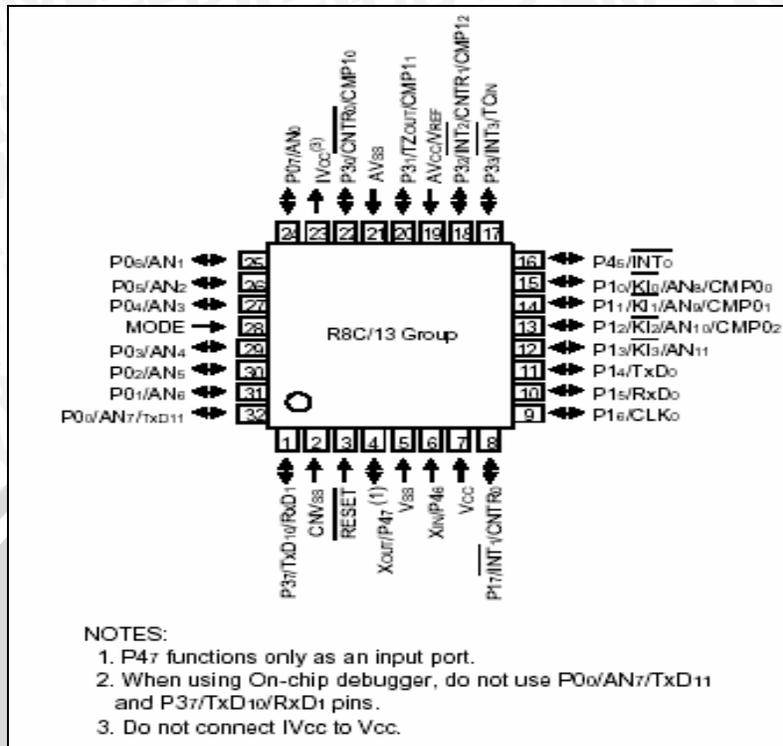
Dari tabel di atas dapat diketahui pendeskripsian tipe dari mikrokontroler R8C tiny 13 beserta seri lain dari mikrokontroler Renesas dengan ROM Capacity yang berbeda dengan grup yang sama.

Mikrokontroler R8C Tiny 13 memiliki struktur yang cukup kompleks. Blok diagram mikrokontroler tersebut dapat dilihat dalam Gambar 2.1.



**Gambar 2.1.**  
Blok Diagram R8C Tiny 13

Mikrokontroler R8C Tiny 13 memiliki 22 I/O port dan 2 port input. Pendeskripsian pin-pin mikrokontroler dapat dilihat dalam Gambar 2.2



**Gambar 2.2.**  
Konfigurasi Pin R8C Tiny 13

**Deskripsi kaki-kaki pin R8C Tiny 13**

- Ø Vcc - dihubungkan dengan tegangan DC 2,7 V sampai 5,5 V.
- Ø Vss – dihubungkan dengan ground.
- Ø IVcc – dipakai untuk menstabilkan *power suplay* internal. Pin ini dihubungkan dengan Vss melalui kapasitor 0,1 µF. Pin tersebut tidak boleh dihubungkan dengan Vcc.
- Ø AVcc dan AVss – merupakan pin power suplay bagi ADC. AVcc dihubungkan dengan Vcc dan pin AVss dengan Vss. Antara AVss dan AVcc dihubungkan dengan kapasitor.
- Ø Reset – Apabila diberi logika 0 akan mereset CPU.
- Ø CNVss – dipakai untuk mengurangi noise pada pin. Pin dihubungkan dengan resistor antara CNVss dan Vss dengan resistor sebesar 5 kΩ. Dan resistor akan sedekat mungkin dengan pin CNVss.

- Ø MODE – dipakai mengubah keadaan antara program mode yang dipakai untuk memasukkan program dan run mode yang dipakai untuk menjalankan program dalam CPU. Untuk keadaan run mode, pin ini harus dihubungkan dengan Vcc melalui resistor.
- Ø Xin dan Xout – pin ini untuk rangkaian pembangkit clock. Pin ini dihubungkan dengan osilator kristal atau resonator keramik. Apabila memakai sumber clock eksternal tersendiri, pin yang dihubungkan hanya pin Xin dan Xout dibiarkan mengambang atau tidak dihubungkan ke mana-mana.
- Ø INT0 sampai INT3 – dipakai sebagai masukan *interrupt* eksternal.
- Ø KI0 sampai KI3 – dipakai sebagai masukan *key input interrupt*.
- Ø CNTR0 – pin output dari timer X.
- Ø CNTR0 – pin input atau output dari timer X.
- Ø CNTR1 – pin input atau output dari timer Y.
- Ø TZout – pin output dari timer Z.
- Ø TCIN – pin input dari timer C.
- Ø CMPO0 sampai CMPO3 dan CMP10 sampai CMP13 pin output dari timer C.
- Ø CLK0 – pin IO untuk transfer pulsa clock.
- Ø RxD0 dan RxD1 – pin untuk masukan data serial.
- Ø TxD0, TxD10 dan TxD11 – pin untuk keluaran data serial.
- Ø Vref – merupakan untuk tegangan referensi ADC.
- Ø AN0 sampai AN11 – merupakan pin masukan ADC.
- Ø P00 sampai P07, p10 sampai P17, P30 sampai P33, P37, P45 – merupakan pin IO CMOS. Tiap pin memiliki register untuk menentukan fungsinya sehingga tiap-tiap pin bisa dipakai sebagai input atau output secara sendiri – sendiri. Apabila diset sebagai input, pin tersebut bisa dihubungkan atau tidak dengan resistor pull-up internal melalui program. P10 sampai P17 bisa juga berfungsi sebagai port untuk driver LED.

Deskripsi pin bisa juga dilihat dari Tabel 2.3.

**Tabel 2.3** eskripsi Pin R8C Tiny 13

Signal name	Pin name	I/O type	Function
Power supply input	Vcc, Vss	I	Apply 2.7 V to 5.5 V to the Vcc pin. Apply 0 V to the Vss pin.
IVcc	IVcc	O	This pin is to stabilize internal power supply Connect this pin to Vss via a capacitor (0.1 $\mu$ F) Do not connect to Vcc
Analog power supply input	AVcc, AVss	I	These are power supply input pins for A/D converter. Connect the AVcc pin to Vcc. Connect the AVss pin to Vss. Connect a capacitor between pins AVcc and AVss.
Reset input	RESET	I	"L" on this input resets the MCU.
CNVss	CNVss	I	Connect this pin to Vss via a resistor <sup>(1)</sup>
MODE	MODE	I	Connect this pin to Vcc via a resistor
Main clock input	Xin	I	These pins are provided for the main clock generating circuit I/O. Connect a ceramic resonator or a crystal oscillator between the Xin and Xout pins. To use an externally derived clock, input it to the Xin pin and leave the Xout pin open.
Main clock output	Xout	O	
INT interrupt input	INT0 to INT3	I	These are INT interrupt input pins.
Key input interrupt input	KI0 to KI3	I	These are key input interrupt pins.
Timer X	CNTR0	I/O	This is the timer X I/O pin.
	CNTR0	O	This is the timer X output pin.
Timer Y	CNTR1	I/O	This is the timer Y I/O pin.
Timer Z	TZOUT	O	This is the timer Z output pin.
Timer C	TCIN	I	This is the timer C input pin.
	CMP00 to CMP03, CMP10 to CMP13	O	These are the timer C output pins.
Serial interface	CLK0	I/O	This is a transfer clock I/O pin.
	RxD0, RxD1	I	These are serial data input pins.
	TxD0, TxD10, TxD11	O	These are serial data output pins.
Reference voltage input	VREF	I	This is a reference voltage input pin for A/D converter. Connect the VREF pin to Vcc.
A/D converter	AN0 to AN11	I	These are analog input pins for A/D converter.
I/O port	P00 to P07, P10 to P17, P30 to P33, P37, P45	I/O	These are 8-bit CMOS I/O ports. Each port has an I/O select direction register, allowing each pin in that port to be directed for input or output individually. Any port set to input can select whether to use a pull-up resistor or not by program. P10 to P17 also function as LED drive ports.
Input port	P46, P47	I	These are input only pins.

Sumber : Datasheet Renesas

2.1.2. Organisasi memori mikrokontroler R8C Tiny 13 dapat dilihat dalam Tabel 2.1.4.

SFR	0000 <sub>16</sub>										
	002FF <sub>16</sub>										
Internal RAM	00400 <sub>16</sub>										
	0XXX <sub>16</sub>										
	02000 <sub>16</sub>										
Internal ROM (data area)1	00FFD0 <sub>16</sub>	<table border="1"> <tr><td>Underlined Instruction</td></tr> <tr><td>Overflow</td></tr> <tr><td>BRK Instruction</td></tr> <tr><td>Address Match</td></tr> <tr><td>Single Step</td></tr> <tr><td>Watchdog timer, Oscillation stop detection, Voltage detection</td></tr> <tr><td>(Reserved)</td></tr> <tr><td>(Reserved)</td></tr> <tr><td>Reset</td></tr> </table>	Underlined Instruction	Overflow	BRK Instruction	Address Match	Single Step	Watchdog timer, Oscillation stop detection, Voltage detection	(Reserved)	(Reserved)	Reset
	Underlined Instruction										
Overflow											
BRK Instruction											
Address Match											
Single Step											
Watchdog timer, Oscillation stop detection, Voltage detection											
(Reserved)											
(Reserved)											
Reset											
	00FFF <sub>16</sub>										
Internal ROM (ProgramArea)	DYYY <sub>16</sub>										
	0FFF <sub>16</sub>										
Expanding Area	-----0FFF <sub>16</sub>										

Notes :

1. The data flash ROM block A (2K bytes) and block B (2K bytes) are shown
2. Blank spaces are reserved. No access is followed

Type name	Internal ROM		Internal RAM	
	Size	AddressOYYY <sub>16</sub>	Size	Address OXXX <sub>16</sub>
<b>R5F21134FP, R5F21134DFP</b>	<b>16K bytes</b>	<b>0C000<sub>16</sub></b>	<b>1K bytes</b>	<b>007FF<sub>16</sub></b>
R5F21133FP, R5F21133DFP	12K bytes	0D000 <sub>16</sub>	768K bytes	006FF <sub>16</sub>
R5F21132FP, R5F21132DFP	8K bytes	0E000 <sub>16</sub>	512K bytes	005FF <sub>16</sub>

Sumber : Datasheet Renesas

Gambar di atas merupakan peta memori dari mikrokontroler R8C Tiny 13. Mikrokontroler tersebut memiliki ruang alamat sebesar 1 Mbytes dimulai dari 00000<sub>16</sub> hingga FFFFF<sub>16</sub>. Internal ROM (*Program Area*) terletak pada alamat 0FFF<sub>16</sub>, sebagai contoh, sebuah internal ROM 16 Kbyte terletak pada alamat-alamat dari 02000<sub>16</sub> hingga 02FFF<sub>16</sub>. Internal RAM terletak pada alamat 00400<sub>16</sub> hingga 007FF<sub>16</sub>. SFR terletak pada alamat-alamat dimulai dari 00000 hingga 002FF, disana pula terletak *peripheral function control register*.

### 2.1.3 Reset

Terdapat tiga macam reset yakni *hardware* reset, *software* reset, dan *watchdog timer* reset.

#### 2.1.3.1 Hardware Reset

*Hardware* reset terdiri dari tiga macam reset yakni *hardware* reset 1, *hardware* reset 2, dan *power-on reset*. Setelah reset, *low speed* dari *on-chip oscillator* chip terbagi menjadi delapan, dimana hal tersebut telah dipilih secara otomatis untuk CPU.

Pada *hardware* reset 1, reset dilakukan oleh pin RESET. Ketika sebuah 'Low' sinyal sebagai masukan pin RESET saat tegangan *power supply* pada kondisi operasi yang diijinkan, pin-pin akan diinisialisasi. Saat level input pada pin RESET bergerak dari 'Low' to 'High', maka CPU dan SFR akan diinisialisasi, dan program dimulai dari alamat yang diindikasi oleh vektor reset. Tabel 2.5 menunjukkan status register CPU setelah reset. Internal RAM tidak diinisialisasi.

**Tabel 2.5** Status pin saat level pin RESET adalah ‘L’

Nama Pin	Status Pin
P0	Input Port
P1	Input Port
P3.0 to P3.3, P3.7	Input Port
P4.5 to P4.7	Input Port

### 2.1.3.2 Software Reset

Saat PM03 bit dan register PM0 di set “1” (mikrokomputer reset), mikrokomputer memiliki pin tersebut, CPU, dan inisialisasi SFR. Kemudian program diputuskan untuk dimulai dari alamat yang terindikasi oleh vektor reset. Setelah reset, *low-speed* dari *on-chip oscillator clock* terbagi menjadi delapan yang secara otomatis terpilih untuk CPU. Beberapa SFR tidak dapat terinisialisasi oleh software reset (keterangan pada *datasheet*).

### 2.1.3.3 Watchdog Timer Reset

Saat PM12 bit dalam register PM1 berlogika “1”, mikrokomputer menginisialisasi pin-pinnya, CPU, dan SFR jika *watchdog timer underflows*. Kemudian program diputuskan untuk dimulai dari alamat yang terindikasi oleh vektor reset. Setelah reset, *low-speed* dari *on-chip oscillator clock* terbagi menjadi delapan yang secara otomatis terpilih untuk CPU. Beberapa SFR tidak dapat terinisialisasi oleh software reset (keterangan pada *datasheet*).

#### 2.1.4 Proteksi.

Jika pada program terjadi sesuatu hal di luar kendali, ada satu register yang memproteksi register yang memiliki fungsi penting sehingga tidak mudah error. Register tersebut adalah PRCR register. Register-register yang diproteksi oleh PRCR register dijabarkan pada penjelasan berikut.

- Ø Register yang di proteksi oleh PRCR 0 bit adalah : register CM0, CM1, dan OCD, HR1, HR0.
- Ø Register yang di proteksi oleh PRCR 1 bit adalah : register PM0 dan PM1.
- Ø Register yang di proteksi oleh PRCR 2 bit adalah : register PD0.
- Ø Register yang di proteksi oleh PRCR 3 bit adalah : register VCR2 dan D4INT.

PRCR 2 *disetting* dengan logika “1” dan menuliskannya pada alamat register lain, kemudian PRCR2 akan *tersetting* dengan logika “0”. Register yang diproteksi oleh PRCR 2 akan berubah pada instruksi selanjutnya setelah setting PRCR 2 berubah menjadi berlogika “1”. Harus dapat dipastikan bahwa tidak terjadi interupsi antara instruksi saat PRCR 2 di set berlogika “1” dengan instruksi selanjutnya. PRCR 0, PRCR 1, dan PRCR 3 tidak secara otomatis di set berlogika “0” hanya dengan *write* pada alamat lain, namun hanya dapat di set dengan logika “0” dalam program.

#### 2.1.5 Komunikasi Serial

Komunikasi serial mikrokontroler R8C Tiny 13 memiliki dua *channel* yakni UART0 dan UART1. Masing-masing memiliki sebuah eksklusif timer untuk mengatur transfer clock sehingga masing-masing *channel* dapat bekerja secara terpisah.

UART0 memiliki dua mode yakni : *clock synchronous serial I/O mode* dan *clock asynchronous serial I/O mode*(UART Mode). Sedangkan UART1 hanya memiliki satu mode saja, yakni *clock asynchronous serial I/O mode*(UART Mode).

##### 2.1.5.1 *clock synchronous serial I/O mode*

Mode ini menggunakan sebuah transfer clock untuk mengirim dan menerima data, dan dapat dipilih dengan UART0. Spesifikasi mode ini dilampirkan dalam *datasheet*. Spesifikasi mode UART dapat dilihat pada Tabel 2.6.

**Tabel 2.6.** Spesifikasi mode UART

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>Transfer data length: 8 bits</li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>CKDIR bit in U0MR register is set to "0" (internal clock): <math>f_i/(2(n+1))</math>  <math>f_i=f_{1SIO}, f_{2SIO}, f_{3SIO}</math>    <math>n</math>=setting value in UiBRG register: 0016 to FF16</li> <li>CKDIR bit is set to "1" (external clock ): input from CLK0 pin</li> </ul>
Transmission start condition	<ul style="list-style-type: none"> <li>Before transmission can start, the following requirements must be met<sup>1</sup> <ul style="list-style-type: none"> <li>TE bit in U0C1 register is set to "1" (transmission enabled)</li> <li>TI bit in U0C1 register is set to "0" (data present in U0TB register)</li> </ul> </li> </ul>
Reception start condition	<ul style="list-style-type: none"> <li>Before reception can start, the following requirements must be met<sup>1</sup> <ul style="list-style-type: none"> <li>RE bit in U0C1 register is set to "1" (reception enabled)</li> <li>TE bit in U0C1 register is set to "1" (transmission enabled)</li> <li>TI bit in U0C1 register is set to "0" (data present in the U0TB register)</li> </ul> </li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>For transmission, one of the following conditions can be selected <ul style="list-style-type: none"> <li>U0IRS bit is set to "0" (transmit buffer empty): when transferring data from U0TB register to UART0 transmit register (at start of transmission)</li> <li>U0IRS bit is set to "1" (transfer completed): when serial interface finished sending data from UARTi transmit register</li> </ul> </li> <li>For reception <ul style="list-style-type: none"> <li>When transferring data from the UART0 receive register to the U0RB register (at completion of reception)</li> </ul> </li> </ul>
Error detection	<ul style="list-style-type: none"> <li>Overrun error<sup>2</sup>  This error occurs if serial interface started receiving the next data before reading the U0RB register and received the 7th bit of the next data</li> </ul>
Select function	<ul style="list-style-type: none"> <li>CLK polarity selection  Transfer data I/O can be chosen to occur synchronously with the rising or the falling edge of the transfer clock</li> <li>LSB first, MSB first selection  Whether to start sending/receiving data beginning with bit 0 or beginning with bit 7 can be selected</li> <li>Continuous receive mode selection  Reception is enabled immediately by reading the U0RB register</li> </ul>

Sumber : Datasheet Renesas

### 2.1.5.2 clock asynchronous serial I/O mode(UART Mode).

UART mode memperbolehkan mengirim dan menerima data setelah mensetting *bit rate* sesuai dengan keinginan dan mengirim format data.

Spesifikasi mode UART dapat dilihat pada Tabel 2.7.

**Tabel 2.7.** Spesifikasi mode UART

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>Character bit (transfer data): selectable from 7, 8 or 9 bits</li> <li>Start bit: 1 bit</li> <li>Parity bit: selectable from odd, even, or none</li> <li>Stop bit: selectable from 1 or 2 bits</li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>UiMR(i=0, 1) register CKDIR bit = 0 (internal clock) : <math>f_j / (16(n+1))</math>  <math>f_j = f_{1SIO}, f_{8SIO}, f_{32SIO}</math> n=setting value in UiBRG register: 0016 to FF16</li> <li>CKDIR bit = "1" (external clock) : <math>f_{EXT} / (16(n+1))</math>  <math>f_{EXT}</math>: input from CLK pin n=setting value in UiBRG register: 0016 to FF16</li> </ul>
Transmission start condition	<ul style="list-style-type: none"> <li>Before transmission can start, the following requirements must be met                             <ul style="list-style-type: none"> <li>TE bit in UiC1 register = 1 (transmission enabled)</li> <li>T1 bit in UiC1 register = 0 (data present in UiTB register)</li> </ul> </li> </ul>
Reception start condition	<ul style="list-style-type: none"> <li>Before reception can start, the following requirements must be met                             <ul style="list-style-type: none"> <li>RE bit in UiC1 register = 1 (reception enabled)</li> <li>Start bit detection</li> </ul> </li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>For transmission, one of the following conditions can be selected                             <ul style="list-style-type: none"> <li>UiIRS bit = 0 (transmit buffer empty): when transferring data from UiTB register to UARTi transmit register (at start of transmission)</li> <li>UiIRS bit = 1 (transfer completed): when serial interface finished sending data from UARTi transmit register</li> </ul> </li> <li>For reception                             <ul style="list-style-type: none"> <li>When transferring data from UARTi receive register to UiRB register (at completion of reception)</li> </ul> </li> </ul>
Error detection	<ul style="list-style-type: none"> <li>Overrun error<sup>1</sup>                              This error occurs if serial interface started receiving the next data before reading UiRB register and received the bit one before the last stop bit of the next data</li> <li>Framing error                              This error occurs when the number of stop bits set is not detected</li> <li>Parity error                              This error occurs when if parity is enabled, the number of 1's in parity and character bits does not match the number of 1's set</li> <li>Error sum flag                              This flag is set (= 1) when any of the overrun, framing, and parity errors is encountered</li> </ul>
Select function	<ul style="list-style-type: none"> <li>TxD10, RxD1 selection (UART)                              P37 pin can be used as RxD1 pin or TxD10 pin in UART1. Select by a program.</li> <li>TxD11 pin selection (UART1)                              P00 pin can be used as TxD11 pin in UART1 or port P00. Select by a program.</li> </ul>

Sumber : Datasheet Renesas

### 2.1.5.3 Bit rate

Pada mode UART, saat memilih internal clock, nilai UiBRG register dapat disetting dengan cara :

$$[f_j \div (\text{Bitrate} \times 16)] - 1 \dots \dots \dots 2.0.a$$

dengan  $f_j$  adalah frekuensi dari register UiBRG ( $f_{1SIO}$ ,  $f_{8SIO}$ , dan  $f_{32SIO}$ ).

Saat memilih eksternal clock, maka nilai register UiBRG dapat dihitung dengan cara :

$$[f_{EXT} \div \text{Bitrate} \times 16] - 1 \dots \dots \dots 2.0.b$$

dengan fEXT adalah frekuensi dari register UiBRG (eksternal clock).

Contoh setting bit rate pada mode UART dapat dilihat pada Tabel 2.8.

**Tabel 2.8.** ontoh Bit rate *setting*

<b>&lt;UART Mode&gt;</b>	
• When selecting internal clock	
Setting value to the UiBRG register =	$\frac{f_j}{\text{Bit Rate} \times 16} - 1$
fj : Count source frequency of the UiBRG register (f1SIO, f8SIO and f32SIO)	
• When selecting external clock	
Setting value to the UiBRG register =	$\frac{f_{\text{EXT}}}{\text{Bit Rate} \times 16} - 1$
fEXT : Count source frequency of the UiBRG register (external clock)	

**Sumber :** Datasheet Renesas

R8C Tiny bisa diprogram menggunakan Bahasa C, kapasitas memori lebih besar, merupakan mikrokontroler keluaran terbaru dengan kecepatan taransfer data tinggi. Data dapat tersimpan walau lampu mati dan dapat terhubung langsung ke printer .

## 2.2 Mikrokontroler AT89S51

Mikrokontroler AT89S51 merupakan mikrokontroler buatan ATMEL yang didesain untuk kompatibel dengan produk MCS-51<sup>TM</sup> dan memiliki *Flash Memory* dengan kapasitas sebesar 4K *Byte*, 128 x 8 bit internal RAM dan 32 jalur I/O yang dapat diprogram, 2 buah *timer/counter* 16 bit, *Full duplex serial port*, *On Chip oscillator*, Beroperasi pada frekuensi 0 – 24 MHz. Mikrokontroler jenis ini menggunakan teknologi mikrokomputer CMOS 8-bit dengan fasilitas *Electrically Erasable Programmable Read Only Memory (EEPROM)* yang dapat diprogram ulang hingga kurang lebih 1000 kali.

Keduanya didesain untuk 2 mode *power saving* yaitu *idle mode* dan *power down mode*. *Idle mode* menghentikan CPU sementara sampai RAM, timer/counter, port serial dan sistem interrupt melanjutkan fungsinya. Sedangkan *Power Down Mode* masih menyimpan RAM tapi *oscillator* menghentikan semua fungsi chip lain sampai *hardware* kembali *reset*. Selain itu mikrokontroler ini mempunyai kemampuan tinggi, fleksibel, dengan harga murah untuk berbagai aplikasi. Konfigurasi pin mikrokontroler AT89S51 ditunjukkan dalam Gambar 2.3.

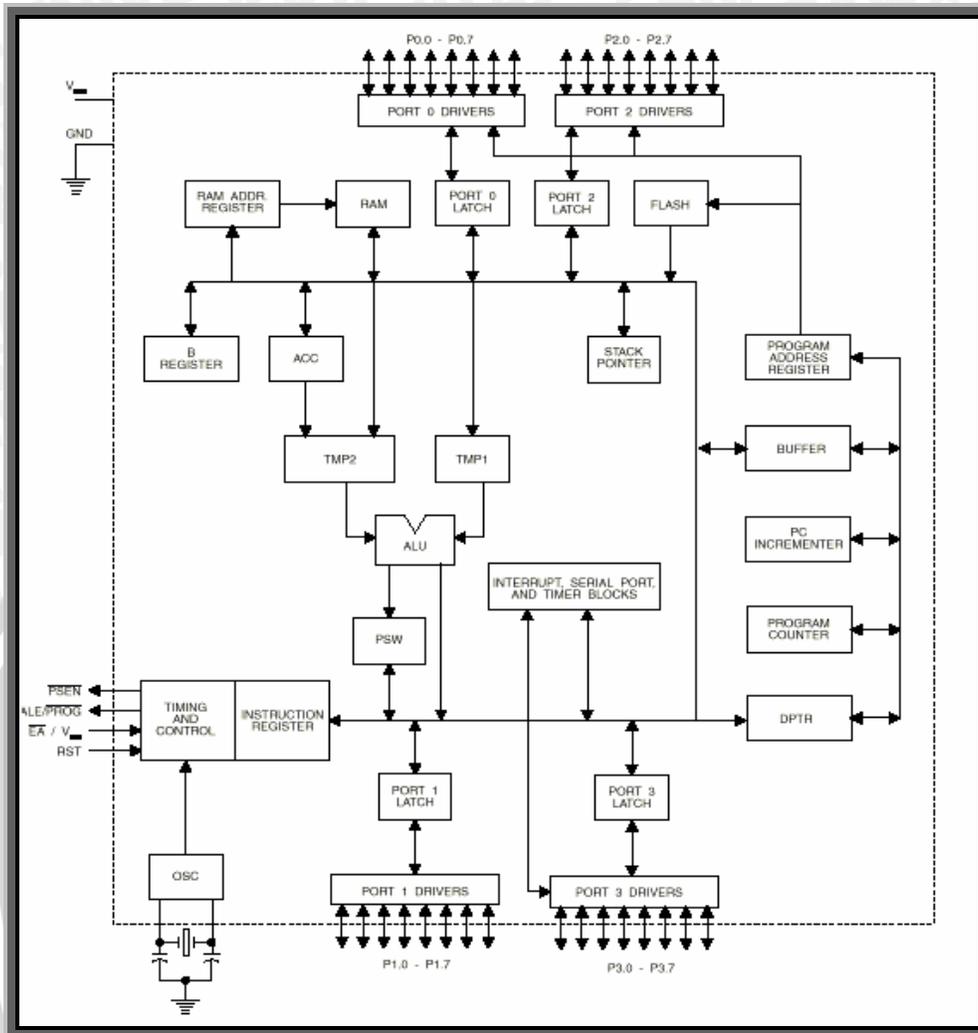
P1.0	1	40	VCC
P1.1	2	39	P0.0 (AD0)
P1.2	3	38	P0.1 (AD1)
P1.3	4	37	P0.2 (AD2)
P1.4	5	36	P0.3 (AD3)
P1.5	6	35	P0.4 (AD4)
P1.6	7	34	P0.5 (AD5)
P1.7	8	33	P0.6 (AD6)
RST	9	32	P0.7 (AD7)
(RXD) P3.0	10	31	EA/VPP
(TXD) P3.1	11	30	ALE/PROG
(INT0) P3.2	12	29	PSEN
(INT1) P3.3	13	28	P2.7 (A15)
(T0) P3.4	14	27	P2.6 (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 2.3**  
Konfigurasi pin AT89S51

Sumber : ATMEL *Data Sheet*, 1997: 1

### 2.2.1 Arsitektur Dalam AT89S51

Berdasarkan fungsinya, memori mikrokontroler AT89S51 dapat dipisahkan menjadi dua bagian, yaitu memori program dan memori data. Memori program digunakan untuk menyimpan instruksi-instruksi mikrokontroler, sedangkan memori data digunakan sebagai tempat penyimpanan data sementara yang akan diolah mikrokontroler. Ada dua jenis memori yang digunakan sebagai pendukung sistem mikrokontroler, yaitu RAM dan ROM. Memori program menggunakan ROM sedangkan memori data biasanya menggunakan memori RAM. Blok Diagram internal mikrokontroler ditunjukkan pada Gambar 2.4.



**Gambar 2.4**  
 Arsitektur Dalam AT89S51  
 Sumber : ATMEL *Data Sheet*, 1997: 2

Bagian-bagian yang dimiliki mikrokontroler ini adalah sebagai berikut :

1. Ruang memori program

Berisi program. Mempunyai standar lebar 4Kbytes. Program mikrokontroler disimpan di dalam memori program berupa ROM. Mikrokontroler AT89S51 dilengkapi dengan ROM internal, sehingga untuk menyimpan program tidak harus menggunakan ROM eksternal yang terpisah dari mikrokontroler. Untuk dapat menggunakan memori program eksternal pin EA dihubungkan dengan pin  $V_{SS}$ .

Memori program mikrokontroler menggunakan alamat 16 bit mulai 0000H sampai FFFFH, sehingga kapasitas penyimpanan program sebesar  $2^{16}$  (65535) byte atau 64 Kbyte. Sinyal yang digunakan untuk membaca memori program eksternal adalah sinyal  $\overline{\text{PSEN}}$  (*Program Store Enable*).

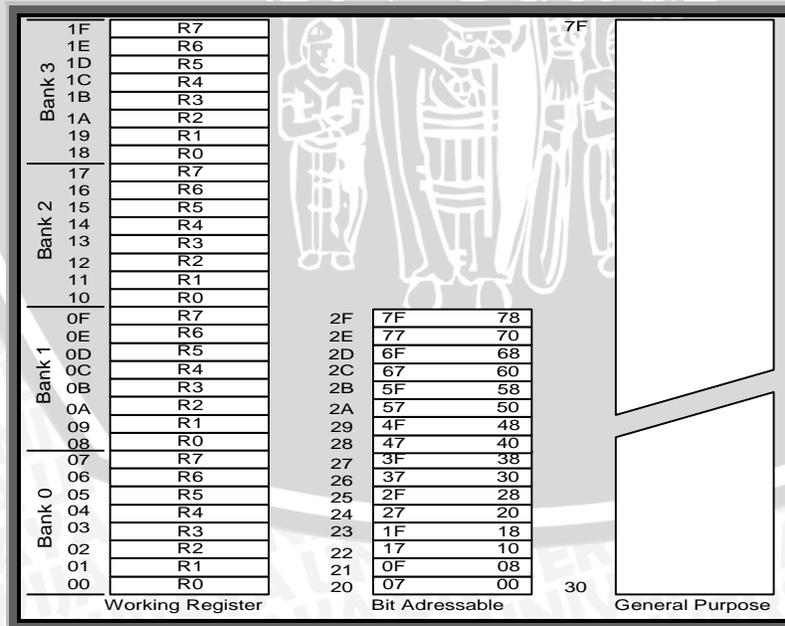
## 2. RAM/GPR

Disediakan untuk keperluan programmer.

RAM diorganisasi menjadi 3 area, yaitu:

- 32 bit (dari alamat 00H-1FH) diorganisir menjadi 4 bank. Masing-masing bank terdiri dari 8 register (R0-R7).
- Area bit *addressable* (alamat 20H-2FH).
- *General Purpose* RAM (alamat 30H-7FH).

Mikrokontroler AT89S51 memiliki memori data internal berupa RAM sebesar 128 byte. Selain itu AT89S51 menyediakan ruang memori data eksternal dengan lebar alamat 16 bit (0000H sampai FFFFH) atau sebesar 64 Kbyte. Untuk mengakses memori data eksternal digunakan sinyal baca  $\overline{\text{RD}}$  dan sinyal tulis. Pemetaan memori internal AT89S51 dapat dilihat pada Gambar 2.5.



**Gambar 2.5.**

Pemetaan Memori Internal AT89S51  
 Sumber : MCS-51 *user's manual*, 1994:2-4

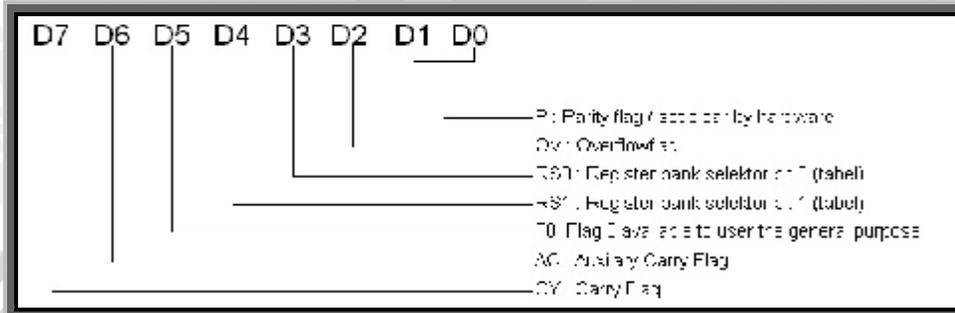
3. SFR (*Special Function Register*)

Fungsi-fungsi khusus dimiliki oleh register pada alamat ini, (30H-7FH).

1. PSW (Program Status Word)

Sesuai namanya adalah merupakan register yang berisi status proses.

Register terletak pada alamat D0H. Didefinisikan sebagai berikut :



**Gambar 2.6.** : Definisi fungsi dan bit PSW

Sumber : Kenneth J. Ayala, 1994: 64

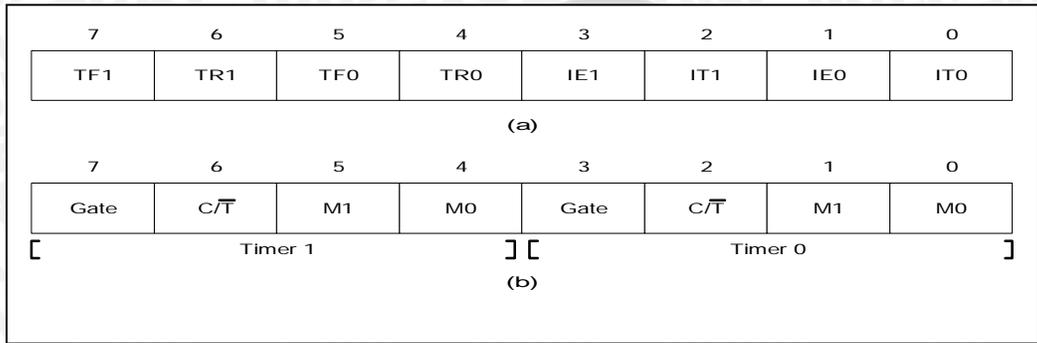
2. TMOD (Timer Mode) dan TCON (Timer Control)

Pada mikrokontroler AT89S51 terdapat dua buah timer/counter 16-bit yang dapat diatur melalui perangkat lunak, yaitu timer/counter 0 dan timer/counter 1. Setiap timer/counter dapat berfungsi sebagai timer yang berjalan *independent* (mencacah siklus processor) atau dapat digunakan untuk mencacah tepi turun sinyal yang diumpungkan pada pin I/O T0 ataupun T1 [Chapman, 1994: 13].

Kedua timer/counter ini dibagi menjadi dua register 8-bit yang disebut dengan timer byte rendah (TL0, TL1) dan timer byte tinggi (TH0, TH1). Seluruh kerja timer/counter dikontrol melalui konfigurasi bit-bit dalam register kontrol mode timer (TMOD), register kontrol timer/counter (TCON), dan instruksi-instruksi program [Ayala, 1991: 29].

Register TMOD didedikasikan untuk kedua timer dan merupakan dua buah duplikat register 4-bit, masing-masing mengontrol kerja dari masing-masing timer. Register TCON memiliki bit-bit kontrol dan *flag*

bagi timer pada *nibble* atas, dan bit-bit kontrol dan flag bagi intrupsi eksternal pada *nibble* bawah. Gambar 2.7. menunjukkan diagram register TCON dan TMOD.

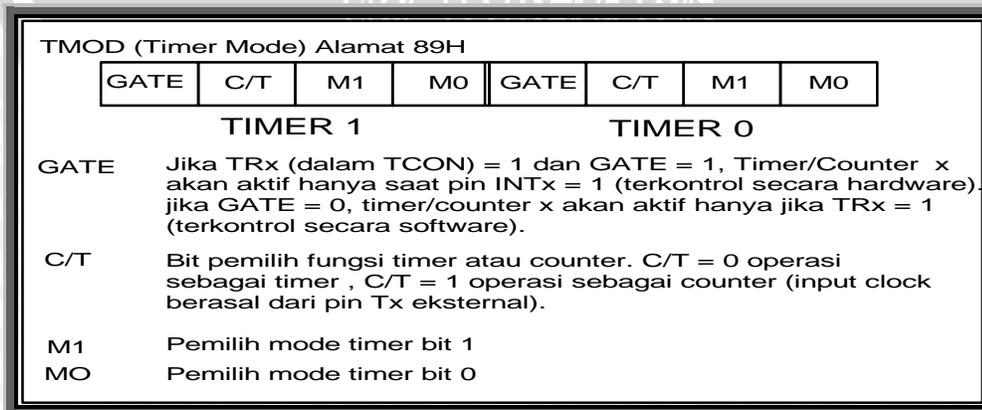


Gambar 2.7.  
Diagram register (a)TCON dan (b)TMOD.

Sumber : Ayala, 1991: 28-29

Timer/counter dapat dioperasikan dalam salah satu dari empat buah mode yang ditentukan oleh bit-bit pemilih mode (M0, M1) dalam register TMOD.

TMOD



**Gambar 2.8.** : Definisi fungsi dan bit TMOD  
Sumber : Kenneth J. Ayala, 1994: 29

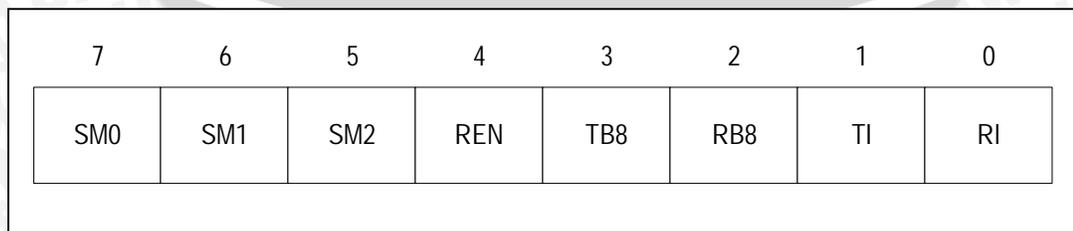
## TCON (Timer Control)

TCON (Timer Control) Alamat 88H							
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
TF1	TCON.7	Timer 1 overflow flag.					
TR1	TCON.6	Timer 1 run control bit.					
TF0	TCON.5	Timer 0 overflow flag.					
TR0	TCON.4	Timer 0 run bit.					
IE1	TCON.3	External Interrupt 1 edge flag.					
IT1	TCON.2	Interrupt 1 type control bit.					
IE0	TCON.1	External Interrupt 0 edge flag.					
IT0	TCON.0	Interrupt 0 type control bit.					

**Gambar 2.9.** : Definisi dan fungsi TCON  
 Sumber : Kenneth J. Ayala, 1994: 65

### 3. SCON (Serial Control)

Mikrokontroler AT89S51 dilengkapi dengan port serial yang memungkinkan untuk mengirimkan data dalam format serial. Port serial terdapat dalam kelompok pin port 3, yaitu port 3.0 sebagai port masukan serial (RxD) dan port 3.1 sebagai port keluaran serial (TxD). Port serial dalam mikrokontroler AT89S51 bersifat full-duplex, yang berarti dapat mengirim dan menerima data secara bersamaan [Malik, 1997: 26]. Register penerima dan pengirim pada port serial dapat diakses pada SBUF (*serial buffer*). Sedangkan register pengontrol kerja port serial ini adalah SCON (*serial control*). Gambar 2.10. menunjukkan diagram register SCON dan gambar 2.8 menunjukkan definisi dan fungsi SCON.



**Gambar 2.10.**Diagram register SCON.

Sumber : Ayala, 1991: 33

SCON (Serial Control) Alamat 98H

	SMO	SM1	SM2	REN	TB8	RB8	TI	RI
SM0	SCON.7							
SM1	SCON.6							
SM2	SCON.5							
REN	SCON.4							
TB8	SCON.3							
RB8	SCON.2							
TI	SCON.1							
RI	SCON.0							

SM0 SCON.7 Pilih mode port serial  
 SM1 SCON.6 Pilih mode port serial  
 SM2 SCON.5 Memungkinkan komunikasi multiprosesor pada mode 2 dan 3. Pada mode 2 atau 3, jika SM2 = 1 maka RI tidak akan diaktifkan bila bit ke 9 yang diterima = 0. Pada mode 1, jika SM2 =1 maka RI tidak akan diaktifkan hingga bit stop yang valid diterima. Pada mode 0, SM2 = 0.  
 REN SCON.4 1= memungkinkan penerimaan, 0= memblok penerimaan (dari port serial)  
 TB8 SCON.3 Bit ke 9 yang akan dikirim dalam mode 2 dan 3.  
 RB8 SCON.2 Pada mode 2 dan 3 bit ini merupakan bit ke 9 yang diterima. Pada mode 1, jika SM2 = 0, RB8 adalah bit stop yang diterima. Pada mode 0, RB8 tidak dipergunakan.  
 TI SCON.1 Flag Transmit Interrupt.  
 RI SCON.0 Flag Receive Interrupt.

**Gambar 2.11** : Definisi dan fungsi SCON

Sumber: Kenneth J. Ayala, 1994: 65

4. IE (Interrupts Enable)

Merupakan register yang berisi bit-bit untuk menghidupkan atau mematikan sumber *interrupt*.

IE (Interrupts Enable) Alamat A8H

	EA	-	ET2	ES	ET1	EX1	ET0	EX0
EA	IE.7							
-	IE.6							
ET2	IE.5							
ES	IE.4							
ET1	IE.3							
EX1	IE.2							
ET0	IE.1							
EX0	IE.0							

EA IE.7 Enable Interrupts Bit. EA = 0 seluruh interrupt akan di disable. EA = 1, memungkinkan masing-masing interrupt dienable/disable  
 - IE.6 Tak digunakan  
 ET2 IE.5 Tidak digunakan, Disediakan untuk penggunaan masa depan.  
 ES IE.4 Enable Serial port Interrupt  
 ET1 IE.3 Enable Interrupt Timer 1  
 EX1 IE.2 Enable External Interrupt 1  
 ET0 IE.1 Enable Interrupt Timer 0  
 EX0 IE.0 Enable External Interrupt 0

**Gambar 2.12** : Definisi dan fungsi IE  
 Sumber : Kenneth J. Ayala: 64

5. IP (Interrupts Priority)

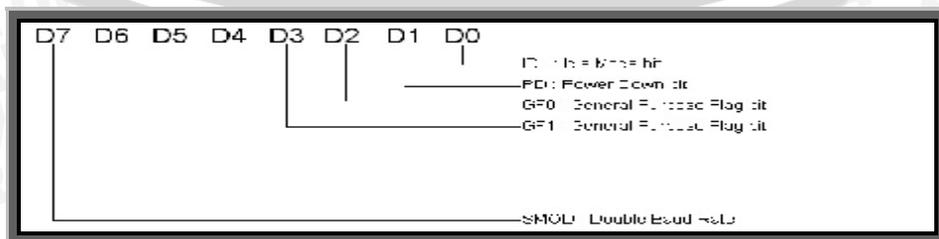
Mikrokontroler AT89S51 mempunyai dua sumber interupsi eksternal dan sumber interupsi internal yang dapat diprogram agar sensitif terhadap perubahan level atau transisi. Interupsi timer aktif saat register timer yang bersangkutan mengalami perputaran (*rollover*), interupsi serial akan aktif pada saat mikrokontroler mengirim/menerima data. Setiap sumber interupsi dapat diaktifkan/dimatikan melalui perangkat lunak. Interupsi yang mempunyai tingkatan prioritas lebih tinggi tidak dapat diinterupsi oleh yang lebih rendah. Meskipun demikian melalui perangkat lunak hirarki tersebut dapat diubah, yaitu dalam register *Interrupt Priority* (IP). Definisi dan fungsi IP dapat dilihat pada Gambar 2.9.

IP (Interrupts Priority) Alamat B8H								
-	-	PT2	PS	PT1	PX1	PT0	PX0	
-	IP.7	Tak digunakan						
-	IP.6	Tak digunakan						
PT2	IP.5	Tidak digunakan, Disediakan untuk penggunaan masa depan.						
PS	IP.4	Prioritas Serial port Interrupt						
PT1	IP.3	Prioritas Interrupt Timer 1						
PX1	IP.2	Prioritas External Interrupt 1						
PT0	IP.1	Prioritas Interrupt Timer 0						
PX0	IP.0	Prioritas External Interrupt 0						

**Gambar 2.13** : Definisi dan fungsi IP  
 Sumber : Kenneth J. Ayala: 64

6. PCON

Register ini terletak pada alamat 87H. Didefinisikan sebagai berikut :



**Gambar 2.14** : Definisi dan fungsi PCON  
 Sumber : Kenneth J. Ayala, 1994: 33

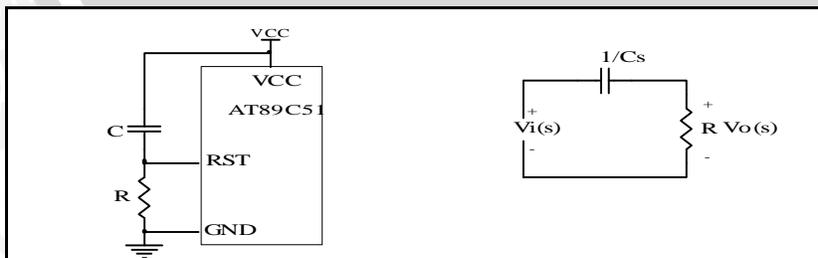
Kondisi SFR setelah di reset ditunjukkan pada tabel 2.9.

REGISTER	VALUE(HEX)
PC	0000
DPTR	0000
A	00
B	00
SP	07
PSW	00
P0-P3	FF
IP	XXX00000b
IE	0XX00000b
TCON	00
TMOD	00
TH0	00
TLO	00
TH1	00
TL1	00
SCON	00
SBUF	XX
PCON	0XXXXXXXb

**Tabel 2.9**  
 Kondisi SFR *Reset*  
 Sumber : Kenneth J. Ayala, 1994: 40

### 2.2.2 Reset

Rangkaian *reset* dibutuhkan untuk me-*reset* mikrokontroler pada saat *power on*. Tegangan berlogika tinggi selama 2 siklus mesin dibutuhkan untuk me-*reset* MCU pada saat dihidupkan. Rangkaian *reset* terdiri dari resistor dan kapasitor yang dihubungkan seperti dalam Gambar 2.7(a).



Gambar 2.15

(a) Rangkaian *n Reset*, (b) Rangkaian Ekuivalen *Reset*

Sumber : Atmel, 1997: 2-63

Dalam rangkaian ekuivalen seperti ditunjukkan dalam Gambar 2.15(b) maka didapatkan :

$$V_o = \frac{R}{R + 1/C_s} \cdot V_i = \frac{RC_s}{RC_s + 1} \cdot V_i \dots\dots\dots(2-1)$$

dengan tegangan  $V_i$  adalah tegangan  $V_{CC}$  yaitu 5V, dimana dalam fungsi Laplace adalah  $\frac{5}{s}$  sehingga :

$$V_o = \frac{RC_s}{RC_s + 1} \cdot \frac{5}{s} = \frac{RC}{RC_s + 1} \cdot 5 = 5 \cdot \left( \frac{1}{s + 1/RC} \right) \dots\dots\dots(2-2)$$

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

$$t = R \cdot C \left( \ln \frac{5}{V_o} \right) \dots\dots\dots(2-3)$$

dengan nilai  $V_o$  adalah tegangan logika nominal yang diijinkan oleh pin RST (Atmel, 1997: 4-37), dimana  $V_o = 0,7 \times V_{CC} = 0,7 \times 5 \text{ volt} = 3,5 \text{ volt}$ , maka :

$t = RC \left[ \ln \frac{5}{3,5} \right]$ , maka

$$t = 0,357 R \cdot C \dots\dots\dots(2-4)$$

Karena kristal yang digunakan mempunyai frekuensi sebesar 11,0592 MHz, maka satu periode membutuhkan waktu sebesar :

$$T = \frac{1}{f_{XTAL}} \dots\dots\dots(2-5)$$

Sehingga waktu minimal logika tinggi yang dibutuhkan untuk mereset mikrokontroler adalah :

$$t_{reset(min)} = T \times \text{periode yang dibutuhkan} \dots\dots\dots(2-6)$$

### 2.2.3 Sistem Port (Masukan dan keluaran)

Mikrokontroler AT89S51 dilengkapi dengan 4 buah *port* dua arah (*bidirectional*) untuk saluran masukan dan keluaran masing-masing sebesar 8 bit. Saluran-saluran ini juga dapat dialamati per bit.

Port 0 digunakan sebagai saluran data yang dimultipleks dengan saluran alamat rendah untuk mengakses memori eksternal, baik memori program maupun memori data. Port 2 mengeluarkan bagian alamat tinggi untuk mode pengalamatan memori 16 bit. Port 1 dan port 3 berfungsi sebagai saluran masukan/keluaran multi fungsi. Penjelasan masing-masing port sebagai berikut.

#### 2.2.3.1. Port 0

Port 0 adalah 8 bit *open drain bidirectional* I/O port. Semua keluaran port setiap pin dapat mensuplai 8 masukan TTL. Saat 1 ditulis ke pin port 0, pin dapat digunakan untuk impedansi input tinggi. Port ini menerima kode *byte* selama *programming flash* dan kode *byte* keluaran dari program selama proses *verifikasi*. *External pull up* dibutuhkan selama verifikasi program berlangsung.

#### 2.2.3.2. Port 1

Port 1 adalah 8 bit *directional* I/O port dengan *pull up*. Keluaran penyangga port 1 dapat *sink/source* 4 buah input TTL. Ketika 1 ditulis ke pin port 1 maka *pull-up* akan berisi 1 pada *pull up* dan dapat digunakan sebagai input. Port ini menerima *low order address byte* selama memprogram dan verifikasi dari flash.

#### 2.2.3.3. Port 2

Port 2 adalah 8-bit *directional* I/O port dengan *internal pull up*. Buffer keluaran port 2 dapat *sink/source* 4 buah input TTL. Ketika 1 ditulis ke pin port 2 maka akan berisi *logic high* oleh *internal pullup* dan dapat digunakan sebagai masukan.

Pin port 2 yang berisi *external pull low* akan berisi arus ( $I_{IL}$ ) sebab ada *internal pull up*. Port ini menerima pengalamatan *high-order bits* dan beberapa sinyal kontrol selama pemrograman dan verifikasi *flash*.

### 2.2.3.4. Port 3

Port 3 adalah 8-bit directional I/O port dengan pull up dalam. Penyangga keluaran port 3 dapat *sink/source* untuk 4 masukan TTL. Ketika 1 ditulis ke port 3 maka akan terjadi *pulled high* oleh *pullup* internal dan dapat digunakan sebagai masukan. Pin port 3 bersifat *external pulled low* akan mengeluarkan arus sebesar ( $I_{PL}$ ), sinyal control untuk pemrograman dan verifikasi diterima oleh port ini.

Nomor Pin	Fungsi Normal	Fungsi Alternatif
P3.0	BIT 0	RXD (input port serial)
P3.1	BIT 1	TXD (output port serial)
P3.2	BIT 2	INT0 (input 0 interup eksternal)
P3.3	BIT 3	INT1 (input 1 interup eksternal)
P3.4	BIT 4	T0 (input 0 counter eksternal)
P3.5	BIT 5	T1 (input 1 counter eksternal)
P3.6	BIT 6	WR (sinyal kontrol memori data eksternal)
P3.7	BIT 7	RD (sinyal kontrol memori data eksternal)

**Tabel 2.10** Fungsi Alternatif Port 3

Sumber : ATMEL *data sheet*, 1997: 3

Port 0 dan 2 memiliki fungsi alternatif menjadi bus alamat dan bus data saat mengakses ruang memori eksternal. Bila pada sistem tidak diperlukan memori eksternal, maka port 0 dan port 2 dapat dipergunakan oleh programmer.

### 2.2.4. Mode-mode transmisi data serial

Mikrokontroler AT89S51 dirancang dengan dilengkapi empat mode transmisi data serial yang memungkinkan komunikasi dapat dilakukan dengan banyak cara dan dengan banyak pilihan kecepatan (baud rate). Mode transmisi ditentukan oleh nilai bit SM0 dan bit SM1 dalam register SCON yang dapat dialamati per bit. Melalui program, mode transmisi dipilih dengan mengisikan bit-bit tersebut dengan nilai yang sesuai.

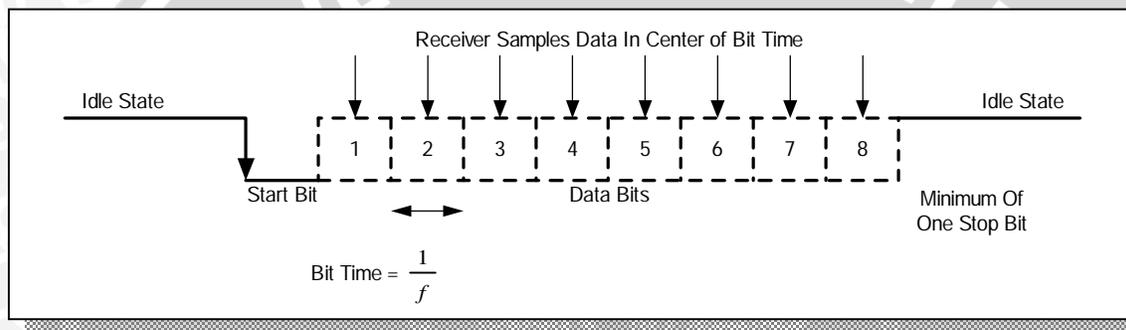
#### 2.2.4.1. Mode 0 (mode register geser)

Mode 0 dipilih dengan mengisi bit SM0 dan SM1 dalam register SCON dengan nilai 00b. Dalam mode ini, register SBUF dikonfigurasi untuk menerima dan mengirim data

melalui pin RxD. Pin TxD terhubung dengan sumber *clock* pergeseran internal untuk mensuplai clock pergeseran rangkaian eksternal. Frekuensi pergeseran, atau baud rate, memiliki nilai tetap yaitu 1/12 dari frekuensi osilator.

#### 2.2.4.2. Mode 1 (UART standart)

Mode 1 dipilih dengan mengisi bit SM0 dan SM1 dalam register SCON dengan nilai 01b. Dalam mode ini, register SBUF berlaku sebagai transceiver full-duplex 10-bit yang dapat mengirim dan menerima data secara bersamaan. Pin RxD menerima semua data, dan pin TxD mengirim semua data. Gambar 2.15. menunjukkan format dari sebuah *word* data.



Gambar 2.16.

Format word data dalam mode 1.

Sumber : Ayala, 1991: 36

Data yang dikirim berupa bit start, delapan bit data (LSB terlebih dahulu), dan bit stop. Flag intrerupsi TI di-set ketika kesepuluh bit telah dikirim. Interval setiap bit adalah kebalikan dari frekuensi baud rate, dan setiap bit dibuat tinggi ataupun rendah dalam interval tersebut.

Data diterima dalam urutan yang sama, penerimaan dipicu oleh tepi turun dari bit start. Dari kesepuluh bit tersebut, bit start dibuang, delapan bit data disimpan ke dalam register SBUF, dan bit stop disimpan ke dalam bit RB8 dari register SCON. RI di-set 1, menandakan bahwa sebuah *byte* data baru telah diterima.

Timer 1 dipakai untuk membangkitkan baud rate untuk mode 1 dengan menggunakan flag *overflow* dari timer tersebut untuk menentukan frekuensi baud rate. Biasanya timer 1

digunakan dalam mode 2 timer sebagai timer *auto-reload* 8-bit yang menghasilkan frekuensi baud rate:

$$f_{\text{baud}} = \frac{2^{\text{SMOD}}}{32d} \times \frac{\text{frekuensi osilator}}{12d \times [256d - (\text{TH1})]} \quad (2.7)$$

SMOD adalah bit kontrol yang berlokasi di dalam register PCON dan dapat bernilai 0 atau 1, yang dapat menentukan nilai  $2^{\text{SMOD}}$  di dalam Persamaan 2.3 menjadi 1 atau 2.

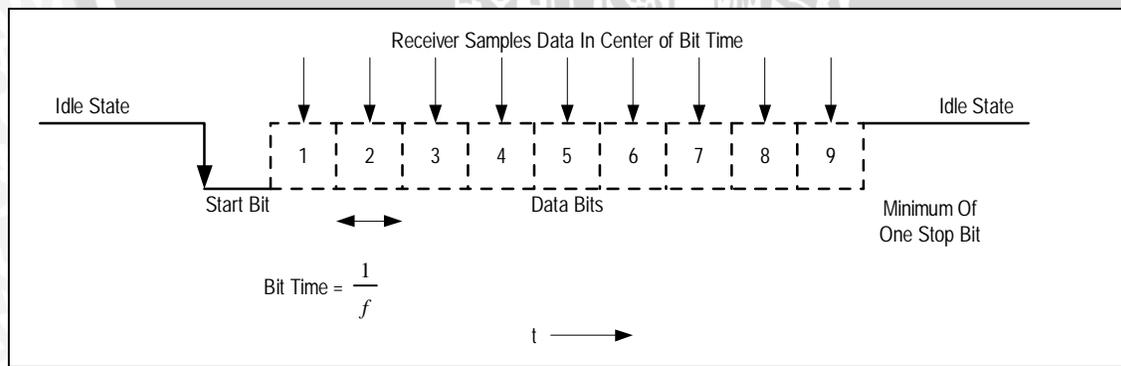
Jika timer tidak dijalankan dalam mode 2 timer, maka nilai baud rate-nya:

$$f_{\text{baud}} = \frac{2^{\text{SMOD}}}{32d} \times (\text{frekuensi overflow timer 1}) \quad (2.8)$$

dan timer 1 dapat dijalankan menggunakan clock internal atau sebagai counter yang menerima pulsa clock dari sumber eksternal melalui pin T1.

#### 2.2.4.3. Mode 2 (Mode Multiprosesor)

Mode 2 mirip dengan mode 1, kecuali 11 bit yang ditransmisikan: bit start, sembilan bit data, dan bit stop, seperti ditunjukkan dalam Gambar 2.17. Bit data ke-9 diambil dari bit TB8 dari register SCON selama pengiriman dan diletakkan ke dalam bit RB8 dari register SCON pada saat data diterima. Baik bit start maupun bit stop dibuang.



Gambar 2.17. Format word data daam mode 2

Sumber : Ayala, 1991: 37

Baud rate di program sebagai berikut:

$$f_{\text{baud}} = \frac{2^{\text{SMOD}}}{64d} \times \text{frekuensi osilator} \quad (2.9)$$

### 2.2.4.4. Mode 3

Mode 3 indentik dengan mode 2 kecuali bahwa baud rate-nya ditentukan sama persis seperti pada mode 1, menggunakan timer 1 untuk menghasilkan frekuensi komunikasi. Port 3 adalah 8-bit directional I/O port dengan pull up dalam. Penyangga keluaran port 3 dapat *sink/source* untuk 4 masukan TTL. Ketika 1 ditulis ke port 3 maka akan terjadi *pulled high* oleh *pullup* internal dan dapat digunakan sebagai masukan. Pin port 3 bersifat *external pulled low* akan mengeluarkan arus sebesar ( $I_{IL}$ ), sinyal control untuk pemrograman dan verifikasi diterima oleh port ini.

Port 0 dan 2 memiliki fungsi alternatif menjadi bus alamat dan bus data saat mengakses ruang memori eksternal. Bila pada sistem tidak diperlukan memori eksternal, maka port 0 dan port 2 dapat dipergunakan oleh programer.

## 2.3 Transistor Sebagai Saklar

Cara yang paling mudah menggunakan sebuah transistor adalah sebagai saklar, artinya transistor dioperasikan pada salah satu dari saturasi atau *cut off*. Jika transistor berada pada titik saturasi, transistor tersebut seperti sebuah saklar yang tertutup dari kolektor ke emitor. Jika transistor *cut off* maka transistor akan seperti sebuah saklar yang terbuka.

Transistor yang difungsikan sebagai saklar diperlihatkan dalam Gambar 2.18. Pada rangkaian tersebut merupakan penjumlahan tegangan disekitar *loop input*, sehingga diperoleh persamaan:

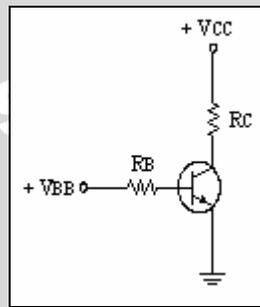
$$I_B \cdot R_B + V_{BE} - V_{BB} = 0 \quad \dots\dots\dots (2.10)$$

$$I_C \cdot R_C + V_{CE} - V_{CC} = 0 \quad \dots\dots\dots (2.11)$$

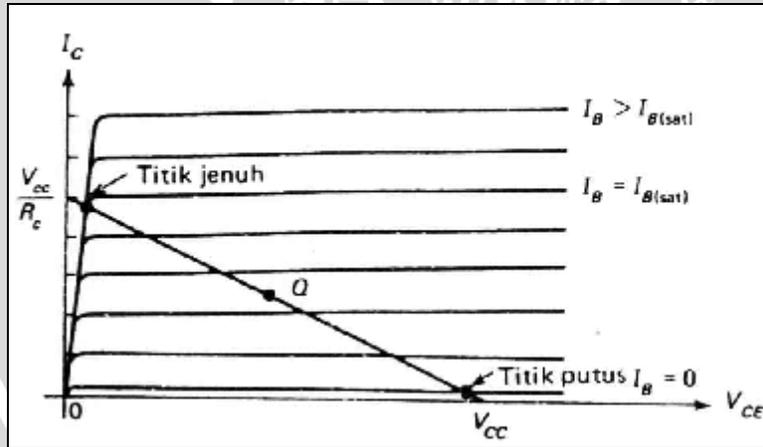
Sehingga dengan persamaan (2-8) didapat persamaan untuk mengetahui besar arus pada kutub basis ( $I_B$ ). Maka persamaan untuk arus pada basis dalam rangkaian transistor sebagai saklar adalah:

$$I_B = \frac{V_{BB} - V_{BE}}{R_B} \dots\dots\dots (2.12)$$

Jika arus basis lebih besar atau sama besar dengan  $I_B$  (sat), titik kerja Q berada pada ujung atas dari garis beban (Gambar 2.18). Dalam hal ini transistor kelihatan seperti sebuah saklar yang tertutup. Sebaliknya jika arus basis nol, transistor bekerja pada ujung bawah dari garis beban, dan transistor kelihatan seperti sebuah saklar yang terbuka.



**Gambar 2.18.** Rangkaian transistor sebagai saklar  
 Sumber: Malvino, 1992: 271



**Gambar 2.19.** Titik jenuh dan titik putus pada garis beban dc  
 Sumber: Malvino, 1994: 139

Titik potong antara garis beban dc dan kurva  $I_B = 0$  dikenal sebagai titik putus (*cutoff*). Pada titik ini, arus basis sama dengan nol, dan arus kolektor juga sangat kecil. Ternyata dalam keadaan ini, diode emiter tidak lagi mendapat prategangan maju dan fungsi normal dari transistor menjadi terhenti..

Titik jenuh (*saturasion*) adalah titik potong dengan kurva  $I_B$  pada ujung teratas dari garis beban dc. Untuk mudahnya, titik jenuh akan diberi koordinat  $I_{C(sat)}$  dan  $V_{CE(sat)}$ . Arus kolektor pada titik ini ditentukan oleh

$$I_{c(sat)} = \frac{V_{cc} - V_{ce(sat)}}{R_c} \dots\dots\dots(2.13)$$

Dari gambar 2.19 tampak jelas bahwa titik jenuh ini sangat berdekatan dengan sumbu tegak. Karena lazimnya  $V_{CE(sat)}$  hanya berharga beberapa kali 0,1 V, persamaan diatas sering dituliskan dalam bentuk

$$I_{c(sat)} \cong \frac{V_{cc}}{R_c} \dots\dots\dots(2.14)$$

Arus basis yang menyebabkan kejenuhan ini diberikan oleh

$$I_{B(sat)} = \frac{I_{C(sat)}}{\beta_{dc}} \dots\dots\dots(2.15)$$

Bila arus basis melebihi  $I_{B(sat)}$ , arus kolektor tidak dapat bertambah lagi karena diode kolektor tidak lagi berfungsi (tidak mendapat prategangan balik). Dengan kata lain,  $I_{C(sat)}$  merupakan harga maksimum dari arus kolektor yang dapat dihasilkan oleh rangkaian dengan prategangan basis pada tegangan catu dan hambatan kolektor tertentu.

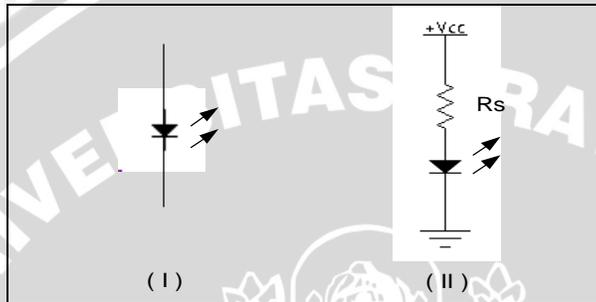
**2.4. Sensor**

Untuk sensor pembacaan digunakan *infrared dioda* dan *photodiode*. Komponen ini berfungsi sebagai sensor, jika tidak ada mobil yang melintas diantara sensor maka sinar dari dioda akan mengaktifkan transistor dan diartikan sebagai logika 1, begitu juga sebaliknya jika ada mobil yang melintas diantara sensor maka transistor tidak mendapat picu sehingga menjadikan logika 0.

**2.4.1. LED (Light Emitting Diode) Infrared**

Dioda pancar cahaya (*light Emiting Diode = LED*) adalah dioda semikonduktor khusus yang dirancang untuk memancarkan cahaya apabila arus melaluinya. Apabila diberi bias maju, energi elektron yang mengalir diubah menjadi energi cahaya. LED harus dioperasikan di dalam ukuran kerja tegangan dan arus tertentu untuk mencegah

kerusakan. Sebagian besar LED membutuhkan 1,2 V sampai 2,5 V dan arus diantara 20 mA sampai 50 mA. Keuntungan utama penggunaan LED sebagai sumber cahaya dibandingkan dengan bola lampu cahaya biasa adalah penggunaan daya yang jauh lebih rendah, jauh lebih lama umurnya, dan beroperasi dengan kecepatan tinggi. Dioda konvensional mengubah energi menjadi panas. Dioda *arsenide gallium* mengubah energi menjadi panas dan sinar inframerah. Lambang skematis dan contoh rangkaian dari LED dapat dilihat dalam Gambar 2.20.



**Gambar 2.20.**

Lambang skematis dan contoh rangkaian dari LED

Sumber: Malvino, 1985 : 97

Cara mengendalikan LED yaitu dengan memperhitungkan arus dan tegangan sumber. Besar arus LED yang diberikan dalam contoh rangkaian Gambar 2.20 adalah :

$$I = \frac{V_{CC} - V_{IR}}{R_s} \dots\dots\dots(2-16)$$

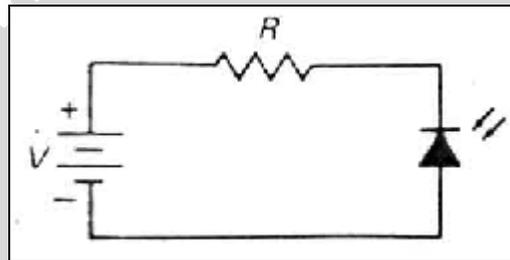
$$R_s = \frac{V_{CC} - V_{IR}}{I} \dots\dots\dots(2-17)$$

**2.4.2. Fotodioda**

Komponen *photodiode* mempunyai karakteristik seperti komponen yang dinamakan ‘*solar cell*’, yang merubah energi cahaya menjadi energi listrik. Jika photodiode ini mendapat cahaya maka akan menghasilkan tegangan sekitar 0,5 volt dan arus yang dihasilkan tergantung dari intensitas cahaya yang masuk pada photodiode tersebut. Teknik ini biasa disebut sebagai ‘*unbiased current sourcing*’ atau ‘*photovoltaic mode*’. Teknik ini jarang digunakan karena tidak efisien dan mempunyai respon yang lambat terhadap pulsa-pulsa cepat sinyal cahaya.

Konfigurasi *photodiode* yang umum dipakai adalah teknik yang dikenal sebagai ‘*reversed biased*’. Pada mode *reverse bias*/bias terbalik, *photodiode* dibias dengan tegangan external mulai dari beberapa volt sampai sekitar 50 volt (tergantung karakteristik *photodiode*). Jika karakteristik *photodiode* tidak diketahui maka bias tegangan dapat diberi 12V agar tidak merusak *photodiode* tersebut.

Ketika *photodiode* ini mendapat cahaya, dalam hal ini cahaya *infra red* maka terdapat arus bocor yang relatif kecil. Besar-kecilnya arus bocor ini tergantung dari intensitas cahaya *infra red* yang mengenai *photodiode* tersebut. Gambar 2.21 di bawah menunjukkan rangkaian penerima *photodiode*.



**Gambar 2.21.** Fotodiode

Sumber: Malvino, 1985 : 98

Dengan  $R_1$  didapatkan dari persamaan sebagai berikut:

$$V_{CC} = V_{fd} + I_{R1} R_1 \dots\dots\dots (2-18)$$

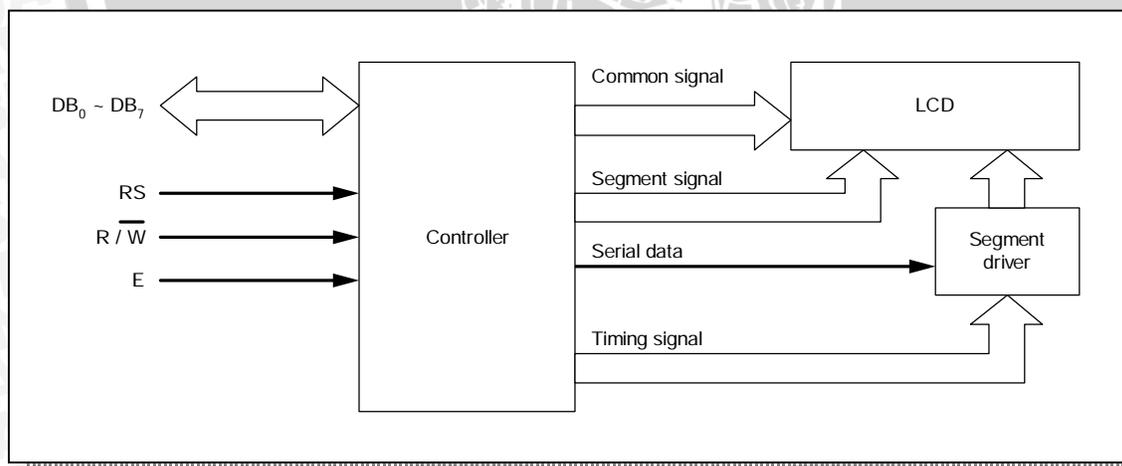
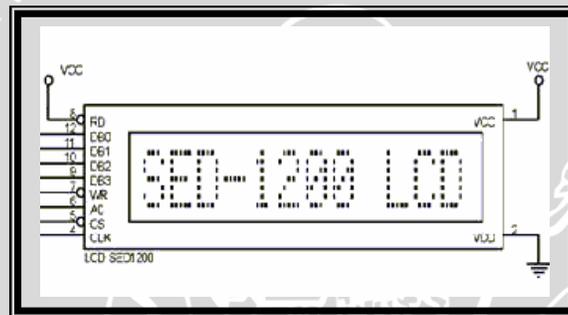
$$R_1 = \frac{V_{CC} - V_{fd}}{I_{R1}} \dots\dots\dots (2-19).$$

**2.5. Modul LCD M1632**

*Liquid Crystal Display* (LCD) adalah suatu media penampil dalam bentuk matrik titik. LCD M1632 ini diproduksi oleh Seiko Instruments Inc. Dalam pembuatan alat ini, digunakan dua buah LCD yang terletak pada pintu utama dan pintu masuk areal parkir. Pada pintu utama, LCD digunakan sebagai penampil mengenai kondisi parkir apakah penuh atau tidak. Sedangkan pada pintu masuk, LCD digunakan sebagai *display* untuk menampilkan karakter huruf yang dipicu dari penekanan tombol *keypad*.

LCD M1632 adalah sebuah modul “*Dot-Matrix Liquid Crystal Display*” yang membutuhkan daya kecil, dan dilengkapi dengan panel LCD dengan tingkat kontras yang cukup tinggi serta pengontrol LCD CMOS yang telah terpasang dalam modul tersebut. Pengontrol mempunyai ROM/RAM pembangkit karakter, dan RAM data *display*. Semua fungsi untuk tampilan diatur menggunakan instruksi-instruksi, sehingga modul LCD ini dengan mudah dapat diantarmukakan dengan unit mikroprosesor atau mikrokontroler [Seiko, 1987: 1].

Masukan yang diperlukan untuk mengendalikan modul ini berupa bus data yang masih termultipleks dengan bus alamat serta 3 bit sinyal kontrol. Sementara pengendalian dot matrik LCD dilakukan secara internal oleh kontroler yang sudah terpasang pada modul LCD. Diagram blok untuk bagian penampil terlihat dalam Gambar 2.22.



Gambar 2.22. LCD (Liquid Crystal Display)

Gambar 2.23. Diagram blok modul LCD M1632.

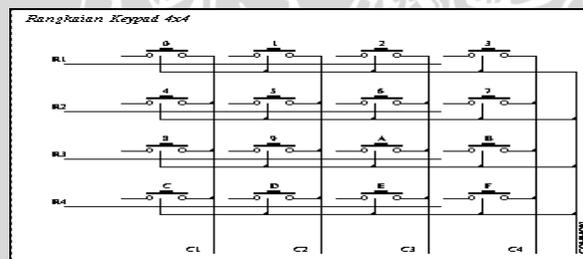
Sumber : Seiko, 1987: 3

LCD modul M1632 mempunyai spesifikasi sebagai berikut:

- § Memiliki 16 karakter dan 2 baris tampilan yang terdiri dari 5 X 7 dot matrik ditambah dengan kursor.
- § Memerlukan catu daya +5 Volt.
- § Otomatis reset saat catu daya dinyalakan.
- § 80 X 8 display data RAM (max 80 karakter).

## 2.6. Keypad / Papan Tombol

Tombol masukan merupakan rangkaian yang digunakan untuk memberikan data masukan, yang mana data tersebut diberikan melalui penekanan tombol yang terdapat pada papan masukan itu sendiri. Ketika tombol masukan ditekan, rangkaian enkoder akan menerjemahkan penekanan pada salah satu tombol tersebut ke dalam bentuk biner. Sehingga jumlah jalur yang akan masuk ke rangkaian berikutnya dapat diperkecil. Kode-kode biner ini yang akan dikirimkan ke unit pengolah. Keypad untuk memungkinkan pengunjung memilih tingkat lantai parkir yang tersedia.



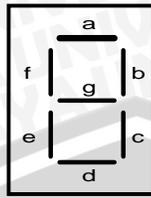
**Gambar 2.24.**

Rangkaian Keypad 4x4

## 2.7. Peraga Seven Segmen

Decoder/Driver BCD ke tujuh seven segmen dan peraga tujuh segmen digunakan untuk mengambil input BCD ke 4 bit dan menghasilkan output-output yang akan melewati arus melalui segmen-segmen sesuai untuk menampilkan digit desimalnya. Pada model penampil yang terlihat pada Gambar 2.25 digunakan tujuh garis (*segment*) yang disusun sedemikian rupa sehingga dengan menyalakan garis-garis tertentu akan terbentuk angka desimal. Dalam Gambar 2.25, tiap garis (*segment*) diberi tanda tersendiri dengan simbol a,b,c,d,e,f,g, setiap garis terbuat dari LED yang apabila diberi tegangan

akan menyala berbentuk garis. LED adalah dioda yang dapat memancarkan cahaya jika diberi suatu energi. Energi yang dimaksud adalah energi listrik.



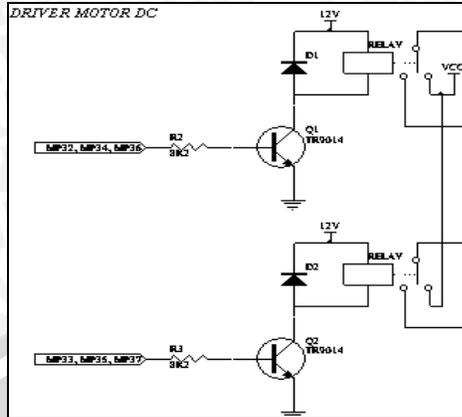
**Gambar 2.25.** Penampil Seven Segmen

Apabila diinginkan untuk menyalakan LED-LED tersebut dengan data 4 bit, maka diperlukan sebuah dekoder/*driver* BCD tujuh segmen. Dekoder ini akan menerima data 4 bit dan memberikan keluaran arus yang akan menyalakan LED-LED tersebut.

Suatu tabel kebenaran dapat disusun dengan empat bit (4 bit dari kode input BCD) dan tujuh output (a,b,c,d,e,f,g) dan kemudian kombinasi bit-bit tersebut dapat digunakan untuk menentukan kasus-kasus dalam tabel di mana setiap output seharusnya tinggi atau rendah. Untuk kombinasi input yang tak terpakai seperti 1010 sampai 1111 dapat diperlukan sedemikian rupa sehingga menghasilkan keluaran segmen-segmen tidak aktif (*blank display*) atau untuk memilih pola-pola segmen non angka pada beberapa decoder/driver tertentu.

### 2.8. Driver Relay

*Driver* yang digunakan untuk menggerakkan relai yang direncanakan dengan menggunakan transistor yang difungsikan sebagai saklar. Apabila ada tegangan pada basis transistor tersebut, maka transistor akan *on* dan akan mengaktifkan relai. Apabila tidak ada tegangan pada basis transistor, maka transistor akan dalam kondisi *off* sehingga relai tidak aktif. *Driver relay* menggunakan sistem DPDT (*Dual pole dual terminal*) yakni akan timbul beda potensial positif maupun negatif pada kutub motor DC bila *relay* dialiri arus. Rangkaian driver relay dapat dilihat dalam Gambar 2.26.



**Gambar 2.26.**

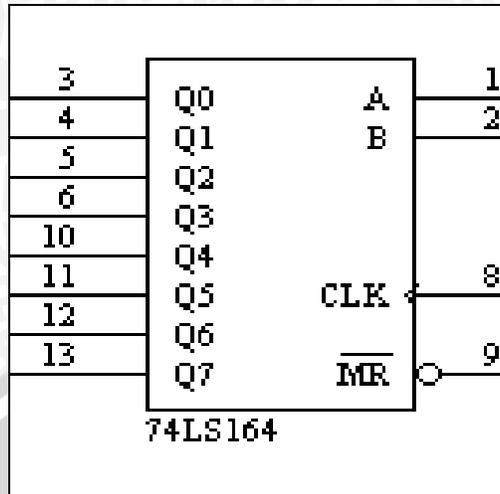
Rangkaian Driver Relay

## 2.9. Enkoder

Enkoder disini berfungsi sebagai port tambahan yang menghubungkan seluruh sensor pada dua lantai pada mikrokontoller. Enkoder adalah rangkaian logika kombinasi untuk mengkodekan suatu sinyal masukan menjadi keluaran berupa kode biner.

Enkoder disusun dari gerbang-gerbang logika yang menghasilkan keluaran biner sebagai hasil tanggapan adanya dua atau lebih variabel masukan. Hasil keluarannya dinyatakan sebagai aljabar boolean tergantung pad kombinasi gerbang-gerbang yang digunakan. Sebuah Enkoder harus memenuhi syarat perancangan. Satu kombinasi masukan hanya boleh mewakili satu kombinasi biner. Dalam hal ini enkoder digunakan untuk mengkodekan 20 kombinasi masukan dari sensor, dimana 10 buah kombinasi masukan diperlukan 4 bit keluaran.

Untuk membuat suatu enkoder dapat menggunakan IC enkoder yang fleksibel untuk penerjemahan sinyal diskrit dengan variabel masukan yang banyak semisal 32 masukan yakni IC 74LS164. Bagan IC 74LS164 dapat dilihat dalam Gambar 2.27.

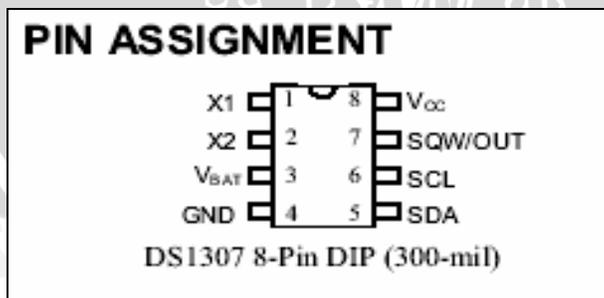


Gambar 2.27.

Bagan IC LS74164

### 2.10 Rangkaian RTC

Rangkaian RTC (*Real Time Clock*) dalam rangkaian ini digunakan untuk mensetting waktu parkir kendaraan, apakah perhari atau perjam, namun dalam simulasinya digunakan per hari. RTC yang digunakan adalah RTC DALLAS DS1307 yang dapat menghitung dan menyimpan data dalam satuan detik, menit, jam, hari, bulan, maupun tahun. RTC juga dilengkapi dengan catu daya back-up yang memungkinkan sistem pewaktuan tetap berjalan walaupun catu daya sistem dimatikan. Konfigurasi pin RTC 1307 dapat dilihat pada Gambar 2.28.

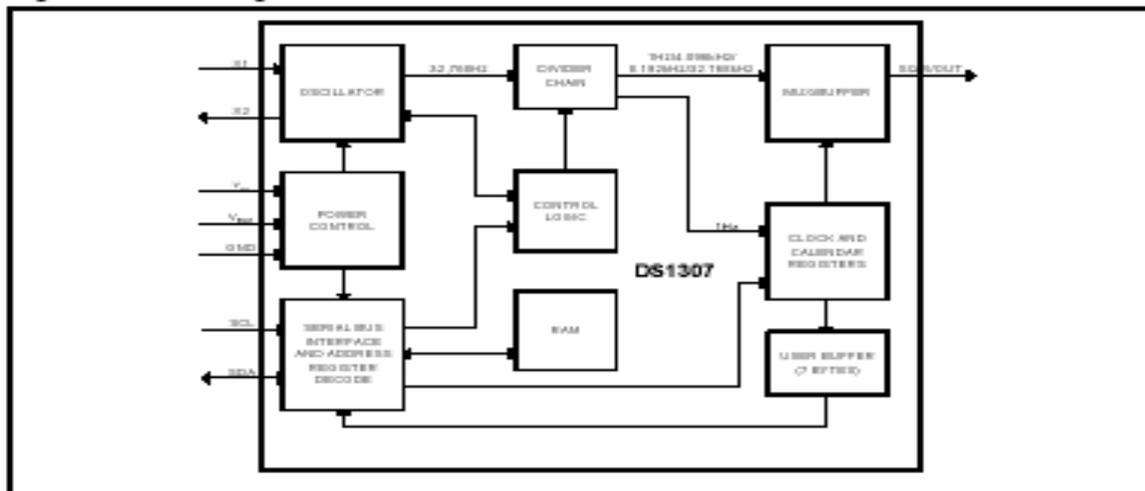


Gambar 2.28. konfigurasi pin RTC 1307

Pendesripsian pin sebagai berikut :

- Ø SCL – Serial Clock Input, digunakan untuk sinkronisasi perpindahan data pada komunikasi data yang diinterfacekan secara serial.
- Ø SDA – Serial Data Input/Output, merupakan input atau output untuk *2-wire serial interface*.
- Ø SQW/OUT – Saat kondisi aktif, bit SQWE di set hingga 1, pin keluaran dari SQW/OUT adalah satu dari empat kali gelombang frekuensi.
- Ø X1,X2 – Merupakan koneksi standar untuk kristal 32,768 kHz. Rangkaian osilator internal didesain untuk operasi dengan sebuah kristal yang memiliki beban kapasitansi tertentu, yakni sebesar 12,5 pF.
- Ø Vcc - Adalah tegangan masukan +5V, ketika Vcc diaplikasikan pada kondisi normal, RTC terakses penuh dan dapat melakukan proses baca tulis.
- Ø VBat – Tegangan batere harus tetap berada diantara 2V dan 3,5V untuk pengoperasian RTC yang berfungsi mem-back up RTC dalam waktu tertentu.

Arsitektur dalam RTC DS1307 dapat dilihat dalam Gambar sebagai berikut.

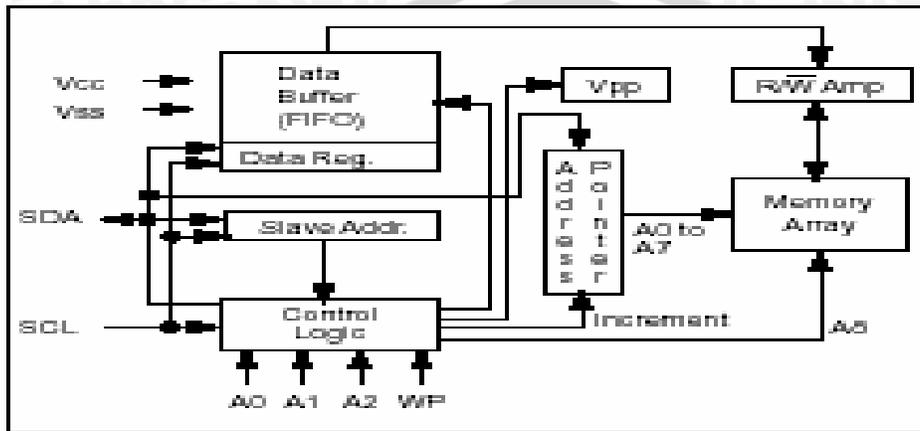


**Gambar 2.30.** Arsitektur dalam RTC DS1307

Dalam pengoperasiannya, RTC beroperasi sebagai slave pada serial bus. Akses awal dimulai dari kondisi start dan identifikasi kode dengan alamat register hingga kondisi stop. Ketika Vcc jatuh diantara  $1,25 \times V_{BAT}$ , alamat counter di treset. Ketika Vcc jatuh di bawah VBAT, maka alat dihubungkan dengan *low current battery back up mode*.

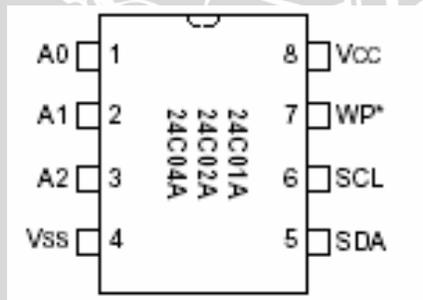
### 2.11 Rangkaian EEPROM

Dalam pembuatan alat ini, EEPROM yang digunakan adalah 24C01A 1k bit serial EEPROM dengan standar pengkabelan dua *serial interface*. Berikut adalah blok diagram rangkaian EEPROM 24C01A.



Gambar 2.31. Arsitektur dalam EEPROM 24C01A

Pin EEPROM dapat dilihat dalam Gambar 2.32.



Gambar 2.32. Pin EEPROM 24C01A

Fungsi dari masing-masing pintersebut adalah :

- Ø A0, dihubungkan dengan Vcc atau Vss.
- Ø A0, A1, A2, merupakan chip alamat masukan.
- Ø Vss, berfungsi sebagai ground.
- Ø SDA, merupakan alamat serial atau data I/O.
- Ø SCL, merupakan serial clock.

- Ø WP, untuk *Write Protect input*.
- Ø Vcc, merupakan *power supply 5V*.

Karakteristik dari EEPROM dapat dilihat dalam tabel 2.11.

**Tabel 2.11.** DC Karakteristik EEPROM 24C01A

VCC = +5V (±10%)		Commercial (C): Tamb = 0°C to +70°C Industrial (I): Tamb = -40°C to +85°C Automotive (E): Tamb = -40°C to +125°C			
Parameter	Symbol	Min.	Max.	Units	Conditions
VCC detector threshold	VTH	2.8	4.5	V	
<b>SCL and SDA pins:</b>					
High level input voltage	VIH	VCC x 0.7	VCC + 1	V	IOL = 3.2 mA (SDA only)
Low level input voltage	VL	-0.3	VCC x 0.3	V	
Low level output voltage	VOL		0.4	V	
<b>A1 &amp; A2 pins:</b>					
High level input voltage	VIH	VCC - 0.5	VCC + 0.5	V	
Low level input voltage	VL	-0.3	0.5	V	
Input leakage current	ILI	—	10	µA	VIN = 0V to VCC
Output leakage current	ILO	—	10	µA	VOUT = 0V to VCC
Pin capacitance (all inputs/outputs)	CIN, COUT	—	7.0	pF	VIN/VOUT = 0V (Note) Tamb = +25°C, f = 1 MHz
Operating current	ICC Write	—	3.5	mA	FCLK = 100 kHz, program cycle time = 1 ms, Vcc = 5V, Tamb = 0°C to +70°C
	ICC Write	—	4.25	mA	FCLK = 100 kHz, program cycle time = 1 ms, Vcc = 5V, Tamb = (I) and (E)
	ICC Read	—	750	µA	VCC = 5V, Tamb = (C), (I) and (E)
Standby current	ICCS	—	100	µA	SDA=SCL=VCC=5V (no PROGRAM active)

Sumber : datasheet

## BAB III

### METODE PENELITIAN

Penyusunan skripsi ini didasarkan pada masalah yang bersifat aplikatif yang diwujudkan dalam bentuk prototipe, yaitu perencanaan dan perealisasiannya agar dapat menampilkan unjuk kerja sesuai dengan yang direncanakan dengan mengacu pada rumusan masalah. Data dan spesifikasi komponen yang digunakan dalam perencanaan merupakan data sekunder yang diambil dari buku data komponen elektronika. Pemilihan komponen berdasarkan perencanaan dan disesuaikan dengan komponen yang ada di pasaran.

Langkah-langkah yang perlu dilakukan untuk mengaplikasikan alat yang akan dibuat, secara umum adalah sebagai berikut:

- Studi Literatur
- Penentuan spesifikasi alat
- Perealisasiannya
- Pengujian alat
- Analisa data dan pengambilan kesimpulan

#### 3.1 Studi Literatur

Literatur yang dibutuhkan adalah dasar teori yang berhubungan, yaitu sebagai berikut :

- a. Sistem Mikrokontroler R8C Tiny 13 sebagai pengontrol utama.
- b. Sistem mikrokontroler AT89S51 sebagai port tambahan.
- c. Rangkaian sensor mobil yang terdiri dari LED *infrared* dan *Photodiode*, dan rangkaian pendukungnya.
- d. Rangkaian komunikasi serial antar Mikrokontroler
- e. Rangkaian LCD M-1632 dan rangkaian pendukungnya.
- f. Rangkaian keypad 4x4.
- g. Motor DC, prinsip kerja dan rangkaian pengaturannya.
- h. Rangkaian RTC dan EEPROM tambahan pada mikrokontroler.

### 3.2 Penentuan Spesifikasi Alat

Sebelum melakukan perencanaan dan perealisasiian alat, maka ditentukan spesifikasi alat yang akan dibuat. Adapun spesifikasi alat yang akan direalisasiikan sebagai berikut:

1. Penggerak *prototipe* palang pintu parkir menggunakan *driver relay* dan motor DC *gearbox*.
2. Pengontrol utama sistem menggunakan Mikrokontroler R8C Tiny 13, terutama untuk rangkaian keypad 4x4, LCD 2 (untuk tampilan pengisian data keypad), 12 pasang sensor (kombinasi LED *Infrared* dan *photodiode*), serta komunikasi serial ke MK AT89S51.
3. Port tambahan menggunakan Mikrokontroler AT89S51, terutama untuk rangkaian printer, Seven segment, kendali motor palang pintu, LCD 1 (untuk tampilan loby depan), serta komunikasi serial ke MK R8C Tiny 13.
4. Sensor mobil menggunakan kombinasi LED dan *photiododa* dengan LED sebagai pemancar yang diletakkan di daerah palang pintu 1, palang pintu 2, dan di di gerang masuk dan keluar lantai 2, dengan *photiododa* yang digunakan sebagai penerima diletakkan di seberang LED *Infrared* sepanjang lebar badan mobil.
5. Rangkaian RTC untuk *mensetting* waktu kendaraan masuk dan keluar.
6. EEPROM sebagai penyimpan data tambahan.

### 3.3 Perealisasiian Alat

#### 3.3.1 Perancangan Perangkat Keras dan Perealisasiian Tiap Blok

- Pembuatan blok diagram lengkap sistem.
- Penentuan dan perhitungan komponen yang akan digunakan
- Merakit perangkat keras masing-masing blok.

#### 3.3.2 Perancangan dan Perealisasiian Perangkat Lunak

Setelah kita mengetahui seperti apa perangkat keras yang kita rancang untuk aplikasi sistem parkir ini, maka kita membutuhkan perangkat lunak untuk mengendalikan dan mengatur kerja dari alat ini. Program yang dibuat dalam bentuk bahasa C yang

disimpan dalam EEPROM mikrokontroler R8C Tiny 13 dan disimpan dalam *EEPROM* mikrokontroler AT89S51.

### **3.4 Pengujian Alat.**

Untuk memastikan bahwa sistem ini berjalan sesuai yang direncanakan maka perlu dilakukan pengujian alat meliputi perangkat keras (*hardware*) maupun perangkat lunak (*software*) yang dilakukan baik secara per blok rangkaian maupun keseluruhan sistem.

#### **3.4.1 Pengujian Perangkat Keras**

Pengujian perangkat keras dilakukan dengan tujuan untuk menyesuaikan nilai tegangan dan arus pada alat dengan cara pengukuran yang dibandingkan dengan nilai tegangan dan arus yang diijinkan bekerja dalam komponen alat berdasarkan data sekunder komponen.

#### **3.4.2 Pengujian Perangkat Lunak**

Pengujian perangkat lunak dilakukan dengan melakukan pengujian pada subrutin-subrutin program dengan memberi masukan dan mengamati keluarannya, maupun pada rangkaian yang telah dirakit secara lengkap.

#### **3.4.3 Pengujian Keseluruhan Sistem**

Tujuan pengujian sistem secara keseluruhan adalah untuk mengetahui unjuk kerja alat setelah perangkat keras dan perangkat lunak diintegrasikan bersama.

### **3.5 Analisa Data dan Pengambilan Kesimpulan**

Kesimpulan didapat berdasarkan dari hasil perealisasiian dan pengujian keseluruhan sistem, berfungsinya alat secara otomatis sejak mobil masuk hingga keluar areal parkir pada prototipe, serta pengendalian kecepatan motor DC yang digunakan sebagai penggerak palang pintu otomatis pada areal parkir.

## BAB IV

### PERANCANGAN ALAT

Pada bab ini akan dibahas tentang perancangan dan pembuatan miniatur sistem parkir otomatis berbasis mikrokontroler R8C Tiny 13, sebagai pengendali utama sistem, serta mikrokontroler AT89S51 sebagai port tambahan. Perancangan sistem ini meliputi perangkat keras dan perangkat lunak. Untuk perangkat kerasnya adalah sistem mikrokontroler R8C Tiny 13 yang komunikasi datanya diinterfacekan secara serial dengan mikrokontroler AT89S51.

Dalam pembuatan alat ini digunakan dua jenis mikrokontroler, yakni satu buah mikrokontroler R8C Tiny 13 sebagai pengontrol sistem serta dua buah mikrokontroler AT89S51 yang difungsikan sebagai port tambahan. Banyaknya port tambahan yang dibutuhkan oleh komponen-komponen pendukung alat tersebut, seperti sepuluh pasang sensor mobil, keypad, dua buah LCD, dan tiga buah motor DC, menjadikan AT89S51 sebagai alternatif paling mudah untuk menjadi port tambahan yang komunikasi datanya diinterfacekan secara serial dengan mikrokontroler R8C Tiny 13 yang memiliki dua port serial, yakni TXD<sub>10</sub> dan TXD<sub>11</sub>.

Data yang dikirim mikrokontroler sensor mobil akan diproses mikrokontroler utama, kemudian data tersebut digunakan sebagai perintah pada LCD yang telah diprogram dengan karakter huruf tertentu. Keypad digunakan untuk mengisi data pada LCD. Data dari keypad diproses oleh mikrokontroler R8C Tiny 13 dimana data tersebut memicu driver relay untuk menggerakkan motor DC baik forward atau reverse. Setelah mengetahui rancangan perangkat kerasnya, maka diperlukan perangkat lunak untuk mengendalikannya melalui mikrokontroler. Bahasa pemrograman yang digunakan dalam perancangan ini adalah bahasa C untuk mikrokontroler R8C Tiny 13 dan mikrokontroler AT89S51.

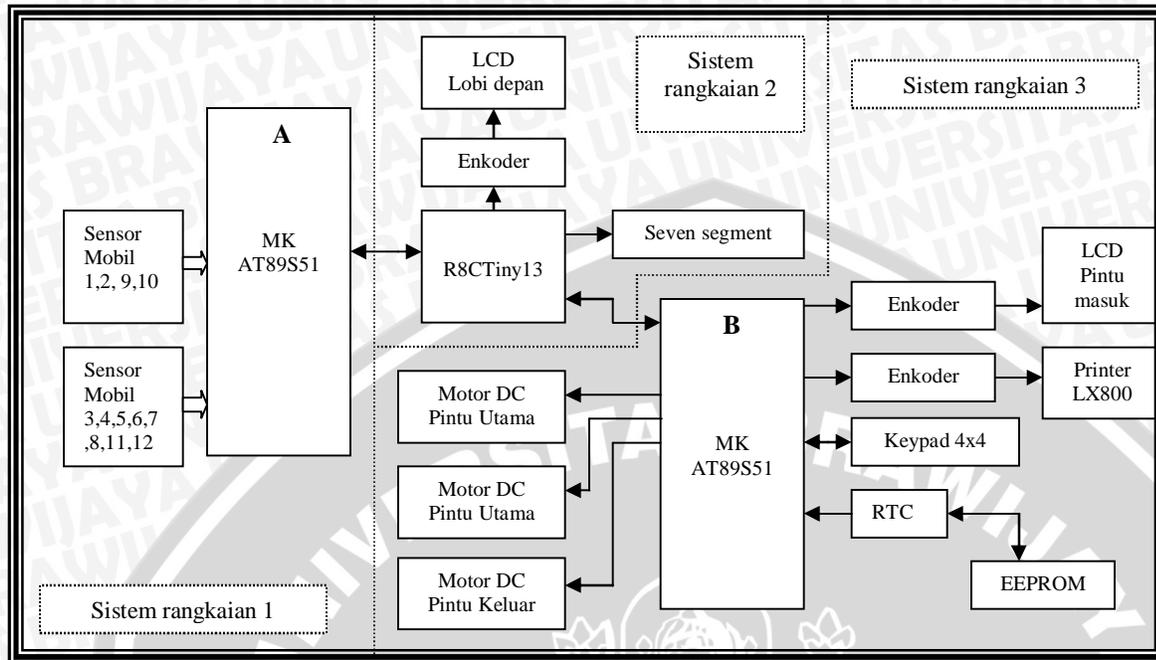
#### 4.1 Penentuan spesifikasi alat

Dalam perancangan sistem parkir otomatis ini spesifikasi yang akan direalisasikan adalah sebagai berikut:

- Miniatur yang terbuat dari kayu dengan dimensi 60cmx25cmx10cm sebagai miniatur dari areal parkir dua tingkat.
- Satu buah sistem mikrokontroler R8C Tiny 13 sebagai kontrol sistem.
- Dua buah sistem Mikrokontroler AT89S51 sebagai port tambahan.
- Dua buah LCD, yakni LCD 16 kolom x 2 baris untuk informasi penuh tidaknya areal parkir pada lobi depan, dan LCD pada operator sebagai layar yang menampilkan data dengan keypad.
- Enam buah *driver relay* yang digunakan untuk menggerakkan motor DC membuka dan menutup palang pintu.
- Tiga buah motor DC *gearbox*.
- Sepuluh pasang sensor mobil yang terdiri dari LED *Infrared* dan fotodioda pada lantai 1 dan 2.
- *Keypad* untuk mengisi *password* dan nomor polisi kendaraan saat akan masuk dan keluar areal parkir. Pada simulasinya akan diisikan nomor urut yang berdasarkan kuantitas areal parkir dengan batas tampung 20 mobil.
- Rangkaian RTC dan EEPROM untuk menyimpan data kendaraan yang masuk.
- Rangkaian enkoder sebagai port tambahan untuk printer.
- Rangkaian *seven segment* untuk menunjukkan kapasitas lantai parkir.

#### 4.2 Pembuatan alat

Agar perancangan dan pembuatan alat dapat dilakukan sistematis dan terstruktur maka perlu dibuat blok diagram yang menjelaskan sistem yang dirancang. Blok diagram sistem ditunjukkan dalam Gambar 4.1



**Gambar 4.1** Blok diagram sistem  
 Sumber : Perancangan

Sesuai dengan blok diagram dalam Gambar 4.1 tersebut maka peralatan yang dirancang antara lain:

- ✚ Unit Mikrokontroler Renesas R8C Tiny 13 sebagai pengendali utama system dimana pin-pinnya dihubungkan dengan LCD, *seven segment*, dan dua buah mikrokontroler AT89S51.
- ✚ Unit mikrokontroler AT89S51 pada system rangkaian 1 pin-pinnya dihubungkan dengan sensor mobil, RTC, dan mikrokontroler R8C Tiny 13. Fungsi mikrokontroler AT89S51 sebagai port tambahan yang di-*interface serial* dengan pengendali utama sistem.
- ✚ Unit mikrokontroler AT89S51 pada system rangkaian 3 pin-pinnya dihubungkan dengan rangkaian *driver relay*, LCD, printer, *keypad*, dan mikrokontroler R8C Tiny 13.
- ✚ Tampilan menggunakan LCD 16 kolom x 2 baris dengan karakter tipe M1632.
- ✚ Sensor mobil yang dipasang pada port mikrokontroler AT89S51 berfungsi mendeteksi mobil dari ukuran panjangnya, akan berfungsi bila badan

mobil tepat mengenai kedua sensor yang dipasang terpisah sesuai panjang mobil.

- ✚ *Seven segment* berfungsi memberi informasi kapasitas ruang parkir yang tersedia pada tiap lantai.
- ✚ *Keypad* berfungsi menuliskan *password* dan nomor polisi untuk membuka palang pintu pada pintu masuk dan pintu keluar areal parkir.
- ✚ *Display* atau LCD 16M32 pada AT89S51 berfungsi menampilkan data yang diisikan melalui keypad.
- ✚ Driver motor DC digunakan untuk menggerakkan relai kemudian relai itu akan menggerakkan motor DC, apakah posisi naik atau turun, pada palang pintu utama, palang pintu masuk, dan palang pintu keluar.
- ✚ Enkoder 74LS164 berfungsi sebagai port tambahan yang dipasang pada AT89S51 untuk printer.

### 4.3 Prinsip kerja alat

Saat mobil akan masuk areal parkir, pengemudi melihat tampilan LCD pada lobi depan apakah parkir penuh atau tidak. Jika parkir tidak penuh maka pada LCD akan muncul tulisan “Selamat datang, silakan memasuki areal parkir”. Sistem parkir ini dirancang dengan kapasitas tempat parkir maksimal untuk dua puluh mobil, dimana jika kapasitas parkir kurang dari dua puluh mobil, maka palang pintu utama akan selalu terbuka.

Mobil melewati palang pintu utama dan menuju palang pintu masuk, Sebelumnya, mobil tepat berhenti pada satu sensor mobil yang terdiri dari LED dan fotodioda dan dapat melihat tampilan pada seven segment untuk memilih lantai parkir. Jika sinar yang dipancarkan LED ke Fotodioda terhalang maka logika “0”, sehingga sensor memberi sinyal ke mikrokontroler, kemudian mikrokontroler memberi perintah pada *display* untuk menampilkan tulisan, sehingga pengemudi mengetikkan nomor polisi dan password pada *display*, dalam hal ini disimulasikan dengan mengetikkan nomor urut parkir, dan data dari pengemudi kemudian disimpan pada EEPROM yang terletak pada RTC untuk mencatat jam awal dan perhitungan biaya parkir. Data tersebut diproses pada mikrokontroler sehingga memicu motor DC membuka palang pintu masuk.

Setelah palang pintu masuk terbuka, mobil masuk dan melewati dua sensor mobil yang terpisah sepanjang badan mobil. Jika mobil tepat secara bersamaan berada pada kedua sensor tersebut maka palang pintu masuk akan tertutup dan counter pada mikrokontroler berkurang satu sehingga tampilan pada seven segment lantai 1 berubah, mobil dianggap parkir pada lantai 1. Jika mobil ingin parkir di lantai dua, maka mobil langsung naik ke lantai dua dan menginjak sensor mobil pada lantai tersebut. Maka program counter pada mikrokontroler berkurang satu pada lantai dua, dan bertambah satu pada lantai satu, sehingga tampilan *seven segment* pun berubah.

Jika mobil keluar dari lantai dua, akan melewati sensor mobil dan secara otomatis counter pada lantai dua akan berkurang satu setelah mobil tepat berada pada dua sensor mobil, dan counter pada lantai satu bertambah satu. Setelah turun, tepat sebelum palang pintu keluar, mobil akan melewati dan berhenti pada sensor mobil yang ada, maka sensor memberi sinyal pada mikrokontroler untuk memberi perintah pada *display* agar pengemudi mengetikkan kembali nomor polisi dan password seperti pada awal masuk. Kondisi diabaikan jika mobil mundur. Simulasi data yang diketikkan berupa nomor urut masuk.

Data yang diketikkan pada saat akan melewati pintu keluar kemudian disesuaikan dengan data awal yang tersimpan pada EEPROM, jika benar, maka akan tertera lama parkir terhitung sejak entry data pada saat masuk dan untuk pembayaran dikalikan dengan rupiah. Waktu acuan yang digunakan tergantung pihak mall, bisa perjam, bisa juga untuk sekali parkir. Sedang tarif perjam atau perhari bisa disesuaikan dengan standar tarif parkir yang ada. Namun pada simulasinya, tarif parkir dihitung per hari.

Setelah tertera jumlah tagihan pada *display*, maka pengemudi menekan enter, kemudian data tersebut memicu mikrokontroler memberi perintah pada printer untuk mencetak karcis yang berisi data pengemudi. Kemudian dilakukan pembayaran yang ditangani oleh pihak operator, jika sudah membayar, maka operator akan menekan opsi enter pada *keypad*, dan data tersebut memicu mikrokontroler untuk memberi perintah pada motor DC yang menggerakkan palang pintu keluar agar bergerak ke atas (terbuka).

Kemudian mobil keluar dan melewati satu sensor mobil, maka counter pada mikrokontroler di lantai satu berkurang satu, dan tampilan pada *seven segment* berubah.

Jika kapasitas parkir yang tersedia penuh, maka data dari counter memicu mikrokontroler memberi perintah pada LCD 1 dengan tampilan “Parkir penuh” dan motor DC agar menutup palang pintu utama.

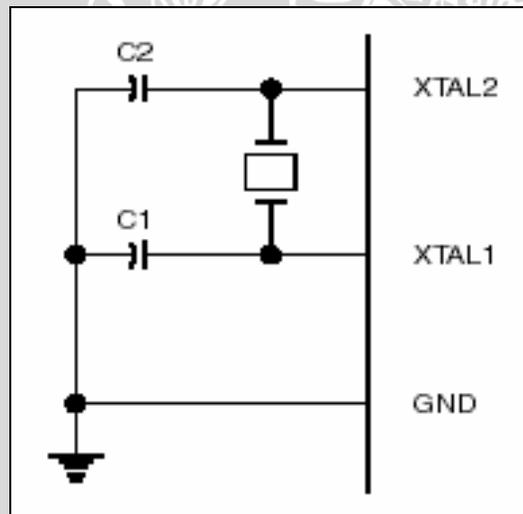
#### 4.4 Perancangan perangkat keras

Perencanaan masing-masing blok diagram sistem terdiri dari perencanaan rangkaian kontrol menggunakan R8C Tiny 13, AT89C51, dan beberapa rangkaian pendukung seperti rangkaian *display*, *seven segment*, RTC, rangkaian printer, *keypad*, rangkaian *driver* relai, dan rangkaian sensor mobil.

##### 4.4.1 Rangkaian kontrol menggunakan AT89S51

###### 4.4.1.1 Pengaturan *baud rate*

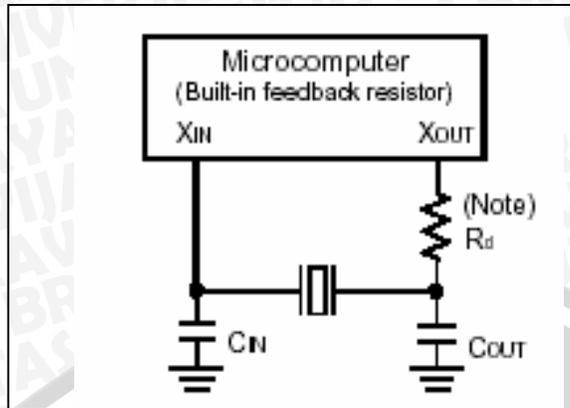
Di dalam proses komunikasi serial antara mikrokontroler R8C Tiny 13 dengan AT89S51 dan rangkaian lainnya terlebih dahulu ditentukan *baud rate* yang digunakan. Pada sistem ini digunakan *baud rate* sebesar 1200 bps dengan menggunakan  $f_{osc} = 11,0592$  MHz karena tidak memerlukan kecepatan transfer data yang tinggi.



Gambar 4.2.a. Rangkaian Osilator

Sumber: Datasheet AT89C51

Rangkaian osilator mikrokontroler R8C Tiny 13 dapat dilihat dalam Gambar 4.2.b.



Gambar 4.2.b. Rangkaian Osilator  
Sumber: Datasheet R8C Tiny 13

Dari persamaan (2-3) dapat dihitung nilai pada register TH1:

$$\text{Baud Rate} = \frac{f_{osc}}{12 \times (256 - TH_1) \times 16}$$

$$1200 = \frac{11,0592 \text{ Mhz}}{12 \times (256 - TH_1) \times 16}$$

$$230400 = \frac{11,0592 \text{ Mhz}}{256 - TH_1}$$

$$256 - TH_1 = \frac{11,0592 \times 10^6}{230400}$$

$$TH_1 = 256 - 48$$

$$= 208$$

$$= D0H$$

#### 4.4.1.2 Rangkaian Reset

Untuk me-reset mikrokontroler AT89S51, maka pin RST diberi logika tinggi selama sekurangnya dua siklus mesin (24 periode osilator). Untuk membangkitkan sinyal reset kapasitor dihubungkan dengan  $V_{CC}$  dan sebuah resistor yang dihubungkan ke ground.

Karena kristal yang digunakan mempunyai frekuensi sebesar 11,0592 MHz, maka satu periode dapat dihitung dari persamaan (2-5):

$$T = \frac{1}{f_{XTAL}} = \frac{1}{11,0592 \text{ MHz}} s = 9,042 \times 10^{-8} s$$

Sehingga waktu minimal logika tinggi yang dibutuhkan untuk mereset mikrokontroler dapat dihitung menggunakan persamaan (2-6) :

$$\begin{aligned} t_{\text{reset(min)}} &= T \times \text{periode yang dibutuhkan} \\ &= 9,042 \times 10^{-8} \times 24 \\ &= 2,17 \mu s \end{aligned}$$

Jadi mikrokontroler membutuhkan waktu minimal 2,17  $\mu s$  untuk mereset. Waktu minimal inilah yang dijadikan pedoman untuk menentukan nilai R dan C. Dari Persamaan (2-4) dengan menentukan nilai R = 10 k $\Omega$ , dan C = 10  $\mu F$ , maka:

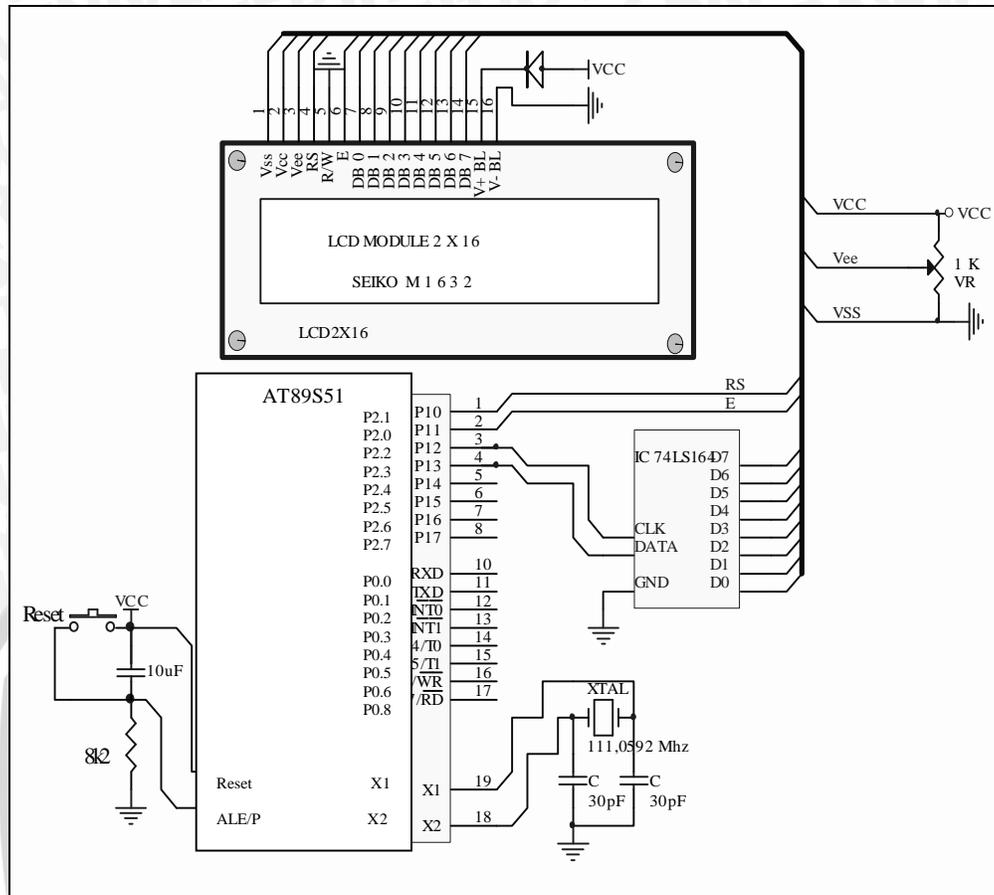
$$t = 0,357 R.C = 0,357 \times 10000 \Omega \times 10 \cdot 10^{-6} = 35,7 \text{ ms}$$

Jadi dengan nilai komponen R = 10k $\Omega$ , dan C = 10  $\mu F$  dapat memenuhi syarat minimal untuk waktu yang dibutuhkan oleh mikrokontroler.

Sedangkan untuk me-*reset* mikrokontroler R8C Tiny 13, maka pin RST diberi logika rendah selama sekurangnya dua siklus mesin (24 periode osilator). Untuk membangkitkan sinyal *reset* kapasitor dihubungkan dengan *ground* dan sebuah resistor yang dihubungkan ke Vcc. Mikrokontroler membutuhkan waktu minimal 500 $\mu s$  untuk me-*reset*. Waktu minimal tersebut yang digunakan untuk menentukan nilai R dan C.

#### 4.4.2 Perancangan Karakter Huruf pada Liquid Crystal Display (LCD)

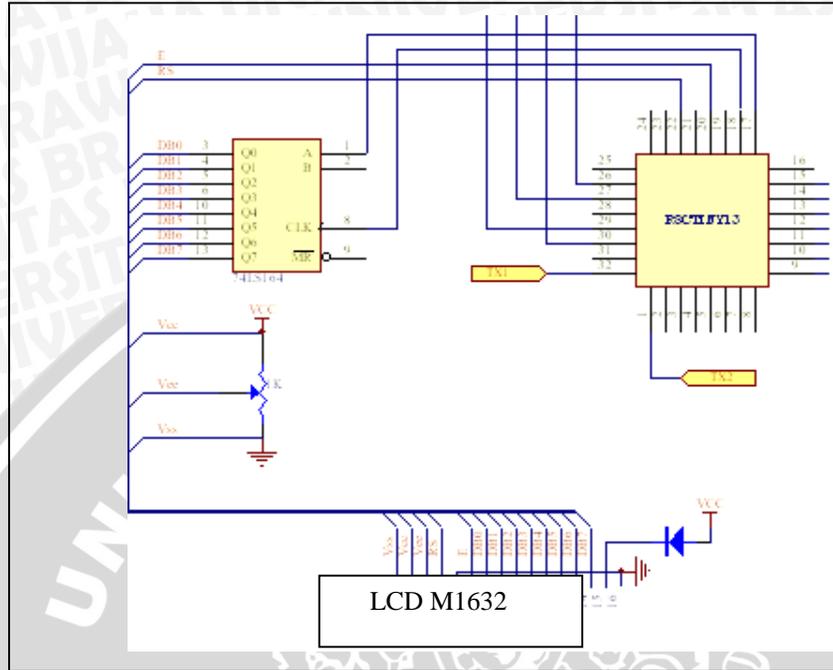
Dalam perancangan alat ini LCD digunakan sebagai sarana informasi bagi pengemudi mengenai kapasitas lantai parkir. LCD tersebut juga digunakan sebagai *display* tempat pengemudi memasukkan data ketika masuk dan keluar areal parkir. aplikasi ini menggunakan sebuah layar LCD (*Liquid Crystal Display*) yaitu jenis *Seiko Instrument M1632* yang merupakan LCD dua baris dengan setiap barisnya terdiri 16 karakter dan menggunakan IC 74LS164 yang merupakan register geser 8 bit yang memiliki jalan masuk deret pergerbang. Gambar hubungan antara LCD, IC74LS164, dan mikrokontroler dapat dilihat dalam Gambar 4.3 dan 4.4.



**Gambar 4.3** Antarmuka LCD dengan mikrokontroler AT89C51

Dalam perancangan ini digunakan dua buah LCD, yakni masing-masing LCD M1632 yang terhubung dengan mikrokontroler R8C Tiny 13 dan mikrokontroler AT89S51. LCD dot matrik ini membutuhkan sepuluh buah pin masukan/keluaran dari mikrokontroler dan IC 74164. Pada Mikrokontroler R8C Tiny 13, dua buah pin yakni port 3.0 pada pin RS yang digunakan sebagai sinyal pemilih register dan port 3.1 pada pin Enable digunakan sebagai sinyal operasi awal, sinyal enable ini mengaktifkan data tulis atau baca oleh mikrokontroler, pin DB0-DB7 yang dihubungkan ke pin data IC74164 digunakan untuk menampilkan karakter yang dikehendaki oleh mikrokontroler. Ketika terdapat data pada jalur data, data tersebut akan ditahan dengan memberikan *clock* pin E pada LCD. Pin RS menentukan apakah data yang ditahan akan digunakan sebagai instruksi untuk mengatur *setting* tampilan pada LCD atau sebagai kode karakter yang diperlukan LCD untuk menampilkan suatu karakter. Sedangkan untuk pin R/W pada LCD dihubungkan ke *ground* karena dalam hal ini LCD hanya melakukan operasi write

atau operasi menampilkan karakter. Pada mikrokontroler AT89S51, pin RS terhubung pada port 2.1 dan pin Enable terhubung pada port 2.0.



**Gambar 4.4.** Antarmuka LCD dengan mikrokontroler R8C Tiny 13

Untuk pin Vcc pada LCD dihubungkan ke supply +Vcc dan Vss dihubungkan ke ground. Pin V<sub>EE</sub> beserta pin Vcc dan Vss dihubungkan ke *trimer potensio* atau kadang disebut dengan *trimpot*. *Trimpot* ini digunakan untuk mengatur kontras dari tampilan LCD dengan cara mengubah tegangan pada pin V<sub>EE</sub>. Daftar tabel fungsi pin pada LCD dapat dilihat dalam Tabel 4.1.

**Tabel 4.1.** Fungsi pin LCD

Pin	Fungsi
DB0 – DB7	Merupakan saluran data, berisi perintah dan data yang akan ditampilkan di LCD.
Enable	Sinyal operasi awal, sinyal ini mengaktifkan data tulis atau baca.
R/W	Sinyal seleksi tulis atau baca 0: tulis 1: baca
RS	Sinyal pemilih <i>register</i> 0: masukan data 1: masukan instruksi

Sumber: EL-TECH Electronics; LCD Modul User Manual 1987: 2

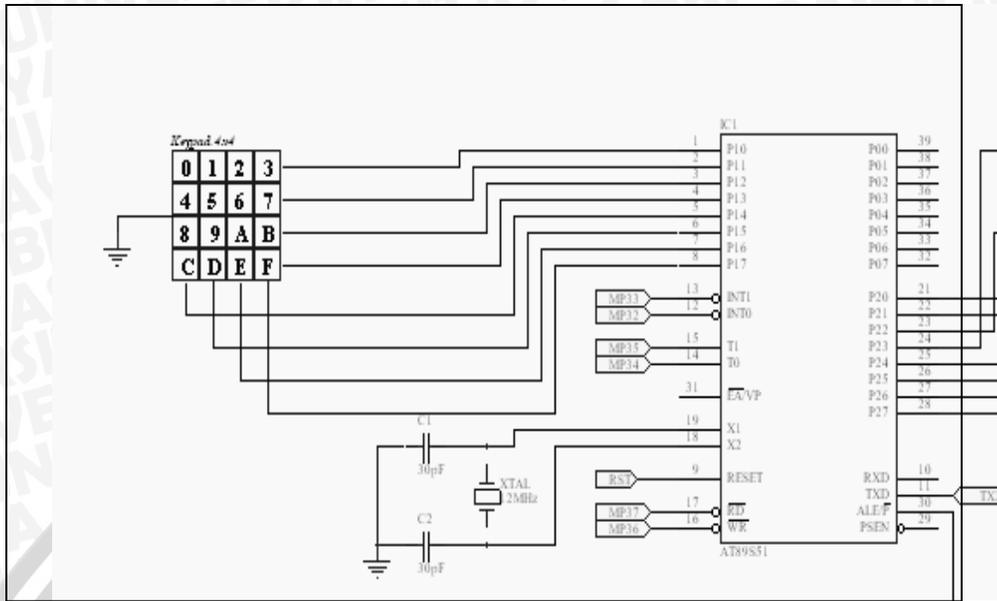
Seperti telah disebutkan sebelumnya bahwa data yang terdapat pada jalur data selain dianggap sebagai kode karakter dapat digunakan sebagai suatu perintah instruksi untuk mengatur setting dari tampilan LCD. Cara pemakaian data antara sebagai instruksi dengan kode karakter berbeda. Perbedaan hanyalah keadaan pin RS ketika data yang ada di jalur data ditahan oleh LCD dengan memberikan *clock* pada pin E.

#### 4.4.3 Rangkaian tombol masukan (*Keypad*)

*Keypad* sering digunakan sebagai media masukan dalam berbagai aplikasi elektronik, seperti halnya pada perancangan ini. Data pada *keypad* akan diproses pada mikrokontroler yang kemudian digunakan untuk menggerakkan motor DC. Seperti yang sudah dijelaskan pada sub bab sebelumnya bahwa *keypad* ini bekerja dengan metode matrik dimana keluaran yang dihasilkan merupakan kombinasi dari baris dan kolom dari tombol yang ditekan.

Pada bagian belakang dari *keypad* terdapat delapan buah pin yang berfungsi sebagai keluaran dari *keypad* tersebut. Delapan buah pin ini terdiri dari 4 buah pin yang menunjukkan kolom yang terhubung dengan port 1.4-1.7 dari tombol yang ditekan dan 4 buah pin menunjukkan baris dari tombol yang ditekan yang terhubung dengan port 1.0-1.3 pada mikrokontroler AT89S51.

Masalah-masalah yang sering timbul dengan digunakannya *keypad* jenis ini adalah adanya *bouncing*. Dengan adanya *bouncing* maka tombol yang ditekan sekali, Akan terdeteksi berulang kali. *Bouncing* ini dapat diatasi dalam pemrograman dengan memberikan jeda waktu atau delay beberapa saat sebelum data benar-benar dimasukkan ke dalam mikrokontroler untuk diolah. Tujuan pemberian jeda waktu ini untuk memberi kesempatan agar benar-benar terjadi kontak antar ujung baris dan kolom dari tombol yang ditekan dengan resistor pull up. Gambar antarmuka *keypad* matrik 4x4 dengan mikrokontroler AT89S51 dapat dilihat dalam Gambar 4.5.

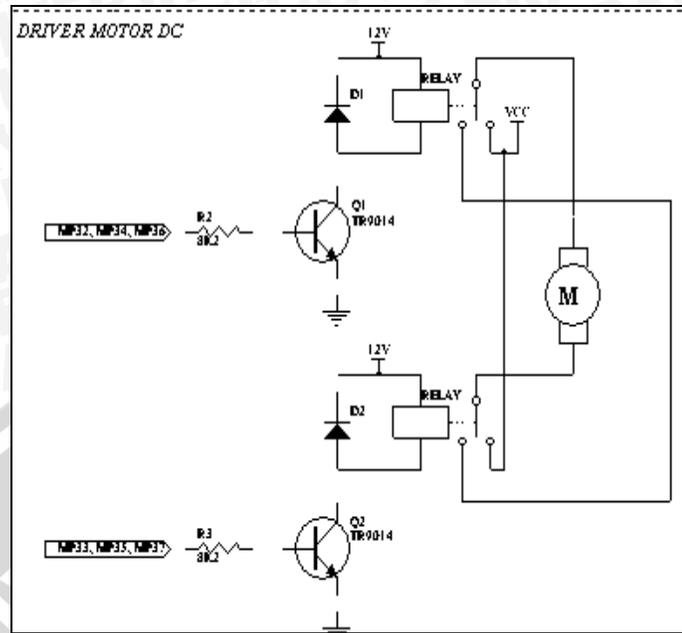


**Gambar 4.4** Antarmuka rangkaian *keypad* dengan mikrokontroler AT89C51

Prosedur untuk pembacaan *keypad* dilakukan dengan teknik *compare* atau membandingkan data. Data yang dimasukkan melalui *keypad* dibandingkan dengan data yang tersimpan di dalam mikrokontroler. Apabila ada yang sesuai maka mikrokontroler akan melakukan instruksi sesuai dengan data yang dimasukkan dari *keypad*. Data yang sah, yang akan diolah oleh mikrokontroler adalah data yang diambil setelah diadakan pemanggilan sub program *delay* untuk mengatasi *bouncing*. Data tersebut kemudian dibandingkan dengan data yang ada di mikrokontroler dan apabila ada yang sesuai maka mikrokontroler akan memproses data tersebut. Apabila data pada akumulator tidak ada yang sesuai dengan data pada mikrokontroler maka akan diulang kembali proses pemasukan data dari *port 2* ke akumulator.

**4.4.4 Rangkaian driver relay**

Rangkaian driver yang digunakan untuk menggerakkan relai yang terhubung dengan motor DC terdiri dari transistor NPN KTC9014 yang difungsikan sebagai transistor *switching*, relai yang dihubungkan dengan motor DC. Rangkaian *driver relay* dapat dilihat dalam Gambar 4.6.



Gambar 4.6 Rangkaian driver relay

Rangkaian *driver* berfungsi untuk menggerakkan relai dan mengaktifkan motor DC. Tegangan keluaran dari mikrokontroler dihubungkan dengan *driver* relai sehingga relai akan aktif dan dapat mengatur gerak motor DC. Perhitungan dari rangkaian *driver* solenoid dalam Gambar 4.5 terlebih dahulu harus dicari nilai  $R_b$ . Data yang diperlukan untuk mencari besar resistansi  $R_b$  adalah sebagai berikut:

Data transistor KTC9014 yang diperoleh dari *datasheet* (KEC, 1994) adalah :

- Besar pengukuran tahanan dalam relai ( $R_{relai} = R_c$ ) = 800  $\Omega$ .
- $V_{ce}$  saturasi = 0,25 volt.
- $H_{fe}$  minimum = 200
- $H_{fe}$  maximum = 1000.
- $V_{be}$  = 0,7 volt.

Dengan resistansi relai sebesar 800  $\Omega$ , tegangan catu sebesar 12 V, dan  $V_{ce}$  saturasi sebesar 0,25 V maka dengan menggunakan Persamaan (2-14) besar arus  $I_{relai}$  adalah :

$$I_{relai} = \frac{V_{cc} - V_{ce}}{R_{relai}}$$

$$I_{relai} = \frac{12v - 0,25v}{800\Omega}$$

$$= 14.68 \text{ mA}$$

Dalam prancangan ini digunakan  $hfe$  minimal 60, maka dapat digunakan  $hfe$  200 dengan pertimbangan transistor tetap mampu menggerakkan relai dengan penguatan yang dipakai adalah optimal, maka dengan menggunakan persamaan (2-15):

$$I_B = \frac{I_c}{hfe}$$

$$I_B = \frac{14,68mA}{200}$$

$$= 0,0734 \text{ mA}$$

Jika  $V_{bb}$  adalah tegangan keluaran dari mikrokontroler sa at logika tinggi yaitu sebesar 2,4V dan  $V_{be} = 0,7 \text{ V}$  maka dengan menggunakan persamaan (2-16) besar resistansi  $R_B$  adalah :

$$R_B = \frac{V_{bb} - V_{be}}{I_B}$$

$$R_B = \frac{2,4v - 0,7v}{0,0734mA}$$

$$= 23,160K\Omega \cong 22K\Omega$$

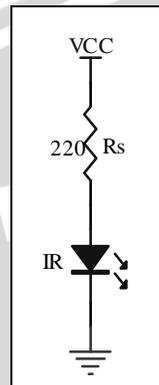
Dengan resistansi  $R_B$  sebesar  $22K\Omega$ , arus  $I_B$  menjadi 0,077mA sehingga telah diperoleh arus  $I_B$  yang dibutuhkan.

#### 4.4.5 Rangkaian Sensor Inframerah

Rangkaian sensor inframerah terdiri dari *transmitter* dan *receiver*, yakni LED *infrared* dan *fotodiode*. Dalam perancangan alat, sensor tersebut digunakan untuk mendeteksi mobil yang masuk pada areal parkir. Berikut penjelasan mengenai rangkaian sensor inframerah. Rangkaian sensor ini disusun sebanyak 12 pasang sensor dimana masing-masing pinnya dihubungkan pada port 2.0 hingga port 2.7 dan port 3.4 hingga port 3.7 pada mikrokontroler AT89S51.

#### 4.4.5.1 Rangkaian Pemancar (*Transmitter*)

Rangkaian pemancar berfungsi untuk memancarkan sinar *infra red* dengan bantuan LED yang memancarkan *infra red* (IRED), yang nantinya diterima oleh rangkaian penerima *infra red*. Gambar 4.7 menunjukkan rangkaian pemancar *infra red*.



Gambar 4.7 Rangkaian pemancar infrared

Mengacu dalam Gambar rangkaian di atas, agar dioda infra merah dapat memancarkan sinar infra merah diperlukan arus sebesar 20 mA dan tegangan ( $V_{IR}$ ) sebesar 1,2 Volt. Sesuai dengan persamaan (2-17) dapat dihitung besarnya  $R_s$  sebagai berikut :

$$R_s = \frac{V_{CC} - V_{IR}}{I}$$

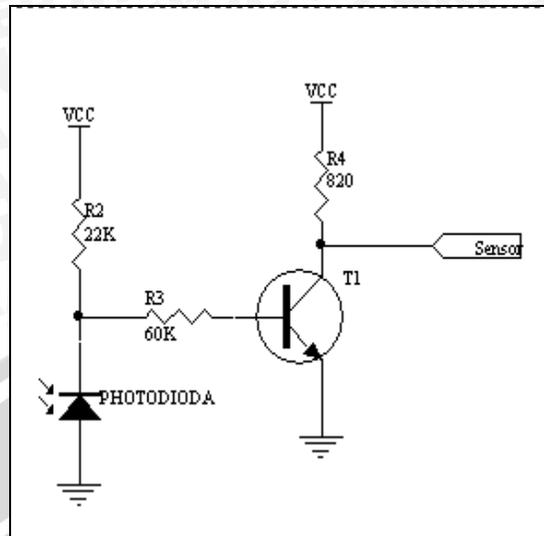
$$R_s = \frac{5 - 1,2}{20 \times 10^{-3}}$$

$$R_s = \frac{3,8}{20 \times 10^{-3}}$$

Jadi resistor yang terpasang pada rangkaian pemancar adalah 220  $\Omega$ .

#### 4.4.5.2 Rangkaian Penerima (*Receiver*)

Rangkaian penerima berfungsi untuk menerima sinyal *infra red* dari pemancar *infra red*, sinyal *infra red* diterima oleh foto dioda dan digunakan untuk membias transistor yang difungsikan sebagai saklar (*switching*). Output dari rangkaian ini berupa tegangan. Gambar 4.8 menunjukkan rangkaian penerima *infra red*.



Gambar 4.8 Rangkaian penerima infra red

Karena transistor dioperasikan untuk saklar (*switching*) maka dioperasikan pada dua titik kerja, yaitu : pada titik saturasi atau pada titik sumbat (*cut off*). Jadi perhitungan nilai resistor yang terpasang pada rangkaian penerima sinyal *infra red* adalah sebagai berikut:

§ **Kondisi Pertama, pada saat IR LED tidak terhalang benda**

Pada saat IR LED RX tidak terhalang dilakukan pengukuran untuk melihat arus. Arus itu memiliki nilai  $I_{R1}=0,12mA$ , Maka menurut persamaan (2-18):

$$V_{CC} = V_{fd} + I_{R1}R2$$

Untuk rangkaian ini didesain  $V_{fd}$  adalah 2,4V

$$5 = 2,4 + I_{R1}R2$$

Kemudian sesuai dengan persamaan (2-17):

$$R2 = \frac{V_{CC} - V_{fd}}{I_{R1}}$$

$$R2 = \frac{5 - 2,4}{0,12mA}$$

$$R2 = 21666\Omega \cong 22K\Omega$$

Ditentukan besar  $R_4 = 820 \Omega$  dan  $V_{CE} = 2,7$ . Maka menurut persamaan (2-9) dan

$R_4 = R_C$ , nilai  $I_C$  adalah:

$$I_C \cdot R_C + V_{CE} - V_{CC} = 0$$

$$I_C = \frac{V_{CC} - V_{CE}}{R_C}$$

$$I_C = \frac{5 - 2,7}{820}$$

$$I_C = 2,8 \times 10^{-3} A$$

Dengan  $h_{fe}$  minimum 60 maka dalam perancangan dapat digunakan  $h_{fe}$  100 agar transistor berkerja optimal dengan menggunakan persamaan (2-13) maka besar arus basis:

$$I_B = \frac{I_C}{h_{fe}}$$

$$I_B = \frac{2,8 \times 10^{-3} A}{100}$$

$$= 0,028 mA$$

Kemudian mencari nilai  $R_2 = R_B$ , sesuai dengan persamaan (2-12) yaitu :

$$I_B = \frac{V_{BB} - V_{BE}}{R_B}$$

$$V_{BB} = V_{fd}$$

$$V_{fd} = I_B R_2 + V_{BE}$$

$$I_B R_2 = V_{fd} - V_{BE}$$

$$R_2 = \frac{V_{fd} - V_{BE}}{I_B}$$

$$R_2 = \frac{2,4 - 0,7}{0,028 mA}$$

$$R_2 = 60,71 K\Omega$$

§ **Kondisi kedua , pada saat IR LED terhalang benda**

Sesuai dengan persamaan (2-12) yaitu

$$I_B = \frac{V_{BB} - V_{BE}}{R_B}$$

$$V_{CC} = V_{BB}, R_B = R_1 + R_2$$

$$I_B = \frac{V_{CC} - V_{BE}}{R_B}$$

$$V_{CC} = I_B R_B + V_{BE}$$

$$V_{CC} = I R_1 + I R_2 + V_{BE}$$

$$V_{CC} = I R_1 + I R_2 + V_{BE}$$

$$V_{CC} = I(R_1 + R_2) + V_{BE}$$

$$V_{CC} = I(21,6K\Omega + 1,36K\Omega) + V_{BE}$$

$$I = I_B = \frac{V_{CC} - V_{BE}}{(21,6K\Omega + 60,71K\Omega)}$$

$$I = I_B = \frac{5 - 0,7}{(82,31K\Omega)}$$

$$I = I_B = 0,052mA$$

Sesuai dengan persamaan (2-15) maka didapatkan nilai  $I_C$  :

$$I_C = I_B \cdot h_{fe}$$

$$I_C = 0,052mA \times 100$$

$$I_C = 5,2mA$$

Maka nilai  $V_{CE}$  (nilai masukan ke mikrokontroler) menurut persamaan (2-13) adalah:

$$I_C \cdot R_C + V_{CE} - V_{CC} = 0$$

$$R_C = R_3$$

$$V_{CC} = I_C R_3 + V_{CE}$$

$$V_{CE} = V_{CC} - I_C \cdot R_3$$

$$V_{CE} = 5 - 5,2 \times 10^{-3} A \times 820\Omega$$

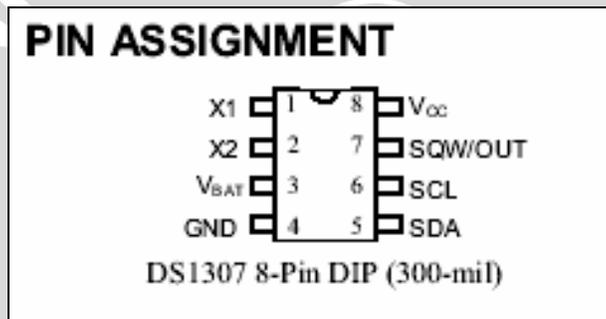
$$V_{CE} = 5 - 4,264V$$

$$V_{CE} = 0,736V$$

Rangkaian pemancar (*transmitter*) dan penerima (*receiver*) di atas digunakan untuk sensor mobil

#### 4.4.6 Rangkaian RTC

Rangkaian RTC (real Time Clock) dalam rangkaian ini digunakan untuk *mensetting* waktu parkir kendaraan, apakah perhari atau perjam, namun dalam simulasinya digunakan per hari. RTC yang digunakan adalah RTC DALLAS DS1307 yang dapat menghitung dan menyimpan data dalam satuan detik, menit, jam, hari, bulan, maupun tahun. RTC juga dilengkapi dengan catu daya back-up yang memungkinkan sistem pewaktuan tetap berjalan walaupun catu daya sistem dimatikan.



Gambar 4.8 konfigurasi pin RTC 1307

Pendeskrripsian pin sebagai berikut :

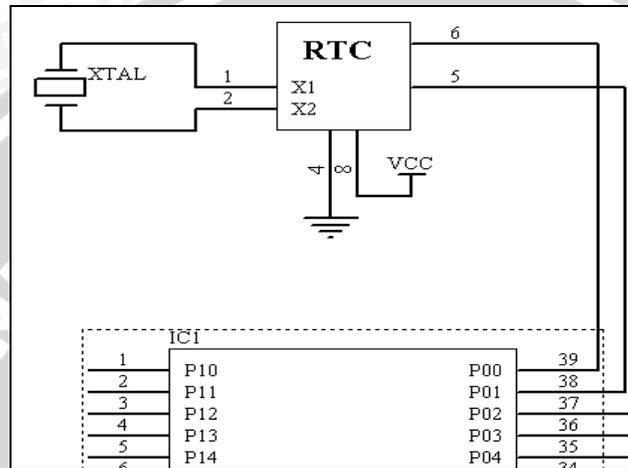
- Ø SCL – Serial Clock Input, digunakan untuk sinkronisasi perpindahan data pada komunikasi data yang diinterfacekan secara serial.
- Ø SDA – Serial Data Input/Output, merupakan input atau output untuk *2-wire serial interface*.
- Ø SQW/OUT – Saat kondisi aktif, bit SQWE di set hingga 1, pin keluaran dari SQW/OUT adalah satu dari empat kali gelombang frekuensi.
- Ø X1,X2 – Merupakan koneksi standar untuk kristal 32,768 kHz. Rangkaian osilator internal didesain untuk operasi dengan sebuah kristal yang memiliki beban kapasitansi tertentu, yakni sebesar 12,5 pF

Alamat dan data ditransfer secara serial melalui dua rangkaian, *bi-directional bus*.

Akses dapat dilakukan dengan mengimplementasikan kondisi start pada suatu rangkaian

dan memberikan kode identifikasi alat dengan alamat register. Akses tidak akan terhenti hingga kondisi STOP, yakni *setting* waktu yang ditentukan. Blok diagram di bawah mendeskripsikan rangkaian utama pada RTC.

Dalam pembuatan alat ini, pin pada RTC dihubungkan dengan pin 0.0 dan pin 0.1 pada mikrokontroler AT89S51.

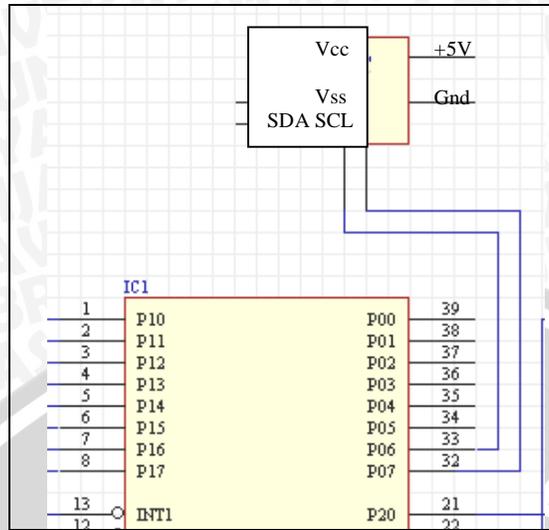


**Gambar 4.9**

Rangkaian antarmuka RTC dengan mikrokontroler AT89S51

#### 4.4.8 Rangkaian EEPROM

EEPROM yang digunakan adalah EEPROM 24C01A, dalam perancangan alat ini berfungsi untuk menyimpan setting waktu dan jumlah kendaraan pada areal parkir. Pin SDA dan SCL pada EEPROM dihubungkan serial dengan mikrokontroler AT89S51 pada sistem rangkaian 3 dalam Gambar 4.1 pada pin 0.6 dan pin 0.7.

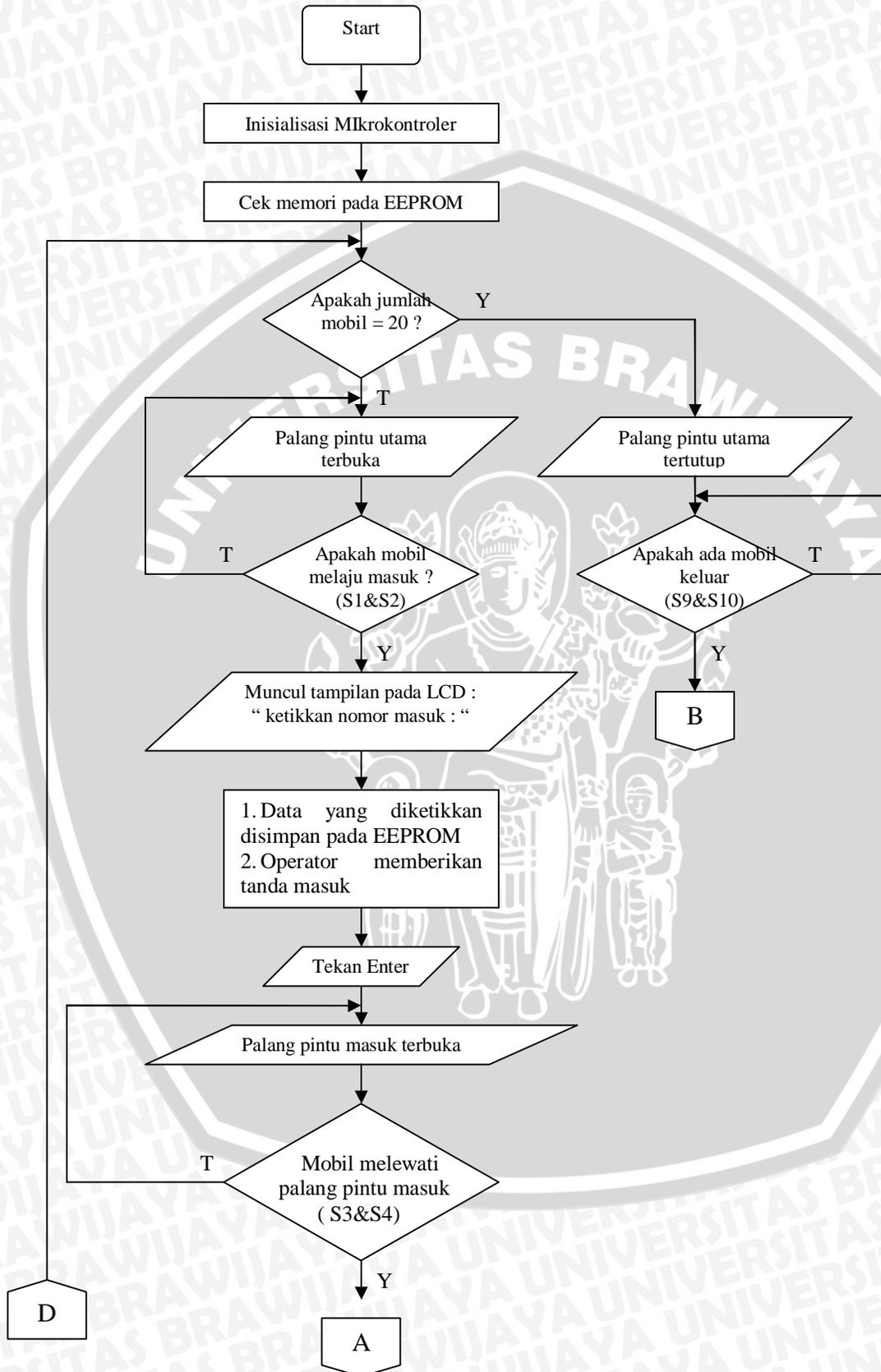


Gambar 4.10

Rangkaian antarmuka EEPROM dengan mikrokontroler AT89S51

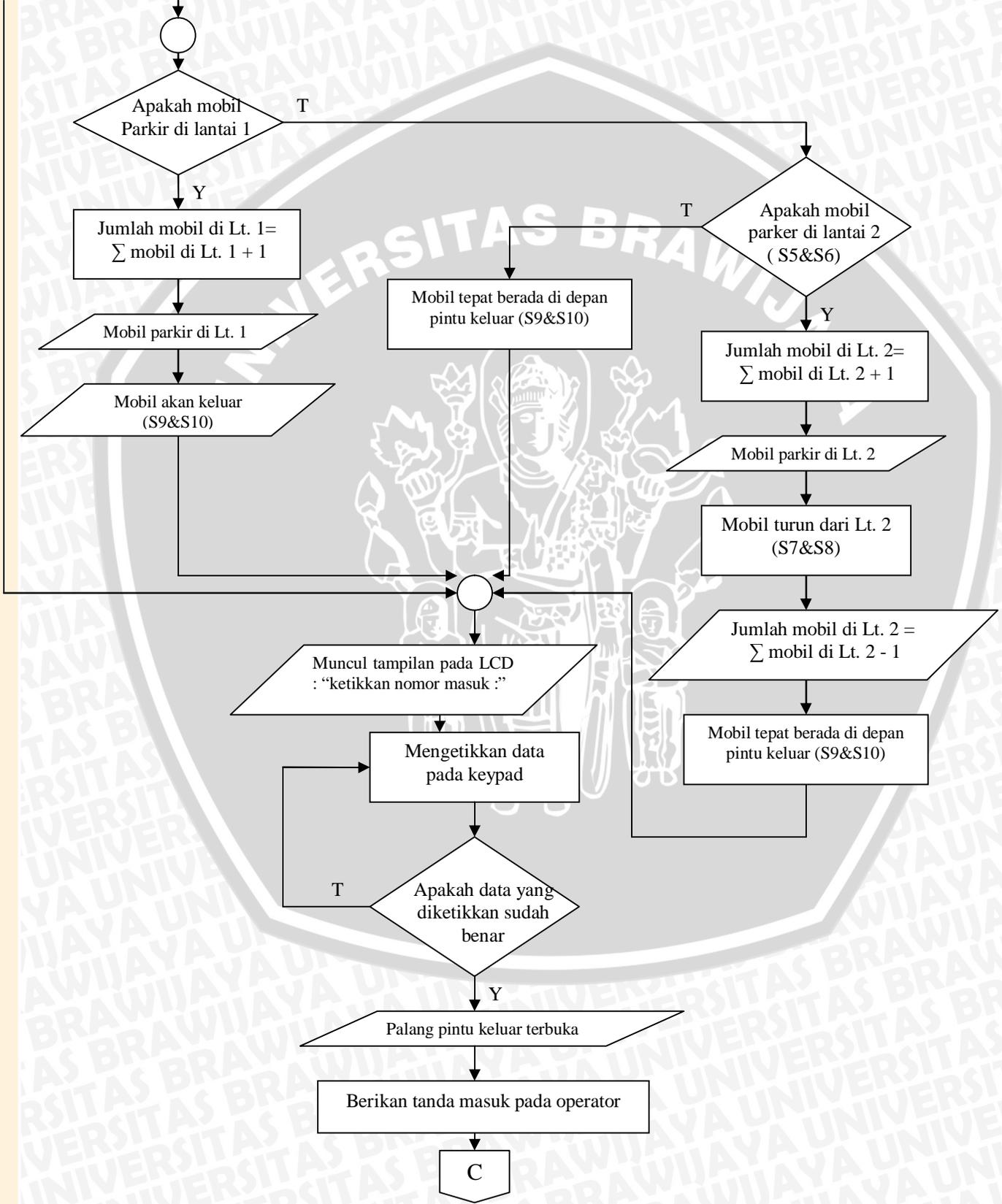


### 4.5 Flowchart Keseluruhan Sistem

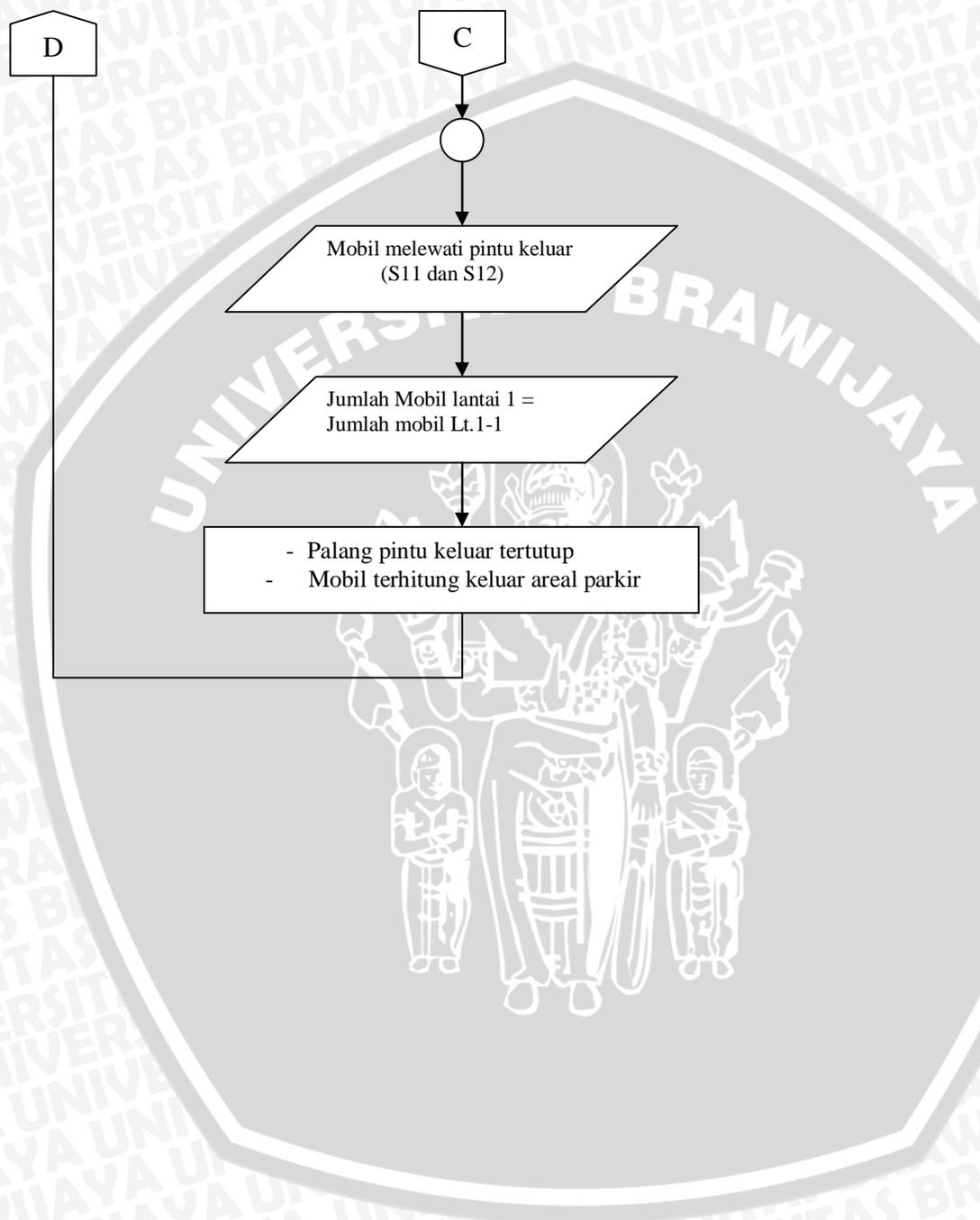


B

A



C



## BAB V

### PENGUJIAN DAN ANALISIS

Pengujian miniatur sistem parkir otomatis berbasis mikrokontroler R8C Tiny 13 dan AT89S51 meliputi pengujian perangkat keras dan perangkat lunak sistem. Pengujian dilakukan perbagian untuk mempermudah dalam menganalisis hasil perancangan dan pengujian yang dilakukan. Bagian-bagian yang diuji adalah:

1. Pengujian rangkaian *driver relay*
2. Pengujian rangkaian sensor mobil
4. Pengujian LCD
5. Pengujian unit masukan *keypad*
6. Pengujian RTC

Setelah semua bagian di atas diuji, langkah berikutnya adalah pengujian sistem secara keseluruhan.

Alat-alat bantu yang digunakan dalam pengujian adalah sebagai berikut:

1. Multimeter digital
2. Catu daya 5 volt dan 12 volt
3. *Logic Probe*

Bab ini membahas pengujian alat yang telah dirancang dan direalisasikan. Pengujian dilakukan untuk mengetahui bahwa alat yang dirancang telah bekerja sesuai dengan spesifikasi yang diinginkan.

#### 5.1 Pengujian rangkaian *Driver Relay*

##### 5.1.1 Tujuan

Pengujian ini bertujuan untuk mengetahui apakah motor DC bekerja dengan baik dan sesuai dengan yang direncanakan.

##### 5.1.2 Peralatan yang digunakan

1. Multimeter digital
2. Catu daya 12V
3. *Logic Probe*

### 5.1.3 Prosedur Pengujian

1. Merancang rangkaian seperti dalam Gambar 5.1
2. Menjalankan rangkaian pengujian
3. Amati keluaran dari kondisi *relay* dan *output relay*



Gambar 5.1. rangkaian pengujian *Driver Relay*

### 5.1.4 Hasil Pengujian dan Analisis

Ketika diberi logika tinggi ("1"), maka *relay* akan 'on' dan palang pintu bergerak, jika sebaliknya maka relay 'off', palang pintu tidak bergerak.

Hasil Pengujian rangkaian *driver relay* ditunjukkan dalam Tabel 5.1.

Logika Tinggi ("1")	Relai on	Palang pintu bergerak	12 volt
Logika Rendah ("0")	Relai off	Palang pintu tidak bergerak	0 volt

Tabel 5.1 Hasil Pengujian Rangkaian *Driver relay* Motor DC

Dari hasil pengujian diketahui bahwa pengaturan motor DC tergantung dari *driver relay*. Hal tersebut menandakan bahwa rangkaian *driver relay* dapat bekerja sesuai yang diharapkan.

## 5.2 Pengujian rangkaian sensor mobil

### 5.2.1 Tujuan

Pengujian ini bertujuan untuk mengetahui apakah kerja sensor mobil sesuai perencanaan

### 5.2.2 Peralatan yang digunakan

1. *Power supply*
2. LED
3. *Photodiode*
4. *Logic probe*

### 5.2.3 Prosedur pengujian

1. Menghubungkan rangkaian seperti ditunjukkan dalam Gambar 5.2
2. Memberikan tegangan catu 5 volt
3. Memberikan keadaan *photodiode* menerima cahaya dan keadaan tidak menerima cahaya
4. Mengamati perubahan logika menggunakan *logic probe*



Gambar 5.2 Blok diagram pengujian sensor mobil

### 5.2.4 Hasil Pengujian dan Analisis

Jika *Photodiode* menerima cahaya maka logika *high* (“1”) dan sebaliknya jika *photodiode* tidak menerima cahaya maka berlogika *Low* (“0”).

Keluaran logika pada keadaan yang diubah-ubah pada *photodiode* ditunjukkan dalam Tabel 5.2.

Keadaan <i>Photodiode</i>	Logika
<b>Menerima cahaya</b>	<b>High (1)</b>
<b>Tidak menerima cahaya</b>	<b>Low (0)</b>

Tabel 5.2. Hasil Pengujian *photodiode*

Sumber: Pengujian

Hal tersebut menandakan bahwa rangkaian sensor mobil dapat bekerja sesuai yang diharapkan.

## 5.3 Pengujian Tampilan LCD

### 5.3.1 Tujuan

Pengujian rangkaian tampilan LCD adalah untuk mengetahui apakah rangkaian tampilan LCD dapat menampilkan karakter/data yang sesuai dengan perencanaan.

### 5.3.2 Peralatan yang digunakan

1. Mikrokontroler R8Ctiny 13
2. Mikrokontroler AT89S51
3. Rangkaian LCD
4. Catu daya 5V

### 5.3.3 Prosedur Pengujian

1. Menyusun rangkaian seperti yang ditampilkan pada Gambar 5.3.1 dan 5.3.2
3. Menuliskan program ke dalam program mikrokontroler
4. Menjalankan program di dalam sistem mikrokontroler
5. mengamati hasil eksekusi program yang ditampilkan oleh LCD



**Gambar 5.3.1** rangkaian pengujian LCD dengan sistem MK AT89S51



**Gambar 5.3.2** rangkaian pengujian LCD dengan sistem MK R8C Tiny 13

### 5.3.4 Hasil Pengujian dan Analisis

Dari pengujian didapatkan hasil bahwa rangkaian tampilan LCD bekerja dengan baik, yaitu dapat menampilkan karakter/data sesuai dengan yang direncanakan.

## 5.4 Pengujian unit masukan keypad

### 5.4.1 Tujuan

Pengujian keypad dilakukan untuk membaca tombol yang disusun secara matrik.

### 5.4.2 Peralatan yang Digunakan

1. Keypad matrik 4x4
2. Enkoder 74LS164
3. Catu daya 5 Volt
4. LED

### 5.4.3 Prosedur Pengujian

1. Menyusun rangkaian seperti yang ditampilkan pada Gambar 5.4
2. Menjalankan rangkaian pengujian
3. Mencatat hasil keluaran pada data *output* dan *data available* (DA)



**Gambar 5.4** Rangkaian pengujian unit masukan *keypad*

### 5.4.4 Hasil Pengujian dan Analisis

Dari hasil pengujian diketahui bahwa ketika ada penekanan tombol maka LED dari pin DA akan menyala dan LED dari data output A,B,C,D akan menyala sesuai dengan kode biner data output. Hal tersebut menandakan bahwa rangkaian *keypad* dapat bekerja sesuai yang diharapkan. Data hasil pengujian dapat dilihat pada Tabel 5.4.

Angka pada Keypad	Nyala LED			
	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

**Tabel 5.4** Hasil pengujian keypad disimulasikan dalam nyala LED

## 5.5 Pengujian RTC

### 5.5.1 Tujuan

Pengujian ini bertujuan untuk mengetahui keluaran rangkaian *Real Time Clock*, dimana setelah dimasukkan ke sistem mikrokontroller, keluarannya ditampilkan oleh LCD, kemudian keluaran tersebut dibandingkan dengan *stopwatch*.

### 5.5.2 Peralatan yang Digunakan

1. Sistem mikrokontroller yang terkoneksi dengan LCD
2. Sumber tegangan +5V
3. Stopwatch

### 5.5.3 Prosedur Pengujian

1. Membuat program penguji rangkaian RTC
2. Mengisikan program RTC ke Mikrokontroller AT89S51
3. Menyusun rangkaian penguji seperti pada Gambar 5.5
4. Mengamati keluaran pada rangkaian LCD



Gambar 5.5 Rangkaian pengujian RTC

### 5.5.4 Hasil Pengujian dan Analisis

Hasil catatan detik-detik yang bergerak dari 00:00:00 sampai dengan 00:00:60. Ketika sistem dinyalakan maka tampilan di LCD menunjukkan perubahan setiap detiknya dan dicocokkan dengan stopwatch. Perbandingan waktu antara RTC dan stopwatch dapat dilihat pada Tabel 5.5.

Detik pada RTC	Detik pada <i>Stopwatch</i>
1	1
2	2
.	.
60	60

**Tabel 5.5** perbandingan waktu RTC dan *Stopwatch*

Dari hasil pengujian diketahui bahwa RTC bekerja dengan baik, hal tersebut dapat dilihat dengan dihasilkannya perubahan tiap detik dan hasilnya relatif sesuai dengan perubahan detik yang dihasilkan *stopwatch*. Hal tersebut menunjukkan bahwa RTC bekerja sesuai dengan perencanaan.

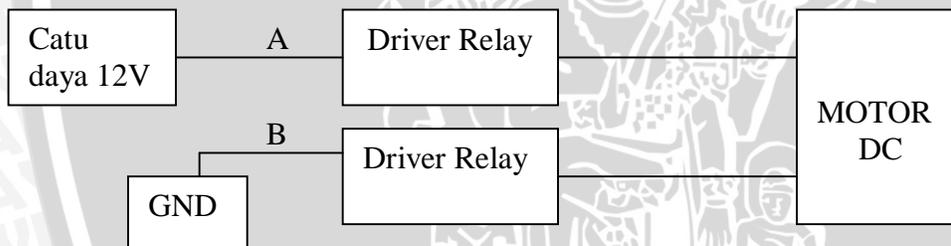
## 5.6 Pengujian rangkaian Motor DC

### 5.6.1 Tujuan

Mengetahui apakah kerja rangkaian motor DC sesuai perencanaan.

### 5.6.2 Langkah pengujiannya adalah sebagai berikut :

- § Menghubungkan rangkaian seperti ditunjukkan dalam Gambar 5.6
- § Memberikan tegangan catu 12 volt
- § Mengamati arah putaran motor DC



**Gambar 5.6** Pengujian motor DC

### 5.6.3 Hasil Pengujian

Masukan logika pada keadaan yang diubah-ubah pada *Driver Relay* yang menentukan arah putaran motor DC ditunjukkan dalam Tabel 5.6.

A	B	Motor DC
0	0	Mati
0	1	Reverse
1	0	Forward
1	1	Forbidden

**Tabel 5.6** Hasil Pengujian Motor DC

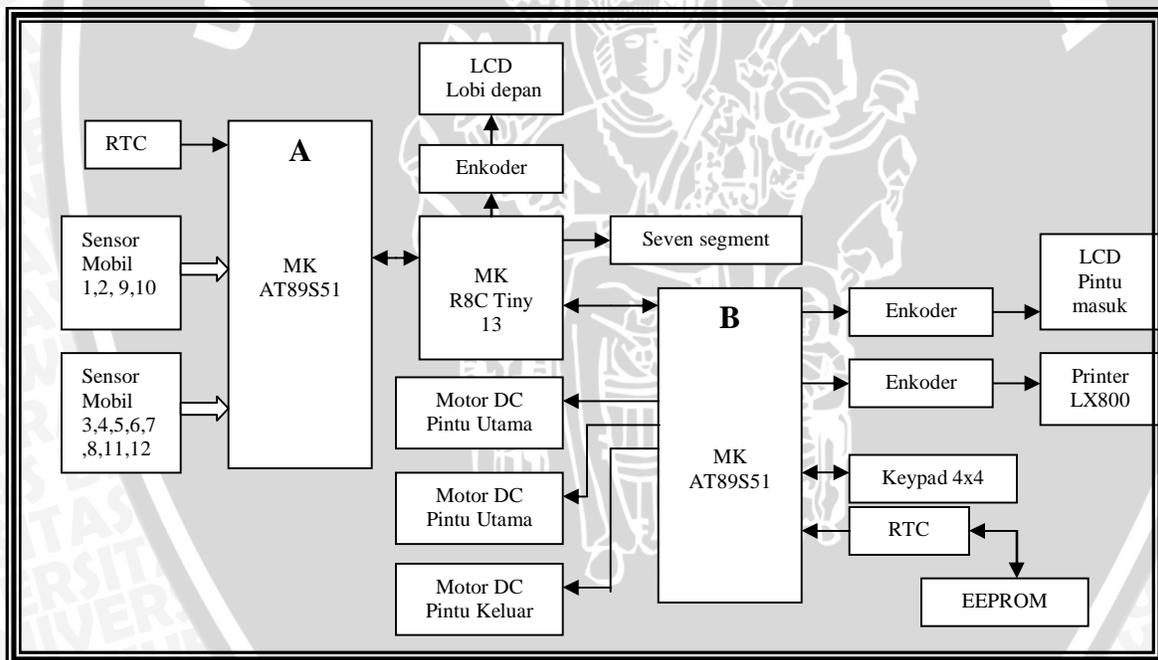
#### 5.6.4 Analisis hasil pengujian

Dari Tabel 5.3 terlihat hasil pengujian sesuai dengan yang direncanakan.. Jika Saklar B “on” maka putaran motor bergerak mundur sehingga palang pintu bergerak naik, sebaliknya maka putaran motor maju sehingga palang pintu bergerak turun. Hal yang dilarang adalah ketika menyalakan saklar A dan B secara bersamaan karena akan terjadi *error* pada system.

### 5.7 Pengujian Sistem keseluruhan

#### 5.7.1 Tujuan

Untuk mengetahui kinerja sistem secara keseluruhan, apakah telah berhasil sesuai yang direncanakan. Blok diagram pengujian sistem keseluruhan ditunjukkan dalam Gambar 5.7.



Gambar 5.7 Pengujian sistem Keseluruhan

#### 5.7.2 Peralatan yang Digunakan

1. Power Supply 5V dan 12V
2. Miniatur sistem parkir secara keseluruhan
3. 2 buah mobil mainan

### 5.7.3 Prosedur Pengujian

1. Membuat program utama dalam bahasa C.
2. Mengisikan program ke dalam mikrokontroller.
3. Menghubungkan catu daya ke rangkaian pengujian.
4. Merangkai perangkat sesuai dengan blok diagram sistem yang direncanakan.
5. Melajukan mobil mainan pada miniatur areal parkir saat sejak masuk hingga keluar areal parkir dengan mengikuti prosedur yang ada.
6. Mengamati hasil uji coba apa sesuai dengan perancangan alat.

### 5.7.4 Hasil pengujian dan analisis

#### 5.7.4.1 Saat Mobil Masuk :

1. LCD 1 bertuliskan “Selamat datang, silakan parkir” jika kuantitas areal parkir kurang dari 20 mobil.
2. Angka pada *seven segment* tertera 00.00.
3. Palang pintu utama terbuka, mobil masuk melewati palang pintu utama kemudian mengenai sensor mobil 1 sehingga LCD 2 bertuliskan “Tekan nomor masuk : “.
4. Menekan angka pada *keypad* dengan batas 0 sampai 20.
5. Palang pintu 1 terbuka, dan menutup secara otomatis setelah mobil melewatinya.
6. Pada *seven segment* angka berubah menjadi 00.01.

#### 5.7.4.2 Saat Mobil Keluar :

1. Mobil akan melewati sensor mobil yang memicu keypad bertuliskan “Nomor keluar : ” jika nomor yang ditekan sesuai dengan nomor masuk maka Printer akan mencetak karcis sesuai nomor parkir.
2. Palang pintu keluar terbuka.
3. Jika nomor salah maka keypad tidak merespon, akan tetap keluar tulisan “Nomor Keluar : “.
4. Palang pintu keluar akan tertutup otomatis setelah mobil keluar.
5. Angka pada *seven segment* berkurang satu.

#### 5.7.4.3 Saat Parkir penuh :

1. Palang pintu utama tertutup secara otomatis.
2. Tulisan pada LCD 1 adalah “Mohon Maaf Parkir Penuh”.

## BAB VI

### PENUTUP

#### 6.1 Kesimpulan

Dari hasil perancangan, pembuatan dan pengujian miniatur sistem parkir otomatis berbasis mikrokontroler R8C Tiny 13 secara keseluruhan, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Alat yang dibuat adalah miniatur areal parkir dua lantai yang terbuat dari papan kayu berbentuk persegi berukuran 60x25x10 cm dengan kapasitas maksimal 20 mobil.
2. *Driver relay* penggerak motor DC pada palang pintu utama akan bergerak *forward* sehingga palang pintu akan tertutup secara otomatis jika areal parkir penuh, dan LCD pada lobi depan akan bertuliskan “Parkir Penuh”.
3. Pada pengujian *keypad* didapatkan bahwa *keypad* bekerja sesuai perencanaan. Hal itu dapat dilihat dari nyala LED yang merupakan simulasi dari penekanan angka pada tombol *keypad* sesuai dengan kode biner.
4. Dari hasil pengujian didapatkan *driver relay* untuk motor DC dapat memicu palang pintu terbuka dan tertutup secara otomatis.
5. Dari hasil pengujian sensor-sensor yang digunakan telah sesuai dengan yang direncanakan dimana LED dan *photodiode* sebagai sensor mobil.
6. Perangkat lunak sistem parkir otomatis ini menggunakan bahasa C untuk mikrokontroler R8C Tiny 13 dan AT89S51 sebagai program untuk mengolah data-data dari sensor-sensor yang terdapat pada areal parkir. Dari hasil pengujian secara keseluruhan terbukti bahwa perangkat lunak yang digunakan telah bekerja sesuai dengan perencanaan.

## 6.2 Saran

Ada beberapa hal yang perlu diperhatikan dalam pengembangan alat ini dikemudian hari. Meskipun alat ini sudah dapat bekerja dengan baik dan sesuai dengan sistem yang direncanakan masih ada hal-hal yang perlu ditingkatkan, diantaranya:

1. Sebaiknya menggunakan sistem catu daya cadangan sehingga apabila terjadi pemadaman aliran listrik sistem masih dapat dijalankan.
2. Untuk pengembangan selanjutnya, dapat digunakan sistem pembayaran otomatis tanpa menggunakan bantuan operator.

UNIVERSITAS BRAWIJAYA



**DAFTAR PUSTAKA**

Tokheim, Roger. 1990. *Elektronika Digital*. Alih bahasa : Ir. Sutisno, M. Eng. Jakarta: Erlangga.

Fitzgerald, Higginbotham, Grabel. *Dasar-Dasar Elektro Teknik Edisi Kelima*. Penerjemah : Pantur Silaban, Ph.D. Jakarta : Erlangga.

Seiko Internacional Inc. M1632 User manual. Liquid Chrystal Display Module.

Malvino, Ap, 1992. *Prinsip-prinsip Elektronika*. Penerjemah: Hanapi Gunawan, Jakarta: Erlangga.

Susana Limanto dan Anton Muljono. 2002. *Algoritma dan Pemrograman*. Jakarta : Dinastindo.

R8C/13 Group Hardware Manual. [HTTP://www.Renesas.com.pdf](http://www.Renesas.com.pdf)

Atmel. 2001. *AT89S51*. ([HTTP://www.Atmel.com.pdf](http://www.Atmel.com.pdf))

